The Minimum Description Length Principle and Reasoning under Uncertainty

Peter Grünwald

Warning: This is the web-version of the thesis. The original version of this thesis was printed as a book by a professional printer. It was set in the *Lucida* font. Because of technical difficulties, the present on-line version had to be set in the standard Latex (Computer Modern) Font. Since the lay-out of the thesis had been specifically designed for the beautiful Lucida font, the present version's lay-out is non-optimal. In order to obtain the book version, please contact the author at `grunwald@cs.stanford.edu`.

ILLC Dissertation Series 1998-03

institute for logic, language and computation

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-5256090
fax: +31-20-5255101
e-mail: illc@wins.uva.nl

The Minimum Description Length Principle and Reasoning under Uncertainty

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof.dr. J.J.M. Franse
ten overstaan van een door het college voor promoties
ingestelde commissie in het openbaar te verdedigen in de
Aula der Universiteit
op donderdag 8 oktober 1998 te 11.00 uur

door

Peter Daniel Grünwald

geboren te Geldrop.

Promotor: Prof.dr.ir. P.M.B. Vitányi

Faculteit Wiskunde, Informatica, Natuurkunde en Sterrenkunde
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

The research reported in this thesis was done at the CWI (Centrum voor Wiskunde en Informatica), the National Centre for Mathematics and Computer Science. The investigations were supported by the Computer Science Research Foundation (SION) which is subsidized by the Netherlands Organization for Scientific Research (NWO). Additional support came from NWO (SIR-grants SIR 13-3211 and SIR 13-3773), the University of Amsterdam, the ILLC, the European Community (ESPRIT Working Group 8556: Neural and Computational Learning (NeuroCOLT)), the University of Helsinki, Finland and from Shell Nederland B.V.

Cover design by Tobias Baanders.
Printed and bound by Printpartners Ipskamp B.V., Enschede, the Netherlands.

ISBN: 90–5776–009–6

"Nulla pluralitas est ponenda nisi per rationem vel experiantiam vel auctoritatem illius, qui non potest falli nec errare, potest convivi."

(A plurality should only be postulated if there is some good reason, experience, or unfallible authority for it.)

– William of Ockham (c. 1285 – c. 1349)

Acknowledgements

The Beginnings In 1992 I was an undergraduate student of Computer Science specializing in Artificial Intelligence. I was interested in the following question: *can computers learn?* I had attended several classes on 'Machine Learning' but somehow I was not satisfied with what I was told there – until one Thursday afternoon I attended professor Vitányi's lecture on Solomonoff's idea - the idea that learning can be cast in terms of data compression, which, in its practical form, leads to the Minimum Description Length (MDL) Principle. I was sold immediately – this was the most fascinating subject I had ever come across.

The Supervisor In 1994, after obtaining my Master's degree, I became a Ph.D. student at the CWI, privileged to work almost everyday on what was still my favorite subject. Paul Vitányi, the man who had introduced me to the field, became my supervisor. I am very grateful for the trust Paul put in me from the beginning, when he decided to appoint me, a student of computer science, to the mathematics job I was applying for. I want to thank him for four years full of advice in scientific and worldly matters alike. I really enjoyed the freedom he gave me: besides MDL, there was opportunity to work on several other topics as well (yes, I do find other topics interesting too).

The Committee The reading committee consisted of Pieter Adriaans, Krzysztof Apt, Johan van Benthem, Phil Dawid, Michiel van Lambalgen, John-Jules Meyer and Kenji Yamanishi. I want to thank them very much for their critical reading of the manuscript. By writing a thesis of 300 pages I made life difficult not just for myself but also for them!

The Colleagues Of my colleagues at CWI I would particularly like to mention my office mates, former office mates and near–office mates: Jeroen van Maanen, André Berthiaume, Louis Salvail, John Tromp, Dieter van Melkebeek, Ronald de Wolf, Jaap-Henk Hoepman, Lance Fortnow, Leen Torenvliet, Harry Buhrman, Wim van Dam and Barbara Terhal. When suffering from thesis stress, buurman Harry was always there to relieve me by telling me that there was really nothing to worry about. Jeroen had an indirect yet important influence on this thesis – much of the work in Part I addresses questions originally raised by him. About Ronald I shall say more on the next page.

MDL Country In 1997, I spent two months at the University of Helsinki, Finland. The work I did there led to several articles, summarized in Part II of this thesis. I want to thank my hosts and co-authors Petri Kontkanen, Petri Myllymäki, Tomi Silander and chef d'équipe Henri Tirri for what I consider a very successful collaboration which I hope we will continue. Henri Tirri pointed out to me that I really *must* read Jaynes' unfinished book and thus has had a considerable influence on part I of this thesis. The hospitality of Petri and his wife Hanna has made my stay up north very pleasant.

The Crisis In the process of writing a thesis, it is not uncommon to develop one or more crises. In my case, it was a single one, short but severe. I am deeply grateful to Martin Donker, Anne Westhoff and upstairs neighbors Daphne de Leur and Guszti Eiben for supporting me during those dark days! Guszti is also to be thanked for his rôle as a kind of 'older brother' in science, giving me useful advice whenever I needed it.

The Inspirers The scientific environment is not always stimulating: papers get rejected, sometimes with good reason, sometimes without; work is sometimes judged solely on whether it belongs to a certain paradigm and not on its scientific value; people are indifferent to your work while you think it should be of great importance to them, and so on. In such an environment, it is crucial that there are also some people who are simply enthusiastic about what you are doing and who urge you to go on with it – it is indeed partly because of them that for the last four years I actually did go on. Several people helped me in this respect at several times – the first ones that come to mind are scientists Bruno Gaume, Johan van Benthem, Phil Dawid and John-Jules Meijer; and friend, former housemate, co-author and paranymph Mark Steijvers.

The Paranymphs Mark and my other paranymph Mischa Bonn are to be thanked for wanting to perform this job – all the more so since they live in Bloomington, Indiana and in Berlin, respectively.

The Readers Apart from my promotor and the reading committee, there have been two people who really spent quite some time reading preliminary bits of this thesis - the aforementioned Ronald de Wolf and my father, Herbert Grünwald. Ronald eats book chapters for breakfast - it's great to have a roommate whom you can simply tell to read and comment upon a chapter, and he'll just do it – often on the same day. My father, who is not at all acquainted with the subject, nevertheless succeeded in giving lots of useful comments. And this brings me to ...

The Dear Ones - I In a lovely valley in the Austrian alps lies the Sanatorium Grünwald, where tired Ph.D. students have always received a warm welcome. When there, my parents made sure that I could relax completely – they took care of everything. I am immensely grateful for their love and their support.

The Dear Ones - II Finally, of all the people who have helped me with my thesis, there is only one who at times really had to suffer. She understood and she hardly complained; she remained as loving and lovely as ever. I cannot express how grateful I am for that, Louise.

Introduction

To be able to forecast future events, science wants to infer general laws and principles from particular instances. Roughly speaking, this process of *inductive inference* is the central theme in statistics, pattern recognition and the branch of Artificial Intelligence called 'machine learning'. The *Minimum Description Length (MDL) Principle* is a relatively recent method for inductive inference [128]. The fundamental idea behind the MDL Principle is that any regularity in a given set of data can be used to *compress* the data, i.e. to describe it using fewer symbols than needed to describe the data literally. The more regularities there are in the data, the more we can compress it.

Formalization of this idea leads to the concept of *Kolmogorov Complexity* [93], which measures the complexity of a set of data by the length of the shortest computer program that prints the data and then halts. The shortest program for the data is then regarded as the optimal model for the data. By equating 'short' with 'simple', we see that this amounts to a mathematical version of Occam's Razor [93]: if several explanations (programs) of a given phenomenon (data) exist, then we should pick the simplest (shortest) one[1].

The Minimum Description Length Principle

It can be mathematically shown that data compression in terms of Kolmogorov Complexity is almost always the best way to proceed [160]. However, Kolmogorov Complexity is uncomputable and cannot directly be used in practice. The MDL Principle scales down the ideas behind Kolmogorov Complexity in order to make them applicable in practical settings. Rather than considering all computer programs as possible models for the data, one focuses on a class of models $\mathcal{M}$ that is simple enough to allow us to compute, for each model or *hypothesis* H in the class $\mathcal{M}$, how much the data can be compressed on the basis of that model. To describe or equivalently, *encode* the data on the basis of a model H, one first encodes H itself and one then encodes the data with the help of hypothesis H. Choosing the H that minimizes the total code length of the data automatically leads to the selection of a hypothesis that incorporates a trade-off between the complexity and the goodness-of-fit of the hypothesis. The reason for this is that the better a hypothesis fits the data at hand, the more *information* it gives us about the data. The more information we have about the data, the fewer bits we need to encode it. Intuitively, this is because we do not have to

[1] See page v for a formulation that can actually be found in Occam's work ([40], pages 115–117).

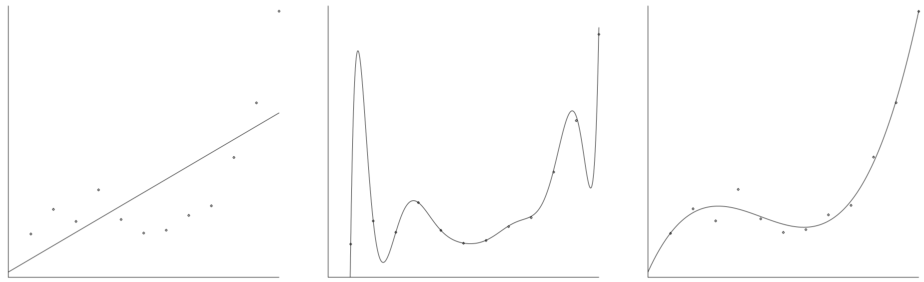


Figure 1: A simple (1.1), complex (1.2) and a trade-off (3rd degree) polynomial.

encode the full data, but only the 'discrepancies' between the hypothesis and the data. It is always possible to find a very complex hypothesis that fits the data extremely well but that does not tell us anything interesting about it. Encoding the data with the help of such a hypothesis takes only very few bits, but to encode (describe) the data we need to describe the hypothesis itself too. The more complex a hypothesis, the more bits we need to describe it. As a result, the gain in description length of the data using a very complex hypothesis is more than offset by the large number of bits needed to describe such a hypothesis. An overly complex hypothesis will therefore not lead to the shortest possible description length. On the other hand, using a very simple hypothesis, the opposite effect occurs: the description length of the hypothesis is very small, but the description length of the data when encoded with the help of the hypothesis will be very high. Assuming that we have 'meaningful' data, that is, it does not consist of purely random noise, the shortest possible description length will usually be attained for relatively simple hypotheses that fit the data reasonably well; according to MDL, these should be preferred both over overly complex hypotheses that have no error at all and over overly simple hypotheses that have very high error. We give a little example. Suppose we want to fit a polynomial to the set of points depicted in Figure 1. Classical linear regression will give us the leftmost polynomial - a straight line that seems overly simple: it does not capture the regularities in the data well. Since for any set of k points there exists a polynomial of the $(k-1)$-st degree that goes exactly through all these points, simply looking for the polynomial with the least error will give us a polynomial like the one in the second picture. This polynomial is overly complex: it reflects the random fluctuations in the data rather than the general pattern underlying it. It seems more reasonable to prefer a simple polynomial with small but nonzero error, as in the rightmost picture. This intuitively right polynomial is exactly the polynomial that the MDL Principle will mechanically give us.

Overview

Here we report the research the author has conducted over the last four years. Most of this research is, either directly or indirectly, related to the MDL Principle. The main research goals were to provide additional justifications for the MDL Principle and to investigate MDL's view on probabilities. We consider both goals in turn.

Justifications of MDL The results of applying the MDL Principle are generally very intuitive. However, mere intuition does not provide enough justification to adopt it for practical use. In this thesis we consider additional justifications for the MDL Principle. Inductive principles like MDL can be justified in different ways. In a theoretical justification, one typically proves that under certain conditions the hypothesis chosen by the principle leads to *provably optimal prediction or classification of future data*. Another theoretical means of justification is to show that a principle can be interpreted as a formal extension of other, existing principles that are already successfully used in practice. A third, practical means of justification is to actually apply the principle in practice and show that this leads to good results. This thesis contains justifications of both theoretical and practical nature.

Probability and Reasoning under Uncertainty Both in the statistical and in the 'reasoning under uncertainty' literature there is fierce debate between those who hold the view that 'all uncertainty should be handled using probability theory' and those who think that probability theory in itself is not enough. In the field of statistics, the two sides in this dispute are occupied by the *Bayesian*[2] or *subjectivist* statisticians on the one hand and the *frequentist* (also called *classical* or *objectivist*) statisticians on the other hand [17]. In the field of reasoning under uncertainty, the discussion is between, once more, the Bayesians on one side, and a menagerie of research communities on the other side; we mention the fields of *fuzzy logic and possibility theory* [41], *certainty factors*, [114], *Dempster-Shafer theory* [138] and *nonmonotonic reasoning* [56].

One of the most controversial aspects of the subjectivist view (actually, it is so controversial that there even are Bayesians who do not accept it[3]) is the *Principle of Insufficient Reason*, advocated as a basis of subjective probability by the great probability theorist P.S. Laplace (1749-1827) [91]:

> If we know that one of k possible alternatives will actually take place, but we have no idea whatsoever whether any of these alternatives is more likely than any other one, then we should assign each of the alternatives probability $1/k$.

The question of whether this makes sense or not has divided philosophers, logicians and statisticians into two camps for over 180 years now [158]. More recently (1957), E.T. Jaynes (1922-1998) extended Laplace's Principle to the 'Principle of Maximum Entropy' [73] which allows one to assign probabilities in the presence of *partial* knowledge. In principle, Maximum Entropy can be used in most if not all of the problems of 'reasoning under uncertainty'. It has actually been applied in lots of practical settings, generally with good results. Nevertheless, it remains as controversial as Laplace's original principle and is still rejected by many [77].

MDL's View on Probabilities It so happens that the notions of 'description method' and 'probability distribution' are very closely connected: every description

[2]Named after Thomas Bayes (1702-1761).

[3]Interestingly, R.T. Cox who provided a very strong justification of the subjectivist viewpoint strongly disagreed with Laplace's principle. See [31], page 31.

method or *code* can be re-interpreted as a probability distribution and vice versa. If data can be coded with the help of model H in only a few bits, then this may be reinterpreted as saying 'the data has a high probability under H - one can define 'probabilities' in this way even for non-probabilistic models like polynomials. From the MDL point of view, a *model* of the data is really a means of describing properties of the data, and hence a 'model' coincides with a 'description method'. Because of the correspondence referred to above, a probability distribution can also be seen simply as a means to describe properties of the data.

This view of probabilities, mainly developed by Rissanen [128], is not so different in principle from the view that is held by some of the Bayesians [36, 38, 77]. What is new is the explicit interpretation of probability distributions in terms of description methods. As discussed informally by Rissanen [128], this interpretation sheds new light on the discussion between the subjectivists and the objectivists. In this thesis, we work out and extend Rissanen's ideas. Briefly, we reach the following conclusion: Probabilities can – and should – indeed be used in many situations where they are not related to frequencies; the 'maximum entropy principle' can indeed be used to assign probabilities in the presence of ignorance. *However, the way such probabilities should be used to arrive at predictions and decisions is different from the way probabilistic knowledge is usually applied.*

Contents of this Work

The thesis consists of three parts. Each part starts with a brief, informal introduction. Part I offers a self-contained introduction to the MDL Principle (chapters 1 and 2) and to Maximum Entropy (Chapter 3). Chapters 4 and 5 introduce some new theoretical concepts concerning MDL. In an Epilogue (page 119) to Part I, it is argued that with the help of these concepts one can identify conditions under which overly simple models can be used unproblematically; this leads to the conclusions about the applicability of probabilistic methods that we sketched above.

In part II, we give an empirical comparison between MDL and some other methods for inductive inference by testing their performance on real-world data. Specifically, in Chapter 6 we compare MDL to the related but distinct Bayesian and maximum likelihood methods of inference. In Chapter 7 we compare MDL to its nearest 'competitor': the Minimum *Message* Length Principle [161, 162].

Part III reports on additional research we have done in the field of 'Reasoning under Uncertainty' within a non-probabilistic paradigm: *non-monotonic temporal reasoning* [56]. Chapter 8 gives an introduction to this field. Chapters 9 and 10 report on the basic research we performed in this field, which is connected to the rest of this thesis through its implicit use of probability, which in turn is implicitly treated from an MDL perspective. In an Epilogue to part III (page 269) we make these connections explicit.

Contents

Part I

Theory of MDL

Introduction to Part I

In 1996, the present author attended a conference at which a well-known statistician gave a talk on Bayesian networks, a particular kind of probabilistic model. He artificially generated a set of data by sampling from a quite complicated Bayesian network. He used a 'learning algorithm' to find a good model for the data, and this learning algorithm came up with a much simpler Bayesian network. While this network was obviously not the 'correct' one, it did work quite well in predicting and classifying future data. Since the predictions of the overly simple model could be calculated much more efficiently than those of the correct model, the statistician suggested to use the simple model instead of the correct one for doing such predictions.

A member of the audience (actually, another well-known statistician) strongly objected. He said that modeling data with an overly simple model can potentially lead to disastrous results (he even claimed that the explosion of the Challenger space shuttle in 1987 was caused by using overly simple models). Of course, he said, 'Occam's Razor' is useful: if we do not have any further knowledge about the process generating our data, it makes sense to pick a reasonably simple model with small error rather than a complex model with zero error. But if we really *know* that the data is better modeled by a complex model, we should certainly not use an overly simple one.

This led to fierce discussions among the audience. On the one side there were those who said that there was nothing wrong in principle with using a model that is overly simple; on the other side were those who agreed that using an overly simple model is always a risky thing to do. Unfortunately, neither of the two sides in this debate had any means of *proving* they were right: all arguments were based on intuition.

When observations arrive sequentially, the MDL Principle will often pick an overly simple model for the first few observations. In the polynomial example (Figure 1 on page x), MDL may prefer a straight line for the first n observations. Only when there is enough data available will the lower error achieved by a third degree polynomial outweigh its extra complexity. Hence, the discussion at the conference is highly relevant to MDL: is the MDL Principle only useful in that it selects a good model *when enough data is available*? Or can the overly simple model that will be selected in the presence of few data be fruitfully used as well, without potentially 'disastrous' results?

It is this question which we attempt to answer in part I of this thesis. Rather than being based merely on intuition, our answer will be mathematically precise. The question is relevant not only to MDL, but actually to most of current statistical practice. For most models we use in practice are evidently too simple: we fit straight lines to data that are clearly not really linearly related (as an example, think about models in economics); we model several parts of data as if they were independent while we know they are not independent at all (an example is speech recognition [148]). Does this lead to potentially disastrous results? If so, how come we have got away with this for so long?

A second goal of part I of this thesis is to provide an introduction to MDL that can be read without any previous knowledge of either information theory or statistics; this introduction will also provide the necessary background needed for parts II and III. Chapter 1 gives a self-contained introduction to the MDL Principle that avoids complicated mathematics. In Chapter 2, we discuss the fundamental concept of MDL: the *stochastic complexity*. Chapter 3 gives an introduction to the *Maximum Entropy Principle* that we mentioned on page xi and connects it to the MDL Principle. Both Chapter 1 and Chapter 2 mention some problematic issues that, we think, are not fully satisfactorily handled by the existing theory. In Chapter 1 this is the issue of what exactly a model says about the modeled situation. In Chapter 2 this is the question of whether and how non-probabilistic model classes can be recast in probabilistic terms. Chapter 3 reviews an argument that has led many researchers to reject Maximum Entropy. In chapters 4 and 5 we show that all three issues are just aspects of the basic question whether or not one can justify the use of an overly simple model (or more generally, a model that captures some, but not all regularities underlying the data), and we make an attempt to answer the question.

Chapter 1

A First Introduction to the MDL Principle

This chapter aims to give a self-contained introduction to the MDL Principle. In Section 1.1 we explain the fundamental ideas and intuitions behind MDL. Section 1.2 provides the information-theoretic background that is necessary to formalize MDL. We proceed to formalize the MDL Principle for probabilistic model classes (Section 1.3) and for non-probabilistic model classes (Section 1.4). Section 1.5 discusses an important aspect of the MDL *Philosophy*, namely, that the goal of inductive inference should be to look for a 'good', and not for a 'true' model.

In order to make the introduction self-contained and at the same time not too complicated we avoid discussion of two very important topics: both the treatment of the *stochastic complexity* and of the exact relation between MDL and Bayesian statistics are deferred to Chapter 2.

1.1 The Fundamental Idea

The task of inductive inference is to find laws or regularities underlying some given set of data. These laws are then used to gain insight in the data or to classify or predict future data. In the statistics literature, the data we are given are usually called the 'sample' or the 'observations'. The aim of finding laws underlying the data is usually cast in terms of finding a good *model* for the data. The process of finding such a model is called *statistical inference*. In the machine learning and neural network literature [110, 111, 72], the sample is usually called the *training set*, a model is often called a *hypothesis* and future data is often called the *test set*. The process of finding a good model is called 'learning'. Throughout this thesis, we use statistical and machine learning terminology interchangeably.

Example 1.1 We start by considering binary data. Consider the following three sequences. We assume that each sequence is 10000 bits long, and we just list the beginning and the end of each sequence.

$$\texttt{00010001000100010001}\ldots 0001000100010001000100010001 \tag{1.1}$$
$$\texttt{01110100110100001010}\ldots 1010111010111011000101100010 \tag{1.2}$$
$$\texttt{00011000001010010100}\ldots 0100010000010000001000110000 \tag{1.3}$$

The first of these three sequences is a 2500-fold repetition of `0001`. Intuitively, the sequence looks regular; there seems to be a simple 'law' underlying it; it might make sense to conjecture that future data will also be subject to this law, and to predict that future data will behave according to this law. The second sequence has been generated by tosses of a fair coin; this means that there is definitely no 'law' or regularity underlying it. Indeed, we cannot seem to find such a law either when we look at the data. The third sequence contains exactly four times as many 0s as 1s. It looks less regular, more random than the first; but it looks less random than the second. There is still some discernible regularity in this data, but of a statistical rather than of a deterministic kind. Again, noticing that such a regularity is there and predicting that future data will behave according to the same regularity seems like a sensible thing to do.

1.1.1 Regularity and Compression

What do we mean by a 'regularity'? The fundamental idea behind the MDL Principle is the following insight: every regularity in the data can be used to *compress* the data, i.e. to describe it using less symbols than the number of symbols needed to describe the data literally. Such a description should always uniquely specify the data it describes - hence given a description or *encoding* D' of a particular sequence of data D, we should always be able to fully reconstruct D using D'.

For example, sequence (1.1) above can be described using only a few words – we have actually done so already: we have not given the complete sequence - which would have taken about the whole page - but rather just a one-sentence description of it that nevertheless allows you to reproduce the complete sequence if necessary. Of course, the description was done using natural language and we may want to do it in some more formal manner.

If we want to identify regularity with compressibility, then it should also be the case that non-regular sequences can *not* be compressed. Since sequence (1.2) has been generated by fair coin tosses, it should not be compressible. As we will show below, we can indeed prove that *whatever* description method C one uses, the length of the description of sequence (1.2) will, with overwhelming probability, be not much shorter than sequence (1.2) itself.

Notice that our description of sequence (1.3) as given above does not uniquely define sequence 3. Therefore, it does not count as a 'real' description: one cannot regenerate the whole sequence if one has the description. A unique description that still takes only a few words may look like this: "Sequence (1.3) is one of those sequences of 10000 bits in which there are four times as many 0s as there are 1s. In the lexicographical ordering of those sequences, it is number i". Here i is some large number that is explicitly spelled out in the description. In general, there are 2^n binary sequences of length n, while there are only $\binom{n}{\gamma n}$ sequences of length n with a fraction of γ 1s. For every rational number γ except $\gamma = 1/2$, the ratio of $\binom{n}{\gamma n}$ to 2^n goes to 0 exponentially fast as n increases (this is shown formally in Chapter 3, page 59; by the method used there one can also show that for $\gamma = 1/2$, it goes to 0 as $O(1/\sqrt{n})$). It follows that compared to the total number of binary sequences of length 10000, the number of sequences of length 10000 with four times as many 0s than 1s is vanishingly small. Therefore, $i \ll 2^{10000}$ and to write down i in binary we need approximately $(\log_2 i) \ll 10000$ bits.

Description Methods and Codes In order to formalize our idea, we have to replace the part of the descriptions that made use of natural language by some formal language. For this, we need to fix a *description method* that maps sequences of data to their descriptions. Throughout this thesis, we assume that our data D always consists of a sequence of symbols coming from some fixed finite or countably infinite *data alphabet*. Each such sequence will be encoded as another sequence of symbols coming from some finite or countably infinite *coding alphabet*. An *alphabet* is simply a countable set of distinct symbols. An example of an alphabet is the binary alphabet $\mathbf{B} = \{0, 1\}$; the three data sequences above are sequences over the binary alphabet. A sequence over a binary alphabet will also be called a binary *string*. Sometimes our data will consist of real numbers rather than binary strings. In practice however, such numbers are always truncated to some finite precision d. We can then again model them as symbols coming from a finite data alphabet.

We let $\mathbf{A}^n$ stand for the n-fold Cartesian product of alphabet $\mathbf{A}$. We define λ as the empty sequence and define $\mathbf{A}^0 = \{\lambda\}$. We let $\mathbf{A}^* = \bigcup_{i=0}^{\infty} \mathbf{A}^i$ and finally we write $\mathbf{A}^+$ as an abbreviation of $\mathbf{A}^* \setminus \mathbf{A}^0$.

In our problem setting we are given a *sample* or equivalently *data sequence* $D = (x_1, \ldots, x_n) \in \mathbf{A}^*$ where each $x_i \in \mathbf{A}$. We call x_i a single observation or, equivalently, a *data item*. We sometimes use the notation x^i for $(x_1, \ldots, x_i)$. Similarly, whenever the *sample size* or equivalently *data sequence length* n is not clear from the context, we write x^n in stead of D. For convenience, it will sometimes be useful to write $x^0 \equiv \lambda$ ($A \equiv B$ is to be read as 'A is defined to be equal to B').

As argued in [128], without any loss of generality we can always describe our data sequences as binary strings. Hence all the description methods we consider map data sequences to sequences of bits. We use the term 'code' as a specific case of a 'description method[1]'.

Definition 1.2 *Let* $\mathbf{A}$ *be some data alphabet. A* code C *is a one-to-one map from* $\mathbf{A}^+$ *to* $\mathbf{B}^+$. *A* description method *is a one-many relation over* $\mathbf{A}^+ \times \mathbf{B}^+$.

Codes and description methods have in common that given any encoding, it is always possible to reconstruct the data sequence that it describes. Description methods are a more general notion than codes in that for a code, there should always be only *one* possible encoding of every data sequence. For a description method, there may be data sequences that can be encoded in several ways.

[1]Our definition of 'description method' coincides with what Rissanen [128] calls a 'coding system'. Our definition of 'code' coincides with Rissanen's [128]. In the standard textbook [30], what we here call a code is called 'uniquely decodable code'.

Here is a simple example (copied from Rissanen [128]) of a code C_1 for a data alphabet $\mathbf{A}_1 = \{a, b, c\}$. C_1 is defined as follows: $C_1(a) = 0, C_1(b) = 10, C_1(c) = 11$; for all $x, y \in \mathbf{A}_1^+, C_1(xy) = C_1(x)C_1(y)$. We call $C_1(a)$ the *codeword* of a. For example, data sequence $aabac$ is encoded as $C_1(aabac) = 0010011$. It is easy to see that no two different data sequences can have the same code word; hence from an encoded sequence $C_1(x)$ we can always retrieve the original sequence x.

Compression and Small Subsets We are now in a position to prove our claim that it is impossible to substantially compress sequences that have been generated by fair coin tosses. Let us take some arbitrary but fixed description method C over the binary data alphabet $\mathbf{A} = \{0, 1\}$. Suppose we are given a data sequence of length n (in our example, $n = 10000$). Clearly, there are 2^n possible data sequences of length n. We see that only two of these can be mapped to a description of length 1 (since there are only two binary strings of length 1: '0' and '1'). Similarly, only a subset of at most 2^m sequences can have a description of length m. This means that at most $\sum_{i=1}^{m} 2^m < 2^{m+1}$ data sequences can have a description length $\leq m$. The fraction of data sequences of length n that can be compressed by more than k bits is therefore at most 2^{-k} and as such decreases exponentially in k. If data are generated by n tosses of a fair coin, then all 2^n possibilities for the data are equally probable, so the probability that we can compress the data by more than k bits is smaller than 2^{-k}. For example, the probability that we can compress the data by more than 20 bits is smaller than one in a million.

Seen in this light, having a short code for the data is equivalent to identifying the data as belonging to a tiny, very *special* subset out of all a priori possible data sequences.

1.1.2 Solomonoff's Breakthrough

It seems that what data is compressible and what not is extremely dependent on the specific description method used. In 1964 – in a pioneering paper that may be regarded as the starting point of all MDL-related research [145] – Ray Solomonoff suggested to use a *universal computer language* as a description method. By a universal language we mean a computer language in which a Universal Turing Machine can be implemented. All commonly used computer languages, like Pascal, LISP, C, are 'universal'. Every data sequence D can be encoded by a computer program P that prints D and then halts. We can define a code that maps each data sequence D to the *shortest program* that prints D and then halts[2]. Clearly, this is a code in the sense of Definition 1.2 above in that it defines a 1-1 mapping from sequences over the data alphabet to binary sequences. The shortest program for a sequence D is then interpreted as the *optimal model* for D. Let us see how this works for sequence (1.1) above. Using a language similar to C, we can write a program

```
for i = 1 to 2500 ; do {print '0001'} ; halt
```

[2]If there exist more than one shortest program, we pick the one that comes first in enumeration order.

which prints sequence (1.1) but is clearly a lot shorter than it. If we want to make a fair comparison, we should rewrite this program in a binary alphabet; the resulting number of bits is still much smaller than 10000. The shortest program printing sequence (1.1) is at least as short as the program above, which means that sequence (1.1) is indeed highly compressible using Solomonoff's code. By the arguments of the previous section we see that, given an arbitrary description method C, sequences like (1.2) that have been generated by tosses of a fair coin are very likely not substantially compressible using C. In other words, the shortest program for sequence (2) is, with extremely high probability, not much shorter than the following:

```
print '01110100110100001010........111010111011000101100010'; halt
```

This program has size about equal to the length of the sequence. Clearly, it is nothing more than a repetition of the sequence.

Kolmogorov Complexity We define the *Kolmogorov Complexity* of a sequence as the length of the shortest program that prints the sequence and then halts. 'Kolmogorov Complexity' has become a large subject in its own right; see [93] for a comprehensive introduction.

The lower the Kolmogorov complexity of a sequence, the *more regular* or equivalently, the *less random*, or, yet equivalently, the *simpler* it is. Measuring regularity in this way confronts us with a problem, since it depends on the particular programming language used. However, in his 1964 paper, Ray Solomonoff [145] showed that it does not matter exactly what programming language one uses, as long as it is universal: for every sequence of data $D = (x_1, \ldots, x_n)$, let us denote by $L_U(D)$ the length of the shortest program for D using universal language U. We can show that for every two universal languages U_1 and U_2, the difference between the two lengths $L_{U_1}(D) - L_{U_2}(D)$ is bounded by a constant that depends on U_1 and U_2 but not on the length n of the data sequence D. This implies that if we have a lot of data (n is large), then the difference in the two description lengths is negligible compared to the size of the data sequence. This result is known as the *invariance theorem* and was proved independently by Solomonoff [145], Kolmogorov [86] (hence the name 'Kolmogorov' complexity) and Chaitin [23]. The proof is based on the fact that one can write a compiler for every universal language U_1 in every other universal language U_2. Such a compiler is a computer program with length $L_{1,2}$. For example, we can write a program in Pascal that translates every C program into an equivalent Pascal program. The length (in bits) of this program would then be $L_{\mathrm{C,Pascal}}$. We can simulate each program P_1 written in language U_1 by a program P_2 written in U_2 as follows: P_2 consists of the compiler from U_1 to U_2, followed by P_1. The length of program P_2 is bounded by the length of P_1 plus $L_{1,2}$. Hence for all data D, the maximal difference between $L_{U_1}(D)$ and $L_{U_2}(D)$ is bounded by $\max\{L_{1,2}, L_{2,1}\}$, a constant which only depends on U_1 and U_2 but not on D.

1.1.3 Making the Idea Applicable

There are two problems with applying Kolmogorov Complexity to practical learning problems. First, the Kolmogorov Complexity as such cannot be computed – there is no computer program that, for every sequence of data D, when given D as input, returns a shortest program that prints D and halts. Neither can there be a program, that for every set of data D returns only the *length* of the shortest program that prints D and then halts. Assuming such a program exists leads to a contradiction [93]. Another problem is that in many realistic settings, we are confronted with very small data sequences for which the 'invariance theorem' is not very relevant since the length of D is small compared to the constant $L_{1,2}$.

The idea behind the MDL Principle is now to scale down Solomonoff's approach so that it does become applicable: instead of a universal computer language as a description method, we should use description methods C which still allow us to compress many of the intuitively 'regular' sequences but which nevertheless also allow us to compute, for every D, the length of the shortest description of D attainable using C. The price we pay is that, using the 'practical' MDL Principle, there will always be some regular sequences which we will not be able to maximally compress. But we already know that there can be *no* method for inductive inference at all which will always give us all the regularity there is - simply because there can be no automated method which for any sequence D finds the shortest computer program that prints D and then halts. Moreover, it will often be possible to guide a suitable choice of C by a priori knowledge we have about our problem domain. For example, we will see below that it is possible to pick a code C with which we can compress all sets of 2-dimensional data which are 'regular' in the sense that the x-values are related to the y-values by some simple polynomial. If we have a priori reasons to believe that our data points are related by *some* polynomial, but we just do not know which, then this C may be good choice.

1.1.4 Two-Part Codes; First Instantiation of the MDL Principle

Let $\mathcal{M}$ be some given class of models, for example, the class of all polynomials. To make our idea work, we need a code that compresses those data sets that are well-described by one of the models in $\mathcal{M}$. For the model class $\mathcal{M}_U$ consisting of all computer programs written in some universal language U, the corresponding code $C_{\mathcal{M}}$ would simply be the code mapping each D to the shortest program that prints D and then halts. For model classes like the polynomials, whose elements do not directly define a description method, one can construct the codes $C_{\mathcal{M}}$ in several ways. In this chapter, we concentrate on the so-called two-part codes, which are just one possible instantiation of $C_{\mathcal{M}}$. We should stress at the outset that there are more possibilities. Since the two-part codes are conceptually the simplest (and also historically the first ones to have appeared [161, 126]), we defer discussion of other possibilities to Chapter 2.

The idea behind the two-part code is to describe a data sequence D by first describing some hypothesis $H \in \mathcal{M}$ and then describing the data with the help of the hypothesis, which usually amounts to specifying the *errors* the hypothesis makes on the data.

MDL Principle (two-part code version) Among the set of candidate hypotheses $\mathcal{M}$, the best hypothesis to explain a set of data is the one which minimizes the sum of

- the length, in bits, of the description of the hypothesis; and
- the length, in bits, of the description of the data when encoded with the help of the hypothesis.

A hypothesis with a long description length is what we would intuitively call a 'complex' one. A description of the data 'with the help of' a hypothesis means that the better the the hypothesis fits the data, the shorter the description will be. A hypothesis that fits the data well gives us a lot of *information* about the data. Such information can always be used to compress the data (we will see a specific instance of this phenomenon in the example below). The better the fit, the more information the hypothesis gives about the data and the less bits we need to describe the data with the help of the hypothesis. The MDL Principle thus gives us a trade-off between hypothesis complexity and goodness-of-fit on the data set. We proceed with an example.

Example 1.3 [under- and overfitting] Let $D = ((x_1, y_1), \ldots, (x_n, y_n))$ be a sequence of data consisting of pairs (x, y) where the x and y are real numbers. We are interested in finding a functional relationship between x and y: we look for a function H such that $H(x)$ predicts y reasonably well. Again we may want to use the H we found for predicting future data. The classical statistical solution to this problem is to do a standard regression: we select the linear function that is optimal in the least-squares sense, i.e. the linear function H for which the sum of the squared errors $\sum_{1 \leq i \leq n} (H(x_i) - y_i)^2$ is as small as possible (Figure 1.1 on page x). We end up with a line that seems to capture *some* of the regularity in the data, but definitely not very much - it seems to *underfit* the data. A more interesting task is to look for the best curve within a broader class of possible candidates. For example, let us look for the best *polynomial*. A naive extension of the classical statistical solution will now typically pick a polynomial of degree $n - 1$ that completely covers the data: it will go *exactly* through all the points (x_i, y_i) (Figure 1.2). Intuitively, this may not be the polynomial we are looking for – we run a large risk of *overfitting*. Again intuitively, we might prefer a third-degree polynomial (Figure 1.3): one that has small (but not 0) error and still is relatively simple (i.e. has few coefficients). In other words, we are looking for an optimal trade-off between model complexity (i.e. the number of coefficients in the polynomial) and goodness-of-fit. The more coefficients a polynomial H has, the more bits we need to describe it. On the other hand, we can use a polynomial H to compress the description of data points (x^n, y^n). Rather than describing them literally by a list $(x_1, \ldots, x_n, y_1, \ldots, y_n)$ (encoded in binary), we encode the list $(x_1, \ldots, x_n)$ and the list of errors $(y_1 - H(x_1))^2, \ldots, (y_n - H(x_n))^2)$. If the polynomial H fits the data well, the *range* of the errors will be much smaller than the range of y_i. Therefore, we need less bits to encode the errors than to encode the y_i. The two-part code MDL Principle opts for the hypothesis that minimizes the sum of the two description lengths involved and therefore achieves the desired trade-off. It can be shown [14] that such a trade-off is not just *intuitively*, but in many cases also *provably* a good choice.

1.2 Probabilistic Preliminaries

We have informally presented the basic ideas behind the MDL Principle. To make these ideas more formal, we review some basic facts of probability theory and information theory. We first introduce much of the notation that will be used throughout this thesis:

Notational Conventions We work with probability distributions P defined over a *sample space* $\mathbf{E}$. Throughout this thesis, $\mathbf{E}$ will always be either finite, countably infinite or a subset of $\mathbf{R}^k$. Here $\mathbf{R}$ stands for the real numbers and $k \geq 1$. If $\mathbf{E} \subset \mathbf{R}^k$ then we call $\mathbf{E}$ *continuous*.

Let X^n be a random variable that can take on values in $\mathbf{E}^n$. In the case of countable $\mathbf{E}$, we abbreviate $P(X^n = x^n)$ to $P(x^n)$. In the case of continuous $\mathbf{E}$, $f : \mathbf{E} \rightarrow [0, 1]$ denotes the *density function* of P. We often deal with sample spaces that are allowed to be either countable or continuous. In that case, we will give the formulas only for the countable case. The appropriate formulas for the continuous case can be arrived at by substituting $f(x)$ for $P(x)$ and replacing all sums by corresponding integrals. For example, $\sum_{x \in \mathbf{E}} P(x)\phi(x)$ becomes $\int_{x \in \mathbf{E}} f(x)\phi(x)dx$.

The definition of $\mathbf{A}^+$ and $\mathbf{A}^*$ (page 7) is extended to sample spaces $\mathbf{E}$. We use 'log' for logarithm to base two, and 'ln' for natural logarithm. For $x \in \mathbf{R}$, the *ceiling* of x, denoted by $\lceil x \rceil$, stands for the smallest integer that is greater than or equal to x. For a finite set $\mathbf{X}$, we use $|\mathbf{X}|$ to denote the number of elements in $\mathbf{X}$.

For all $D \in \mathbf{E}^*$, we denote by $L_C(D)$ the length (number of bits) of the description of D when the description is done using code C. In some cases we will use indexed codes C_i. We then use L_i as shorthand for L_{C_i}.

We extend the definition of codes to *partial* odes:

Definition 1.4 *Let* $\mathbf{A}$ *be some alphabet. Let* $\mathbf{A}'$ *be some subset of* $\mathbf{A}^+$. *A* partial code *for the set* $\mathbf{A}'$ *is a one-to-one map from* $\mathbf{A}'$ *to* $\mathbf{B}^+$.

If, for a partial code C, $C(D)$ is not defined, then we write $L_C(D) = \infty$. Henceforth, we use the word 'code' to denote both proper (as in Definition 1.2) and partial codes.

1.2.1 Probabilistic Models

Many interesting model classes are *probabilistic* in the sense that each model in the class represents a probability distribution over all possible data sequences. Examples are the class of all Bernoulli processes, the class of all Markov Chains, the class of all normal distributions etc. Intuitively, a *probabilistic model* is a probability distribution defined over arbitrarily long samples[3]. To formally define probabilistic *models* as a particular

[3]Our definition of a probabilistic model coincides with what Rissanen [128] calls an *information source* and what is called a 'random process' in probability theory.

class of probability *distributions* requires measure theory, which we want to avoid here. We therefore adopt a standard alternative definition:

Definition 1.5 *Let* $\mathbf{E}$ *be a sample space. A* probabilistic model *over* $\mathbf{E}$ *is a function* $P : \mathbf{E}^* \to [0,1]$ *such that for all* $n \geq 0$*, all* $x^n \in \mathbf{E}^n$ *we have:*

1. $\sum_{z \in \mathbf{E}} P(x^n z) = P(x^n)$ *(this is usually called the* compatibility condition*).*

2. $P(\lambda) = 1$ *where* λ *is the empty sequence.*

The two conditions above simply say that the 'event' that data $x^n z$ arrives is equivalent to the event that data x^n arrives first and data z arrives afterwards [128]. If $\mathbf{E}$ is continuous then the sum in the definition above is replaced by an integral but otherwise nothing changes.

Let P be a probabilistic model over $\mathbf{E}^*$. If for all n, all $x^n \in \mathbf{E}^n$ and $x_{n+1} \in \mathbf{E}$, $P(x_{n+1}|x^n) = P(x_{n+1})$, we say that the data are *i.i.d.* (independently and identically distributed) according to P. We call a class of probabilistic models an *i.i.d. model class* if it fully consists of probabilistic models which render the data i.i.d.

1.2.2 Connecting Codes and Probability Distributions

In this subsection we establish a very fundamental connection between codes and probability distributions; it is this connection which brings MDL into the realm of probability theory and traditional statistics.

We first reconsider the code C_1 for alphabet $\mathbf{A}_1 = \{a, b, c\}$ introduced in Section 1.1.1. An interesting feature of this code is that we do not need any commas separating the code words. This is possible because no extension of a code word can itself be a code word. Hence if we de-code from left-to-right, we always know at what point the subsequence used to encode one particular data symbol ends and the encoding of the next symbol starts, thus eliminating the need for commas. Codes with this property are usually called *instantaneous* or *prefix* codes (since no code word can be a *prefix* of any other code word).

Notice that if it were necessary to use commas in our description, we would really be working with a ternary, not a binary alphabet. So once we have decided to encode all our data in binary, we should use only zeroes and ones and no commas. We already implicitly demanded this in Definition 1.2 where a code was defined as a mapping into $\mathbf{B}^+ = \{0,1\}^+$, not $\{0, 1, `,'\}^+$. Not all such comma-free codes are also prefix codes. However, one can show (see [30], Theorem 5.2.2, combined with Theorem 5.5.1) that, for every comma-free code $C : \mathbf{A}^+ \to \mathbf{B}^+$, there exists a prefix code C' such that for all $D \in \mathbf{A}^+$, $L_C(D) = L_{C'}(D)$. Since MDL is only concerned with code*lengths* rather than actual encodings, it follows that we may restrict ourselves to prefix codes without any loss of generality. Henceforth, whenever we use the word 'code', we therefore really mean a 'prefix code'. One can relate prefix codes to probability distributions, and this has some very important consequences. At this point, the reader who only wants to get a quick overview of MDL may wish to skip the technicalities below and only read about these consequences; they are summarized in Section 1.2.3.

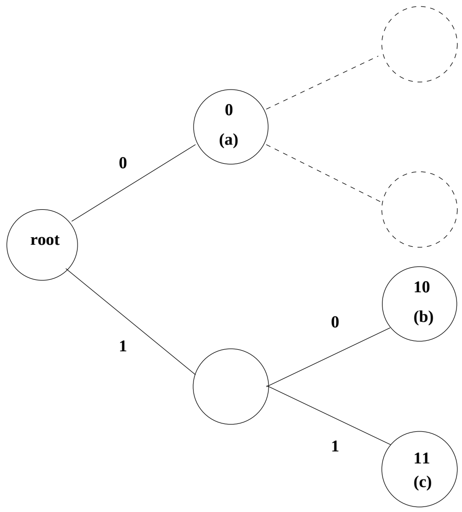


Figure 1.1: Binary code tree for the Kraft inequality using alphabet $\{a, b, c\}$ and code $C_1(a) = 0; C_1(b) = 10; C_1(c) = 11$

The Kraft Inequality

As was discussed on page 8, a crucial property of codes is the fact that each code for $\mathbf{E}$ can assign short code lengths only to very few elements in $\mathbf{E}$.

Probability distributions P over $\mathbf{E}$ have the property that $\sum_{x \in \mathbf{E}} P(x) = 1$. Hence, each P can only assign high probability to very few elements in $\mathbf{E}$.

There is an analogy here. Indeed, it turns out that short code lengths can be formally related to high probabilities. The exact relation is given by the *Kraft Inequality* [88]:

Theorem 1.6 (Kraft Inequality) *For any instantaneous code (prefix code) $C : \mathbf{A} \to \mathbf{B}^*$ for a finite alphabet $\mathbf{A} = \{1, \ldots, k\}$, the codeword lengths $L_C(1), \ldots, L_C(k)$ must satisfy the inequality*

$$\sum_{x \in \mathbf{A}} 2^{-L_C(x)} \leq 1. \tag{1.4}$$

Conversely, given a set of codeword lengths that satisfy this inequality, there exists an instantaneous code with these word lengths.

Proof: [adapted from [30]] We write l_{max} for the length of the longest codeword in the set of codewords: $l_{max} = \max_{x \in \mathbf{A}} L_C(x)$. We write l_i as an abbreviation of $L_C(i)$. Consider the full binary tree of depth l_{max}, where each branch is associated with either 0 or 1 (see Figure 1.1). In this tree, each codeword corresponds to a unique path starting at the root node: the first (leftmost) bit of the codeword determines which of the two children of the root node is visited; the second bit determines which of the two children of the first visited node is visited etc. The length of the path is equal to the length of the codeword. As an example, Figure 1.1 shows the paths for our example code C_1 of the previous subsection. With each codeword we associate the node at the end of the path that coincides with the codeword. For example, the node '10' is the node at the end of the path taking branch 1 at the root node and then taking branch 0.

We define the i-th *level* (denoted by LEVEL_i) of the tree to be the set of all nodes of the tree at depth i. The prefix condition on the codewords implies that no codeword is an ancestor of any other codeword on the tree. Hence, each codeword eliminates its descendants as possible codewords. The codeword i, together with its descendants, defines a subtree of possible extensions of the codeword up to depth l_{max}. If $l_i = l_{max}$, this subtree coincides with the node i itself. If $l_i = l_{max} - 1$, the subtree consists of the node i together with its two children, and so on. Let D_i be the set of leaves of the subtree belonging to i. We have $2^{-l_i} = |\text{D}_i| \cdot 2^{-l_{max}}$. Also $|\text{LEVEL}_{l_{max}}| 2^{-l_{max}} = 2^{l_{max}} 2^{-l_{max}} = 1$. Since the prefix property implies that all D_i are disjoint, we have

$$\sum_{i \in A} 2^{-L_C(i)} = \sum_{1 \leq i \leq m} 2^{-l_i} = \sum_{1 \leq i \leq m} |\text{D}_i| \cdot 2^{-l_{max}} \leq |\text{LEVEL}_{l_{max}}| 2^{-l_{max}} = 1$$

which implies the Kraft inequality (1.4).

Conversely, given any set of codeword lengths $l_1, \ldots, l_m$ satisfying the Kraft inequality, we can always construct a tree like the one in Figure 1.1. Label the first node (lexicographically) of depth l_1 as codeword of 1. Label the first node (lexicographically) that has depth l_2 and that is not a descendant from the node 1 as codeword of 2 etc. In this way, we construct a prefix code with the specified $l_1, \ldots, l_m$. $\Box$

Let $\mathbf{E}$ be finite and C be a code for $\mathbf{E}$. From the Kraft inequality we immediately see that there exists a probability distribution P defined over the space $\mathbf{E} \cup \{\Box\}$ such that for all $x \in \mathbf{E}$, $P(x) = 2^{-L_C(x)}$ (we need the extra symbol '$\Box$' to ensure that $\sum_x P(x) = 1$).

Similarly, let P be a probability distribution over the finite space $\mathbf{E}$. By the Kraft inequality (1.4) we see immediately that there exists a code C for $\mathbf{E}$ such that for all $x \in \mathbf{E}$: $L_C(x) = \lceil -\log P(x) \rceil$. This code is called the *Shannon-Fano code* [30].

We will ignore the effect of rounding up and drop the integer requirement for code lengths. Once we have done this, we obtain a *a correspondence between probability distributions and prefix codes*: to every P, there is a corresponding C such that the code lengths $L_C(x)$ are equal to $-\log P(x)$ for *all* $x \in \mathbf{E}$. At the same time, for every C there is a corresponding P such that the code lengths $L_C(x)$ are equal to $-\log P(x)$ for *all* $x \in \mathbf{E}$.

Since an analogue to the Kraft Inequality can be proven for countably infinite alphabets, the correspondence still holds for countably infinite $\mathbf{E}$ ([30]). We now consider the case where $\mathbf{E} \subset \mathbf{R}^k$. As first noted on page 7, in practice we will always be confronted with data that is recorded up to a finite precision d. That is, instead of using a probability distribution P with density function P defined over $\mathbf{E}$, in reality we use a discretized version P' of P, with probability mass function P', defined on a large but finite subset $\mathbf{E}'$ of $\mathbf{E}$. For this P' and all $x \in \mathbf{E}'$, we have $P'(x) = (1/K)P(x)$ for some constant K which depends on the precision that is being used. Using the code lengths of the code C' corresponding to P' we find that $L_{C'}(x) = -\log P(x) + \log K$ for all $x \in \mathbf{E}'$. Since $\log K$ does not depend on x, we may safely neglect it in our analyses (see [128] for details). We can then once more regard $-\log P(x)$ as an 'idealized' code length.

1.2.3 Identifying Codes and Probability Distributions

The above can be summarized as follows:

Correspondence between Probability Distributions and Prefix Codes
Let $\mathbf{E}$ be a sample space. Let P be a probability distribution over $\mathbf{E}^n$, the set of sequences of length n. Then there exists a prefix code C for $\mathbf{E}^n$ such that for all $x^n \in \mathbf{E}^n$, $L_C(x^n) = -\log P(x^n)$. C is called the *code corresponding to* P.
Similarly, let C' be a prefix code for $\mathbf{E}^n$. Then there exists a probability distribution P' such that for all $x^n \in \mathbf{E}^n$, $-\log P'(x^n) = L_{C'}(x^n)$. P' is called the *probability distribution corresponding to* C'.

The correspondence allows us to *identify* codes and probability distributions, such that a short code length corresponds to a high probability and vice versa.

In this correspondence, probability distributions are treated as mathematical objects and *nothing else*. If we use a code C to encode our data, this definitely does *not* necessarily mean that we assume our data is drawn according to the probability distribution corresponding to C.

The identification of probability distributions and codes allows for a probabilistic reinterpretation of the *minimum description length* principle as a *maximum probability* principle, since short code lengths correspond to high probabilities and vice versa. This reinterpretation will be worked out in the sections to come.

1.3 Formalizing the Two-Part Code

Let $\mathcal{M}$ be some class of models and let D be a given data sequence. As we saw in Section 1.1.4, the (two-part code) MDL Principle tells us to look for the hypothesis $H \in \mathcal{M}$ that minimizes the sum of the description length of H plus the description length of the data when encoded with the help of H. This means that two different codes are involved: a code C_1 to encode our hypotheses, and a code C_2 to encode our data D using the hypothesis. In mathematical notation, the MDL Principle then becomes: *we should pick the* $H_{mdl} \in \mathcal{M}$ *with*

$$H_{\text{mdl}} = \arg\min_{H \in \mathcal{M}} \{L_{C_2}(D|H) + L_{C_1}(H)\} \tag{1.5}$$

Here the notation $\arg\min f(x)$ denotes the x minimizing $f(x)$. If there is more than one $H \in \mathcal{M}$ for which $L_{C_2}(D|H)+L_{C_1}(H)$ is minimized, we pick the one with minimum hypothesis complexity $L_{C_1}(H)$. If this still leaves us with several possible H, we do not have a further preference among them. For each different H, C_2 will encode the data D in a different way: it encodes it 'with the help of H'. Therefore, C_2 is not really one single code but rather a function that maps each $H \in \mathcal{M}$ to a different code $C_2(\cdot|H)$. The notation $L_{C_2}(D|H)$ in Equation. (1.5) is used to denote the length of encoding $C_2(D|H)$.

The single code over data sequences D that results from coding each D using the H_{mdl} that is optimal for this specific D is called the *two-part MDL-code*. We denote the code lengths resulting from this code by $L_{\textit{2-p}}(\cdot|\mathcal{M})$:

$$L_{\textit{2-p}}(D|\mathcal{M}) = L_{C_2}(D|H_{\text{mdl}}) + L_{C_1}(H_{\text{mdl}}) \tag{1.6}$$

where H_{mdl} depends on D and is given by (1.5).

1.3.1 Two-Part Codes for Probabilistic Model Classes

Most of traditional statistics is concerned with classes consisting of probabilistic models. We assume such classes to be parameterized by a vector θ coming from some set of possible parameter vectors Γ. The class of models can then be written as

$$\mathcal{M} = \{P(\cdot|\theta) \mid \theta \in \Gamma\}$$

where the $P(\cdot|\theta)$ are all probabilistic models for the same sample space $\mathbf{E}$. $P(\cdot|\theta)$ can be read as 'the probability of the data given that the model used is θ'; examples will be given below. Notice that, strictly speaking, θ does not denote a model but a *name* for a model. This difference will become relevant in later chapters. For now, we simply use H to denote non-probabilistic models and θ to denote probabilistic models and write, with some slight abuse of notation, $\theta \in \mathcal{M}$ instead of $\theta \in \Gamma$.

By the remarks in Section 1.2.3, we know that there exists a code C such that $-\log P(D|\theta) = L_C(D)$ for all D of length n. It is this code we will use for C_2: for every $\theta \in \mathcal{M}$, we get $L_{C_2}(D|\theta) = -\log P(D|\theta)$ for all D. Notice that this choice for C_2 gives a short code length to sequences which have high probability according to θ while it gives a high code length to sequences with low probability. The code length thus reflects the goodness-of-fit of the data with respect to θ, since the higher the probability of D according to θ, the better θ fits D. As explained at the beginning of this section, our aim is a trade-off between goodness-of-fit and complexity. This makes the choice of C_2 such that $L_{C_2}(D|\theta) = -\log P(D|\theta)$ a natural one. Whether this is really the only reasonable possibility for C_2 is a subtle issue. It will be discussed further in Chapter 2 (Sections 2.2) and then, at length, in Chapter 5, Section 5.4). For now, we assume C_2 is defined as indicated above. The MDL Principle for probabilistic model classes thus tells us to pick the following model θ_{mdl} when given data D:

$$\theta_{\text{mdl}} = \arg\min_{\theta \in \mathcal{M}} \{-\log P(D|\theta) + L_{C_1}(\theta)\} \tag{1.7}$$

where C_1 still has to be specified.

MDL and ML In the well-known method of *Maximum Likelihood (ML) Estimation* (see any introductory textbook on statistics, for example [6]), we estimate a probability distribution from a given set of data D by taking the probabilistic model θ that maximizes the probability of D. We denote this θ by $\hat{\theta}$. Sometimes we write $\hat{\theta}(D)$ rather than $\hat{\theta}$, in order to stress that $\hat{\theta}$ actually depends on D. If we pick our models from model class $\mathcal{M}$, we can write

$$\hat{\theta}(D) = \hat{\theta} = \arg\max_{\theta \in \mathcal{M}} P(D|\theta) = \arg\min_{\theta \in \mathcal{M}} \{-\log P(D|\theta)\} \tag{1.8}$$

where the last equality follows from the fact that log and '$-$' are monotonous transformations. Combining (1.7) and (1.8), we see that the ML estimator $\hat{\theta}$ can be seen as an approximation of the *MDL estimator* θ_{mdl}: the MDL estimator becomes the ML estimator $\hat{\theta}$ if we neglect the complexity term $L_{C_1}(\theta)$.

Example 1.1 (continued) We consider the class of *Bernoulli models* for the binary sequences given in Example 1.1 on page 5. Each model is identified with a parameter $\theta \in [0, 1]$:

Definition 1.7 *Let* $\mathbf{E} = \{0, 1\}$. *The Bernoulli model* $P(\cdot|\theta)$ *is defined as follows: for all* n, *all* $x^n \in \mathbf{E}^n$, $P(X_{n+1} = 1|X^n = x^n) = P(X_1 = 1) = \theta$.

In other words, we regard the data as being generated by independent tosses of a biased coin with probability θ of coming up 'heads' (which we identify with '1'). Let us consider sequence (1.3) of Section 1.1 again. This sequence contains four times as many 0s than 1s. Clearly, the ML estimator for this sequence is $\hat{\theta} = 1/5$. In general, for a sequence D of length n with a fraction of γ 1s, we have $\hat{\theta}(D) = \gamma$: the ML Bernoulli model coincides with the frequency of 1s. We have:

$$-\log P(D|\hat{\theta}(D)) = -\log \gamma^{\gamma \cdot n}(1-\gamma)^{(1-\gamma)\cdot n} \approx \log \binom{n}{\gamma n}. \tag{1.9}$$

where the rightmost (approximate) equality will be shown to hold in Chapter 3, Section 3.5, page 59. In Section 1.1, we suggested to code sequence (1.3) by first stating γ, the frequency of 1s in the data string D, and then giving its index in the lexicographical ordering of all sequences with the same fraction γ of 1s. We saw that the second part of this code would take approximately $\log \binom{n}{\gamma n}$ bits. We now see that if we code the data D using the code corresponding to a Bernoulli model, we are really doing something similar: we first, using code C_1, encode the number $\hat{\theta}$, which corresponds to the frequency of 1s in D. Then we code the data 'with the help of' $\hat{\theta}$, which takes approximately $\log \binom{n}{\hat{\theta}\cdot n}$ bits, about the same amount of bits as it would take to specify the index in the ordering referred to above.

However, there is also a big difference between coding by giving an index and coding using the code corresponding to a distribution θ: in the second case, we can also encode data strings D' for which the frequency of 1s is *not* equal to θ, and this will take us $-\log P(D'|\theta)$ bits. To fully describe the data D, we also need to describe the parameter θ_{mdl} itself. θ_{mdl} is *not* necessarily equal to $\hat{\theta}$. To encode $\hat{\theta}$, we need to work with a fairly high precision. Often, there is another $\theta \in \mathcal{M}$ for which $-\log P(D|\theta)$ is just a little lower but which can be encoded using much lower precision. For such a θ, the sum (1.7) will be lower than for $\hat{\theta}$. We now show how to encode values of θ.

The Code C_1 - Descriptions as Messages To explain the code C_1, it will be useful to interpret descriptions as messages. We can always think of a description as a message that some sender or *encoder*, say Mr. A, sends to some receiver or *decoder*, say Mr. B. Before sending any messages, Mr. A. and Mr. B. meet in person. They agree upon a description method that will be used by A to send his messages to B. Once this has been done, A and B go back to their respective homes and A sends his messages to B in the form of binary strings. The fact that a description method must be one-to-many (page 7) implies that, when B receives a message, he should always be able to decode it in a unique manner.

In the 2-part coding scheme, A first sends a parameter value $\theta \in \mathcal{M}$ to B; only then does he send an encoding of the data D. If $\mathcal{M}$ contains only one single model θ_1, then B knows beforehand what A wants to send. So A does not have to send anything and the description of θ_1 takes 0 bits.

If we consider a model class containing a finite number of elements k, then there are k different possible messages. Hence A and B must agree on a description method that assigns different encodings to all these k messages. If $k = 2^d$ for some d, then there exists a description method for doing this that assigns code length d to all k possibilities. Similarly, if, as in our Bernoulli example, we want to describe a parameter $\theta \in \mathbf{R}$, we have to truncate θ to a finite precision. If we use a fixed precision d, then we need d bits to send one of the 2^d possible truncated parameter values.

Since, for every $\theta \in \mathcal{M}$ ($\mathcal{M}$ still denoting the Bernoulli models), we would like to be able to describe parameter values that are as close to θ as we want, we need to work with a precision that is *not* fixed. Therefore, when sending a value θ_d, we first need to encode the particular value d with which θ will be encoded, otherwise a decoder would not know how many bits the description of θ would take. We have to encode d using a prefix code, otherwise the decoder cannot decide once more at what point he has finished decoding the number d and should start decoding the value of θ itself. To see how to code d, first consider a primitive prefix code which encodes the natural numbers $n > 0$ as follows: the number n is encoded as $0^{n-1}1$, i.e. $(n-1)$ 0s followed by a 1. We call this the *trivial* code for n. Clearly, this code has the prefix property. We can turn this code into a more efficient one that is still prefix as follows: we first encode, in the 'trivial' way, the *length* that the description of n would have if it were written in binary in the standard (non-prefix) manner (that is, we write '1' as 0, '2' as 1, '3' as 10 and so on). This length is $\lceil \log n \rceil$. We let this first encoding be followed by the encoding of n in this standard non-prefix manner. This 2-part message will take $2\lceil \log n \rceil$ bits. A decoder can now first decode the length of n; he then knows how many bits the coding of n will take so he can start decoding it and will know exactly when he is done decoding it: the resulting code still has the prefix property. Hence the number n is encoded using $2 \cdot \lceil \log n \rceil$ bits. But we can also repeat the trick, and first encode *the length of the length* of n in the trivial manner, and then encode first the length of n and then n itself in the non-prefix free manner. Again, the resulting code will be prefix free. In this way, we need $2 \cdot \lceil \log \lceil \log n \rceil \rceil + \lceil \log n \rceil \approx 2 \log \log n + \log n$ bits.

If we encode the precision d in this way, the decoder can first decode d. He then knows that d more bits follow for the parameter value itself, so he also knows at what point he will be done decoding θ and can start decoding the data using θ: the resulting code C_1 takes

$$L_{C_1}(\theta) \approx d + \log d + 2 \log \log d \text{ bits.} \tag{1.10}$$

One can show that asymptotically, the optimal precision d for encoding a sample of size n is given by

$$d(n) = \frac{1}{2} \log n + c \tag{1.11}$$

where c is a constant that can also be computed ([128], page 57). So, alternatively, in our encoder-decoder setting, sender and receiver may opt for a code in which the precision is given by (1.11). Then the receiver always knows what precision the sender will use in coding the value θ and can uniquely decode it (we assume here that the sample size n is known to both sender and receiver in advance). In this way, the code length $L_{C_1}(\theta)$ becomes equal to $d(n)$ as given in (1.11): coding the parameter θ takes $1/2 \log n + c$ bits, while d itself does not have to be encoded any more, so the $\log d + 2 \log \log d$ term disappears. For large n, (1.11) will lead to a slightly shorter code length than the code with lengths (1.10).

In this case of a model class with a single parameter, $L_{C_1}(\theta)$ (growing logarithmically in n) will in general be much smaller than $L_{C_2}(D|\theta)$ which clearly grows linearly in n for all $0 < \theta < 1$. Hence, neglecting $L_{C_1}(\theta)$ will not lead to much trouble. This is in sharp contrast with the many-parameter setting, as shown in the next example.

There is one important detail in the coding procedure that we have glossed over so far: how do we map parameters to their encoded values? If we use precision $d = 2$, we can encode four possible parameter values $(\theta_{00}, \theta_{01}, \theta_{10}, \theta_{11})$. Which four should we take? $(0, 1/3, 2/3, 1)$? Or $(1/5, 2/5, 3/5, 4/5)$? Or something else? This tricky question has been given a full answer only very recently [129]; see also Chapter 7. In most practical situations, codes with equal spacing between the parameter values will work reasonably well.

1.4 Two-Part Codes for Non-Probabilistic Model Classes; Supervised Learning

We now discuss how to construct two-part codes for non-probabilistic model classes. As in the probabilistic case, we must first construct the code C_2 and then the code C_1, where C_1 and C_2 are used as in Equation 1.6. We show how to do this using our polynomial example:

Example 1.3 (continued) Let the model class $\mathcal{M}$ consist of the class of all polynomials. We seek a polynomial H that provides a good trade-off between complexity and goodness-of-fit as measured by the total squared error $\sum_{1 \leq i \leq n} (H(x_i) - y_i)^2$. In Section 1.5 (and more completely in Chapter 5, Section 5.3.5) we discuss why it makes sense to take the squared error function here.

Supervised Learning The code C_1 will be used to encode polynomials; the code C_2 will be used to encode data $D = (x^n, y^n)$ with the help of a polynomial H. Since we are only interested in finding out how the y_i *depend* on the x_i, we may regard the x_i as given. Indeed, after one has selected a polynomial H on the basis of (x^n, y^n) and one is asked to use it to predict future data, one will usually also be *given* some new value of x and one will then predict y as $H(x)$. Learning functions $H : \mathbf{E}_x \to \mathbf{E}_y$ with the goal of predicting y-values on the basis of $H(x)$ is called *supervised learning* in the machine learning/neural network literature [72]; it corresponds to *regression* in statistics [6].

Regarding the x_i as given means that we do not have to encode them[4]. In the encoder-decoder setting introduced in the previous section, this corresponds to the situation that $x_1, \ldots, x_n$ are known both to encoder and decoder.

The Code C_2 In the case of probabilistic model classes, we turned each model into a code such that high probability of the data corresponded to a short code length of the data; our goal in doing this was to get the code length of data D to reflect the goodness-of-fit of D with respect to the hypothesis: the higher the probability of the data according to a hypothesis, the better the hypothesis *fits* the data. In the present (polynomial) example, goodness-of-fit is defined in terms of squared errors: polynomial H gives a better fit on the set of points (x_i, y_i) than on the set of points (x_i, z_i) iff $\text{ER}_{sq}(y^n|H, x^n) < \text{ER}_{sq}(z^n|H, x^n)$ where ER_{sq} is the total squared error: $\text{ER}_{sq}(y^n|H, x^n) = \sum_{i=1}^n (y_i - H(x_i))^2$. In analogy to the case of probabilistic models, we would like the description length of the data $y_1, \ldots, y_n$ given hypothesis H and $x_1, \ldots, x_n$ to reflect how well H fits data (x^n, y^n): the shorter the code length, the better the fit. This will be achieved if we can manage to construct a code C_2 such that for all $D = ((x_1, y_1), \ldots, (x_n, y_n))$ and all H:

$$L_{C_2}(y_1, \ldots, y_n | H, x_1, \ldots, x_n) = \text{ER}_{sq}(y^n|H, x^n) + K \tag{1.12}$$

where K is a constant that may depend on n but *not* on H or any of the y_i. To see why a code with lengths given by (1.12) is what we want, note that (1.12) implies for all $(y_1, \ldots, y_n)$ and all $(z_1, \ldots, z_n)$ that

$$\begin{array}{rcl} \Delta L_{C_2} & = & L_{C_2}(z^n|H, x^n) - L_{C_2}(y^n|H, x^n) \quad = \\ = \Delta \text{ER}_{sq} & = & \text{ER}_{sq}(z^n|H, x^n) - \text{ER}_{sq}(y^n|H, x^n) \end{array} \tag{1.13}$$

which means that the difference in goodness-of-fit between every two datasets is precisely reflected in the difference in code length. It turns out that we can indeed construct such a code C_2. The reason is that for each $H \in \mathcal{M}$ there exists a probability distribution $P(\cdot|H)$ such that, for all (x^n, y^n),

$$-\log P(y^n|H, x^n) = \text{ER}_{sq}(y^n|H, x^n) + K \tag{1.14}$$

We know from Section 1.2.3 that the existence of $P(\cdot|H, x^n)$ implies the existence of a code $C(\cdot|H)$ with code lengths given by $L_C(y^n|H, x^n) = \text{ER}_{sq}(y^n|H, x^n) + K$. Below

[4]Encoding the x_i after all does not change anything really essential though, as will be shown at the end of this section.

we verify that the probability distribution P which achieves this for us is simply the n-fold product distribution of a conditional

Gaussian (normal) distribution. More precisely, it has density function f given by:

$$\begin{aligned} f(y^n|H, x^n) &= \prod_{i=1}^{n} f(y_i|H, x_i) \text{ where} \\ f(y|H, x) &= \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y - H(x))^2}{2\sigma^2}) \text{ with variance } \sigma^2 = (2 \ln 2)^{-1}. \end{aligned} \tag{1.15}$$

Hence, conditioned on x_i the y_i are independently normally distributed with mean $\mu_i = H(x_i)$ and constant variance σ^2. We can now verify at once that (1.14) holds:

$$- \log f(y^n|H, x^n) = - \sum_{i=1}^{n} \log f(y_i|H, x_i) = \text{ER}_{sq}(y^n|H, x^n) + K$$

where both equalities follow by plugging in the expressions for $f(y_i|H, x_i)$ given in (1.15).

The Code C_1 and the Sum Encoding a polynomial $H_{k,d}$ of the $(k-1)$-st degree up to d bits precision per parameter is done by a straightforward extension of the procedure used for Bernoulli models: we first code d and k in a prefix-free manner, and then we encode the k parameter values using kd bits. We omit the details. Combining the resulting L_{C_1} with L_{C_2} as given by (1.12), the sum of the two description lengths becomes:

$$\begin{aligned} & L(y^n; H_{k,d}|x^n) = \\ & \overbrace{\sum_{i=1}^{n} (y_i - H_{k,d}(x_i))^2}^{\text{error term}} + \overbrace{2(\log \log k + \log \log d) + \log kd + kd}^{\text{complexity term}} + K \end{aligned} \tag{1.16}$$

Here K is a term that does not depend on $H_{k,d}$ and that therefore does not play any role when finding the $H_{k,d}$ that minimizes (1.16) (but see below!). According to the MDL Principle, we should now prefer the $H_{k,d}$ for which (1.16) is minimized. This is the optimal trade-off between the error term $(y_i - H_{k,d}(x_i))^2$ and the complexity terms involving k and d. The overall optimum depends on both the *degree* of the polynomial and the *precision* with which to encode the parameter values.

If we want to optimize the behaviour of MDL, the constant K becomes important after all. Matters get a lot more complicated once we start involving K, and we defer their discussion to Chapter 5, Section 5.2.

Let $\alpha_1, \ldots, \alpha_k$ be the parameters of a $(k-1)$-st degree polynomial. Just like in the

Bernoulli example, it can be shown that asymptotically, the optimal precision for each parameter α_i is given by $d_i = 1/2 \log n + c_i$ for some constant c_i ([128], page 56). Using this precision (which can be derived from the sample size n) the value d does not have to be encoded any more and (1.16) becomes:

$$L(y^n; H_{k,d}|x^n) = \sum_{i=1}^{n} (y_i - H_{k,d}(x_i))^2 + \tfrac{k}{2} \log n + 2 \log\log k + \log k + K' \tag{1.17}$$

for some constant K'.

When x_i are not regarded as given The MDL Principle tells us that we should find a short description for *all* our data (Section 1.1.1). One may argue that, since the data include x^n, we should also describe x^n which would lead to a code length larger than (1.16). However, this will still lead us to pick the same optimal $H_{k,d}$ as shown by the following argument: suppose we first encode the x_i using some code C_3. We then code a polynomial H using code C_1, followed by an encoding of the y_i based on the code C_2. We then get as a total code length:

$$L(x^n, y^n; H_{k,d}) = L(y^n; H_{k,d}|x^n) + L_{C_3}(x^n) \tag{1.18}$$

where $L(y^n; H_{k,d}|x^n)$ is as in (1.16). Clearly, for every fixed D, the $H_{k,d}$ minimizing (1.18) coincides with the $H_{k,d}$ minimizing (1.16).

***Semi*-Probabilistic Interpretation of Non-probabilistic Model Classes**

The squared error function led to a specific code C_2. We have seen that this code coincided with the code corresponding to a distribution that makes the y_i independently distributed around $H(x_i)$ with Gaussian noise of a specific variance. It *seems* as if using the squared error function implies an implicit assumption of Gaussian noise. But this is not true: at this point we should stress once again that this probability distribution is just a means of describing, or equivalently *modeling* our data. We discuss in the next section (and prove in Chapter 5) that even if the data is *generated* (drawn) according to a completely different distribution, it may still be useful and harmless to *model* it using a normal distribution.

1.4.1 MDL and the model that best fits the data

It can be shown [128] that an analogue to Equation 1.17 holds in great generality; for all sufficiently regular model classes the optimal number of bits needed to encode the parameters is $1/2 \log n$ for each parameter. If we use a class $\mathcal{M}_k$ containing only models with a fixed number of parameters k (for example, the class of $(k-1)$-st degree polynomials), then the number k does not have to be encoded, and we get:

$$L(y^n; \theta_d|x^n) = -\log P(y^n|\theta_d, x^n) + \frac{k}{2} \log n + O(1) \tag{1.19}$$

This formula can be used both for the case of probabilistic and non-probabilistic model classes; for, as we shall see in the next chapter, the trick we applied to the squared error to turn our hypotheses into codes, and hence probability distributions, can be employed in general.

From (1.19) we see that for a class with a fixed number of parameters, as n increases, the model in $\mathcal{M}$ that maximizes the likelihood and the model θ_{mdl} in $\mathcal{M}$ that minimizes the description length (1.19) *converge*: for increasing n, the precision with which parameters θ_d are encoded becomes higher and higher, while the complexity term $k/2 \log n$ is independent of θ_d if k is fixed. Therefore, θ_{mdl} and $\hat{\theta}$ must become closer and closer.

However, if k is not fixed (for example, when the model class of *all* polynomials is used), and when the data comes from a noisy source, the model in $\mathcal{M}$ that maximizes the likelihood or, if a non-probabilistic model class is used, minimizes the error will generally 'jump around' as the size of the data increases: one can show that if we have k data points, we will generally select a k-th degree polynomial (as, for example, in Figure 1.2 on page x).

In such a case though, the MDL estimator will keep choosing simple polynomials. One can show that if the data are truely 'generated' by some polynomial H^* of degree k with Gaussian noise, then the MDL estimator will generally, once the size of the data set becomes large enough, converge to the 'true' polynomial H^* with probability 1 [14]. However, if the true degree k is high, then for small samples, MDL will usually 'underfit': it will pick a polynomial of degree lower than k. This can be seen from Equation 1.17: the lower error term for the high degree polynomial does not yet outweigh its higher complexity term. In the next section we will consider the question of whether such a provisional, overly simple model can still be useful.

1.5 MDL is Looking for a Good, not for a True Model

indexreliable!prediction We just saw that, if not enough data is available, MDL will sometimes pick a model that is 'too simple'. It is important to realize that this does *not* mean that MDL assumes that 'simple models are a priori more likely to be true', or that 'nature prefers simplicity' or anything like that: the rationale behind picking only simple models when few data is available is rather that *the data set is too small to identify a complex model with any reliability.* This is nicely illustrated for *very* complex models in Figure 1.2 on page x: if the model class $\mathcal{M}$ contains just as many parameters as your sample size n, then for each possible realization of the data you will find a model with a perfect fit to the data; if the data is generated by a process that is prone to small random fluctuations, such a model clearly does not give you any information about the data: the probability that you will hit upon the right model is about 0. On the other hand, if you pick the model that best fits the data within a small (simple) enough model class then the model you pick will typically be close to the model *within your class* that will give the best predictions of future data (in Chapter 2, Section 2.7 we make this more precise and consider how the error you make when predicting on the basis of a model inferred from a k-dimensional class depends on k).

The fact that MDL may pick an overly simple model if only few data are available raises the question whether we can 'provisionally' use the overly simple model to make reasonable predictions of future data. We give an example:

Example 1.8 [Bernoulli and Markov] Suppose you are given a sequence of 0s and 1s where the 0s and 1s are not really independent. You are not aware of this fact, and you decide to model your data using the Bernoulli model class (Definition 1.7) which treats the data items as if they were independent.

To make the example concrete, suppose the data are really distributed according to a first-order Markov chain [47]: a distribution such that the probability of 1 at the $(i+1)$-st outcome depends on the i-th outcome. Let us assume that the 'true' Markov chain P^* is given by

$$P^*(X_{i+1} = 1 | X_i = 1) = \frac{2}{3} \quad ; \quad P^*(X_{i+1} = 1 | X_i = 0) = \frac{1}{6} \tag{1.20}$$

It is easy to show that under such a distribution the frequency of 1s in a sequence $x_1, x_2, \ldots$ will, as its length n increases, tend to $1/3$ with probability 1.

Therefore, if enough data is available and you model the data using the Bernoulli model class, your optimal model $\theta_{\mathrm{mdl}}(x^n)$ will converge to $1/3$ with probability 1 (recall that the MDL and the maximum likelihood estimator converge). On the other hand, if data were truely distributed according to a Bernoulli distribution with $P(X=1) = 1/3$, then your model would also, with probability 1, converge to $\theta_{\mathrm{mdl}}(x^n) = 1/3$: in both cases you would end up with the same optimal model, which, also in both cases, you would consider to be a reasonable model for your data since it allows for considerable compression: you would need approximately

$$-\frac{n}{3} \log \frac{1}{3} - \frac{2n}{3} \log \frac{2}{3} + O(\log n) \tag{1.21}$$

bits to code the data[5] (see page 18), which is smaller (by a linear amount) than n, the number of bits you would have needed if you had encoded the data literally.

However, in the case where data are distributed according to the Markov chain, you would be able to achieve a lot of additional compression if you had used the model class consisting of all Markov chains rather than the Bernoulli model class. One can show [30] that with probability 1, coding the data using the code based on the 'true' Markov chain will allow for additional (by an amount linear in n) compression of the data. The first-order Markov chain itself can be encoded by two parameters. Again, the optimal precision for doing this will be on the order of $1/2 \log n$ per parameter, which means that as n gets larger, the total two-part codelength based on the class of Markov chains will be smaller than the two-part code length based on the Bernoulli model class by an amount linear in n.

In practice one will often use a model class $\mathcal{M}_1$ and not be aware that there exists another model class $\mathcal{M}_2$ with which one can achieve much more compression. The

[5]Equation 1.21 needs some additional justification since both the model $\theta_{\mathrm{mdl}}(x^n)$ and the frequency of 1's will, for finite n, not be completely equal to $1/3$. However, using results from [26] it is not hard to show that asymptotically, (1.21) will indeed hold with probability one.

only way to avoid this would be to use the class of all computer programs relative to some universal computer language; but we have already seen that we cannot do that in practice. If one uses a more restricted class, it can always happen that one does not capture all regularity in the data (Section 1.1.3): if one uses the Bernoulli model class, it may be that the class of all Markov chains provides a better (more compressing) model[6]. If one uses the class of all Markov chains, then it may be that some completely different model class achieves more compression. One can of course 'lump together' several different model classes into one big model class, but in general, it will simply be unknown what the optimal model class is for the data at hand and one will use a suboptimal one.

Such a suboptimal model class will contains models *that allow for substantial compression of the data but that are (a) not in any respect 'true' and (b) lead to much less compression of the data than a shortest computer program that prints the data and then halts.*

According to Rissanen's MDL Philosophy [128], this is the situation we will typically be in - and we agree. In fact, in every application of MDL we know of, the model class is immediately seen to be not completely correct or even clearly incorrect. A concrete example of this can be found in part II of this thesis, where we use the very simple 'naive Bayes' model class to model economic, biological and medical data, which obviously has not been generated by any 'naive Bayes' model. Nevertheless, the naive Bayes model allows us to discern regularities in the data which can be used to predict and classify future data with surprisingly high accuracy.

The MDL philosophy even goes one step further and says that there is no such thing as a 'true model' at all: all our models will always be wrong to some extent. Nevertheless, it is of interest to study the hypothetical situation where some 'true model' exists but is quite different from any of the models in our class. If we can show that MDL is well-behaved in such a situation, then this can serve as an indication that it will work well in practical settings. In this thesis we will study the situation where data are distributed according to some unknown P^* that is not necessarily related to any of the models in the model class $\mathcal{M}$ used (the models in $\mathcal{M}$ may even be non-probabilistic). Such a situation has been studied before [14, 167]. It is relatively easy to show that, under quite general conditions, the MDL estimator $\theta_{\mathrm{mdl}}(x^n)$ will converge to the model $\tilde{\theta}$ in $\mathcal{M}$ that in some precisely defined sense is closest to P^*.

However, one may ask what will happen if one uses such a model $\tilde{\theta}$ - which is partially wrong - instead of P^* for *predicting* aspects of future data. It is this question which we will study further in chapters 4 and 5. Roughly, in those chapters we show that a model that is quite different from the 'true' P^* can nevertheless be 'safe' to use in the sense that it gives you a correct impression of how well you can predict future data. Let us assume once more, as in Example 1.8, that data are generated by a Markov chain but modeled by the Bernoulli class. If we used the model $\theta_{\mathrm{mdl}}(x^n) = 1/3$ to predict the value of the next bit, we would predict that a 0 will occur rather than a 1. Our model tells us that this will be right in about 2/3 of the cases. If future data

[6]Of course there are also cases where one has additional reasons for assuming that the data are independent (for example, they are outcomes of a repeatable experiment). But generally speaking it is nearly always the case that there may exist some alternative model class for the data at hand that is better than the one one is using.

will still be distributed according to the Markov chain, then this will give a correct impression of the frequency of mistakes we will make. As is clear from (1.20), if we use the 'true' Markov chain instead for predicting future data, we will predict a 1 if our last seen outcome was a 1 and a 0 if it was a 0. This will lead to a much lower frequency of error in predicting - the complex model is better. However, using the simpler Bernoulli model we never *expected* to predict better than with accuracy 1/3. Therefore, though the model is too simple, it is so in a relatively harmless way, since the model itself *tells* us that it is not very accurate: its *average* performance over future data will be reasonably close to its *expected* performance over future data. In other words, when using the overly simple Bernoulli model for the data generated by the Markov chain, *you do not know very much about the data generating process, but you* know *that you do not know very much.*

It turns out (Chapter 5) that a similar statement holds in general: the optimal model in $\mathcal{M}$ for present data will give a correct impression of the error you will make if you use it to predict future data – provided present and future data are generated by the same probability distribution. However, this distribution need not be identical or even 'close' to any of the models in $\mathcal{M}$.

Let us give another example, involving once again the class of polynomials. We assume throughout this example that data are actually sampled independently from the third-degree polynomial depicted in Figure 1 with some noise (not necessarily Gaussian), and that we model the data using only the class of first degree polynomials. In that case, the optimal two-part code MDL model as well as the model maximizing the fit to the data will, with probability 1, converge to the same first-degree polynomial $\widetilde{H}$ (this will be shown in Chapter 5, Lemma 5.14). This polynomial will look like the one depicted in Figure 1.1. It is clearly too simple; if we look at the average error per data item (x_i, y_i), we see that it is quite high; the third degree polynomial would be a better model for the data at hand. Nevertheless, as we will show in Chapter 5 (Theorem 5.19), if we use $\widetilde{H}$ as a basis of predicting future data, then, with high probability, we will make approximately the same error on average as we have made on the data D we have seen so far: the error we will make in predicting future data is close to the error we would *expect* to make *if our model $\widetilde{H}$ were actually true*. Hence, also in this case, using the overly simple model is 'safe' in that the model does not promise you more than you will get.

Modeling versus Generating Distributions Model classes that do not contain the 'true' probabilistic model are usually called 'misspecified'. We show in Chapter 5 that under a wide variety of probabilistic underlying laws, basing estimates on a 'misspecified' model class will be 'safe' in the sense above. Also, it turns out (Chapter 5, Theorem 5.19) that if we use the squared error function and hence model the data *as if* it were subject to Gaussian noise, we will still obtain good, 'safe' (in the sense above) results even if the data *generating* distribution is not Gaussian at all!

We obtain this result (which in itself is not new) as a special case. It partially explains why modeling errors using a Gaussian distribution generally leads to good results. People have often wondered how this is possible, since when we inspect observational data, we find more often than not that the empirical distribution of the errors is not really Gaussian. We quote from Jaynes' forthcoming book ([77], Chapter 7):

> "In the middle 1950's the writer heard an after-dinner speech by Professor Willy Feller (writer of the classic [47], PG) , in which he roundly denounced the practice of using Gaussian *probability* distributions for errors, on the grounds that the *frequency* distributions of real errors are almost never Gaussian. Yet in spite of Feller's disapproval, we continued to use them, and their ubiquitous success in parameter estimation continued."

By always keeping in mind that our *modeling* probabilities are monotone transforms of code lengths and not frequencies, we see that Feller's reasoning is not necessarily correct.

What does a model say about the world? Once we acknowledge that our models always have a chance of being partially wrong, a new difficulty arises, for it follows that *we should not just act as if our model were true in all respects*. Let us return to the example where we assume data to be generated by the Markov chain (1.20) and we inferred the Bernoulli model $\theta_{\text{mdl}}(x^n) = 1/3$. According to our Bernoulli model, which models the data as being independently distributed, the frequency of '11' over future data will be $1/3 \cdot 1/3 = 1/9$. But this will not be the case (neither in the given sample nor in future data): an easy calculation shows that it tends to $2/9$ instead. So the conclusion that the frequency of 1's in future data will be approximately $1/3$ seems more 'safe' than the conclusion that the frequency of '11' will be approximately $1/9$: it is less sensitive to whether or not our model is really 'true' in any sense. In general, it may be unwise to draw just *every* conclusion which would follow if an inferred model for data really were 'true' and *would* generate the data. You could, of course, for each new conclusion you would like to draw from your model inspect the given data D on which your model is based to see whether it really holds for D, but then you would not be able to use your model as a 'stand-alone' which is what you usually want. Apparently, some conclusions about the model are much more sensitive to the model 'really being true' than others. This raises the important question of what can and what cannot be reliably inferred from a statistical model for the data: what exactly does a model say about the modeled situation? We deal with this question at length in Chapters 4 and 5; but in order to do so, we need to introduce the fundamental concept of MDL, the *stochastic complexity*. This will be the subject of the next chapter.

Chapter 2

Stochastic Complexity

In this chapter we introduce the fundamental concept of MDL: the *stochastic complexity*. We start by motivating the need for the concept. In Section 2.2 we prepare the formal definition of stochastic complexity by arguing that *every* model together with every reasonable error function can be re-interpreted as a probabilistic model or, equivalently, a description method.

In Section 2.3 the stochastic complexity is introduced formally. This is followed by discussions of various interpretations and applications of stochastic complexity. Section 2.7 provides an alternative derivation of stochastic complexity by reinterpreting the whole theory in terms of proportional betting. Stochastic complexity is closely related to the 'marginal distribution' of Bayesian statistics. The relation is discussed in some detail in Section 2.8. The chapter ends with a short summary.

2.1 Motivation

We can distinguish between the general MDL Principle and its instantiations. The general MDL Principle tells us the following: the more we can compress our data, the better we have captured the regular features of the data and hence the more we have learned about it. In Chapter 1 we instantiated this to the two-part code MDL Principle given on page 11, where a particular code $C_{2\text{-}p}$ was defined relative to a model class $\mathcal{M}$ so as to allow compression of those data sequences that are well-described by one of the models in $\mathcal{M}$.

Let $\mathbf{E}$ be a sample space and let C and C' be codes over $\mathbf{E}$. We call C' *more efficient* than C if for all $D \in \mathbf{E}$, $L_{C'}(D) \leq L_C(D)$ while for at least one $D \in \mathbf{E}$, $L_{C'}(D) < L_C(D)$. If, for a code C, there exists a more efficient code C', we call C *redundant*.

According to the general MDL Principle, a code C' that is more efficient than the two-part code $C_{2\text{-}p}$ is preferable over $C_{2\text{-}p}$: it will compress some data sequences more, and no data sequences less. We now show that $C_{2\text{-}p}$ is indeed redundant, so such a preferable C' must exist.

Note first that the two-part code instantiation of MDL is based on a description method that is actually *not* a code: it is based on a one-many relation and not on a one-to-one mapping (see Chapter 1, Definition 1.2). For example, using the model class of polynomials we can describe each set of points $D = ((x_1, y_1), \ldots, (x_n, y_n))$ in an infinite number of ways: for each k, we can describe D by first describing a polynomial H_k of degree k and then describing the data with the help of that polynomial. By picking always the k for which the overall description length is minimized, we turned our description method into a code, but the fact remains that the coding scheme as a whole allows each data set to be encoded in an infinite number of ways. This means that an infinite number of codewords are reserved for each data sequence where only one is needed; it follows that there must exist some code C' which is more efficient (in the sense defined above) than $C_{2\text{-}p}$.

This leads us to search for an alternative code which does give the *shortest possible code lengths* to those data sequences for which there exists a good model in $\mathcal{M}$. According to MDL, this would be the truly optimal code to use in inductive inference; the two-part code would serve only as approximation. To find this code, we have to make mathematically precise the notion of 'the shortest possible code lengths for all data sequences for which a good model in $\mathcal{M}$ exists'. This is tricky but possible; we will do it in Section 2.3. Once this has been done, the optimally compressing code can be defined explicitly. *The code length of D when encoded using this MDL-optimal code is called the* **stochastic complexity** *of D with respect to the model class $\mathcal{M}$.*

This gives only a first idea; once we will have formally established the concept, we will see that it can be interpreted in several different ways. This is done in Section 2.4. We will then also see how stochastic complexity can be applied to two very important problems that arise in statistical practice: *model class selection* and *prediction*.

2.2 The Concept of 'Model'

In this section we will argue that all models (hypotheses) can be interpreted as codes and therefore also as probability distributions. We start with a discussion on the notion of 'model'.

2.2.1 Models are Fit-measurers

In Chapter 1 we considered probabilistic and non-probabilistic models. The latter were implicitly taken to be functions over the data (in our example we considered the class of polynomials). Now a function H is really an incomplete specification of a model, since, given data D, H in itself does not tell us how well it fits D. It is for this reason that a function becomes a model only if it is accompanied by an *error function* which specifies for each possible realization of the data how well H fits that data, or, equivalently, how well that data is *explained* by H. In the polynomial example, this was achieved by associating with each polynomial the squared error function. Henceforth we shall call a model that gives a 'fit' to *all* possible realizations of the data *complete*:

Definition 2.1 *A* complete model *M for sample space $\mathbf{E}$ is a function $M : \mathbf{E}^* \to \mathbf{R}$. For each realization of the data D, $M(D)$ measures how well M fits D.*

We use the convention that the higher $M(D)$, *the higher the error* M *makes on* D *and hence the* worse M *fits* D.

We note that a probabilistic model is a special case of a complete model. Specifically, for a given probabilistic model P we can define $M_P(D) = g(P(D))$ where g is some monotone decreasing function. In this way, the higher the probability, the better M_P fits the data.

Definition 2.2 *Let* $\mathcal{M}$ *be a set and* $\mathbf{E}$ *be a sample space. An* error function *for* $\mathcal{M}$ *is a total function* $\text{ER} : \mathbf{E} \times \mathcal{M} \rightarrow \mathbf{U}$ *where* $\mathbf{U} \subseteq \mathbf{R}$. *For* $x \in \mathbf{E}$ *and* $H \in \mathcal{M}$, *we write* $\text{ER}(x|H)$ *rather than* $\text{ER}(x, H)$. *We restrict ourselves to* additive *error functions*[1]*:* ER *is extended to outcomes* $x^n \in \mathbf{E}^n$ *by* $\text{ER}(x^n|H) = \sum_{i=1}^{n} \text{ER}(x_i|H)$.

Obviously, we can define a mapping such that each $H \in \mathcal{M}$ is mapped to a complete model M_H with for all $D \in \mathbf{E}^*$:

$$M_H(D) = \text{ER}(D|H) \tag{2.1}$$

The set containing M_H for all $H \in \mathcal{M}$ forms a class of complete models. In the supervised setting (Chapter 1, Section 1.4), $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$ and the data D can be written as $D = (x^n, y^n)$ where the x^n may be regarded as given. The models $\mathcal{M}$ will then be functions from $\mathbf{E}_x$ to $\mathbf{E}_y$. In this case, we should consider *conditional* complete models $M_H(y^n|x^n)$ where, for each (x^n, y^n), $M_H(y^n|x^n)$ measures the error that H makes on y^n given x^n. Here is an example:

Example 2.3 Let $\mathbf{E}_y = \mathbf{R}$. Let $\mathcal{M}$ be a class of functions from $\mathbf{E}_x$ to $\mathbf{E}_y$ (for example, the polynomials). The typical error function to use in this case is the squared error with $\text{ER}_{sq}(x^n, y^n|H) = \sum_{i=1}^{n}(y_i - H(x_i))^2$. We usually write $\text{ER}_{sq}(y|H, x)$ rather than $\text{ER}_{sq}(x, y|H)$ in order to express the fact that x may be regarded as given. For every H, we can arrive at a complete model M_H by setting $M_H(y^n|x^n) := \text{ER}_{sq}(y^n|H, x^n)$.

Here is another example:

Example 2.4 [concept learning] An important form of 'supervised learning' is *concept learning*. Here $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$, $\mathbf{E}_y = \{0, 1\}$ and the model class $\mathcal{M}$ consists of functions $H : \mathbf{E}_x \rightarrow \mathbf{E}_y$ called 'concepts'. If $H(x) = 1$, then x 'belongs to the concept'; otherwise x falls outside the concept. Most of the model classes usually considered in *Computational Learning Theory* [2] are classes of concepts. A natural way to measure how well a concept H fits given data $D = (x^n, y^n)$ is simply to count the number of mistakes H makes on D. For this, we define the *0/1-error function* ER_{01} (using the same notational convention as for ER_{sq}) as

$$\text{ER}_{01}(y|H, x) = \begin{cases} 0 & \text{if } H(x) = y \\ 1 & \text{otherwise.} \end{cases}$$

[1]We can, of course, define non-independent error functions such that the error H makes on x_i depends on one or more of the outcomes $x_{i-1}, x_{i-2}, \ldots$ but we will not consider such cases in this thesis.

This is extended to sequences of outcomes by

$$\text{ER}_{01}(y^n|H, x^n) := \sum_{i=1}^{n} \text{ER}_{01}(y_i|H, x_i).$$

Using Equation 2.1 we can turn each concept H into an equivalent complete model M_H.

Henceforth, we assume each model class to consist either of complete models or to be associated with an error function ER so that it can be turned into an equivalent class of complete models in the way indicated above.

Discussion Is this reasonable? One may argue that, in realistic situations, we want to learn a model from the data and we then want to use this model to give good predictions under all kinds of loss functions. For example, we may want to find a polynomial that allows for good prediction of future data not only in the squared error sense, but also when other loss functions are used. Nevertheless, as long as we are in the learning stage, we must have *some* means to determine how well a model fits the given data, otherwise there may appear data D such that, for two different models H_1 and H_2, we have no way of deciding which of the two better fits the data. Therefore we must use some error function when *learning* from a sample D - irrespective of whether or not that same function will be used later to measure prediction loss over future data. Indeed, in Chapter 4 we will distinguish between 'error functions', used in the learning phase, and 'loss functions', used for prediction of future data. In the Epilogue to Part I of this thesis (page 269), we will see that under some circumstances, learning using error function ER will give good predictions under some loss functions LOSS $\neq$ ER. However, there are also circumstances under which using a loss function different from the error function may lead to bad results.

2.2.2 Complete Models correspond to Probability Distributions

et $\mathcal{M}$ be a class of complete models. It so happens that, under quite general conditions on $\mathcal{M}$ (we will identify these conditions in Chapter 5, Section 5.2), there exists a corresponding class of probabilistic models $\mathcal{M}_{pr}$ that is 'equivalent' to $\mathcal{M}$ in the following sense: there exists a bijection g that maps each $M \in \mathcal{M}$ to a $P(\cdot|M) \in \mathcal{M}_{pr}$ such that, for all n, all $x^n \in \mathbf{E}^n$,

$$-\log P(x^n|M) = M(x^n) + K_n \tag{2.2}$$

where K_n depends only on n, and not on either M or x^n (see [127, 128]).

In the case that $\mathcal{M}$ itself is the complete model class corresponding to a class of functions $\mathcal{M}_{\text{H}}$ and an error function ER, this gives by (2.1) that for all $x^n \in \mathbf{E}^n$:

$$-\log P(x^n|M_H) = M_H(x^n) + K_n = \text{ER}(x^n|H) + K_n \tag{2.3}$$

Hence $-\log P(x^n|M_H)$ is equal to the error H makes on x^n up to an additive constant. In the case of supervised learning, where $D = (x^n, y^n)$ and $\mathcal{M}_H$ consists of functions from $\mathbf{E}_x$ to $\mathbf{E}_y$, we should regard the x^n as given. Therefore, we should map M_H to a *conditional* distribution yielding:

$$-\log P(y^n|M_H, x^n) = M_H(y^n|x^n) + K_n = \text{ER}(y^n|H, x^n) + K_n \tag{2.4}$$

Example 2.5 In the case of the squared error, the probability distribution $P(y^n|M_H, x^n)$ in (2.4) is the conditional normal distribution that was defined in Chapter 1, Equation 1.15 on page 22.

Example 2.6 For the 0/1-error, let us define

$$P(y^n|H, x^n) = \prod_{i=1}^{n} P(y_i|H, x_i) \text{ where}$$
$$P(y|H, x) = \frac{1}{Z} 2^{-\text{ER}_{01}(y|H,x)} \text{ with } Z = \sum_{y \in \{0,1\}} 2^{-\text{ER}_{01}(y|H,x)} = 3/2.$$

which is immediately seen to satisfy (2.4).

2.2.3 Probability Distributions are Codes are Models

Rissanen [127, 128] claims that *every* reasonable model class can be transformed into an associated class of probabilistic models in the manner indicated above. This requires that, for every model class $\mathcal{M}$ and every n, a suitable constant K_n exists such that (2.3) holds. Since the constant K_n plays no role in the inferences made[2], this leads directly to the further claim that, in developing a theory of inductive inference, we need consider only probabilistic models[3]. Bayesian statisticians [17, 38] arrive at that same conclusion on quite different grounds. For the time being, we will assume that Rissanen and the Bayesians are right, since this greatly facilitates the treatment of stochastic complexity. However, we will have much more to say on this issue in the Epilogue to Part I of this thesis, page 119.

Let us assume then that every reasonable model class has a corresponding class of probabilistic models. By the arguments of Chapter 1, Section 1.2.3, for fixed sample size n such a class can be seen as a class of probability distributions P over $\mathbf{E}^n$ all of which have a corresponding code C_P such that $L_{C_P}(x^n) = -\log P(x^n)$ for all $x^n \in \mathbf{E}^n$. Putting everything together, we can interpret all models, probabilistic or not, as *codes*. The obvious advantage of doing this is that it allows for a unified treatment of probabilistic and non-probabilistic model classes.

2.2.4 All models are 'probabilistic'

From now on, following Rissanen [128], we *identify* models, codes and probability distributions. Therefore, all model classes (unless explicitly stated otherwise) will be assumed to consist of *probabilistic models*. For example, we will regard the class

[2] This is actually true only if the constant has the same value among all the models being compared. Let us assume for the moment that it is true (it is immediately seen to be true for the squared error and the 0/1-error). We discuss this further in Chapter 5, Section 5.2.

[3] See [128], page 18/19.

of polynomials as being probabilistic too, identifying each polynomial H with the associated $P(\cdot|H)$ as given by Equation 1.15 on page 22.

In Chapter 1, pages 11 and 16, we defined the two-part code length $L_{2\text{-}p}(D|\mathcal{M})$ as the sum of the code length of a hypothesis θ and the code length of the data D when encoded 'with the help of θ'. The latter term was denoted $L_{C_2}(D|\theta)$. From now on, we regard θ itself simply as a code and drop the subscript C_2. Coding the data 'with the help of θ' now becomes equivalent to coding the data 'using the code θ'.

We stress once more that it is a subtle issue whether one should associate non-probabilistic models with probabilistic ones on the basis of Equation 2.3 (rather than in some other way) and whether one should associate probabilistic models θ with codes with lengths $L(x^n|\theta) = -\log P(x^n|\theta)$ (rather than in some other way). In Chapter 5, sections 5.4 and the Epilogue to Part I of this thesis this is discussed further. For now, we will simply go along with it and assume that it is unproblematic.

Modeling vs. Generating Distributions It is very important to keep in mind that our probabilistic models are not necessarily related to the traditional notion of probability distributions 'according to which the data are drawn'. Indeed, as we show in Chapter 4, only in some special cases will probabilities have anything to do with frequencies. From this it follows that we may not just take expectations over probabilistic models the way we are used to. What we can and what we cannot do with our probabilistic models will be discussed in Chapter 4.

Sometimes we do want to speak of a 'classical' probability distribution. Whenever we do that, we call it a (data) *generating* distribution; we denote such a distribution by P^* and a parameter vector that indexes it by θ^*. In contexts where there could arise confusion, we call probability distributions that are used as hypotheses '*modeling* probability distributions'.

Model achieving Least Error for given data D =
Probability Distribution achieving Maximum Likelihood of D =
Code achieving Minimum Codelength for D

If the data D is then such that the maximum likelihood estimator $\hat{\theta}(D) \in \mathcal{M}$ exists, we see by Equation 1.8 (Chapter 1) that $\hat{\theta}(D) \in \mathcal{M}$ coincides with the code in $\mathcal{M}$ which yields the shortest code length of D. In case that $\mathcal{M}$ is a probabilistic version of a non-probabilistic model class $\mathcal{M}'$, we see by Equation 2.3 that $\hat{\theta}(D)$ corresponds to the model in $\mathcal{M}'$ under which D has the least error. Hence we can identify the best-fitting model with the maximum likelihood model and the minimum-code length code.

2.3 Stochastic Complexity

From the previous section we know that, without any loss of generality, we may confine ourselves to probabilistic model classes. We also know that, for given data D, the model H in model class $\mathcal{M}$ that maximizes the likelihood of D is also the model that, interpreted as a code, assigns the shortest description length to D. These are the two necessary ingredients for a formal treatment of stochastic complexity, which now follows.

Recall that at the beginning of this chapter we defined the stochastic complexity of data D relative to model class $\mathcal{M}$ as the description length of D obtained when it is encoded using some special code C_{sc}. We now derive a mathematical expression for C_{sc}. For simplicity, we only consider the case where $\mathcal{M}$ can be parameterized by a fixed number of parameters k. This is formalized in the following definition:

Definition 2.7 *Let $\mathcal{M}$ be a class of probabilistic models over sample space* $\mathbf{E}$ *and let $\Gamma \subset \mathbf{R}^k$. We say that $\mathcal{M}$ is finitely parameterized by Γ if*

1. *There exists a bijection $g : \Gamma \rightarrow \mathcal{M}$.*
2. *Let $D \in \mathbf{E}^*$ be arbitrary but fixed. Then $P(D|\theta)$ as a function of θ is the restriction to domain Γ of a continuous function $\mathbf{R}^k \rightarrow \mathbf{R}$.*
3. *If* $\mathbf{E}$ *is continuous, then for all n, the density function $f(x^n|\theta)$ as a function of x^n is continuous at each x^n in the interior of $\mathbf{E}^n$.*

Throughout this section we will assume a setting with a sample space $\mathbf{E}$ and a model class $\mathcal{M}$ of models over $\mathbf{E}$ such that

C1 All outcomes fall within $\mathbf{E}_* \subseteq \mathbf{E}$ where, if $\mathbf{E}$ is discrete, $\mathbf{E}_*$ contains a finite number of elements and if $\mathbf{E}$ is continuous, $\mathbf{E}_*$ is compact[4] (closed and bounded).

C2 For all $x^n \in \mathbf{E}^n$, the ML estimator $\hat{\theta}(x^n)$ exists.

C3 $\mathcal{M}$ is finitely parameterized by some $\Gamma \subset \mathbf{R}^k$. Hence $\mathcal{M}$ can be written as $\mathcal{M} = \{P(\cdot|\theta) \mid \theta \in \Gamma\}$.

Let $D \in \mathbf{E}^n$ be our observational data. The MDL Principle tells us to look for an encoding of D that is as short as possible. As noted above, the model in $\mathcal{M}$ that permits the shortest encoding is the maximum likelihood model $\hat{\theta}(D)$. It *seems* we should code our data D using the ML model, in which case the MDL Principle would reduce to the maximum likelihood method of classical statistics!

However - and this is the crucial observation which makes MDL very different from ML - we are looking for a single, *fixed* optimal code C_{sc}, which compresses *all* data samples that are well modeled by $\mathcal{M}$. But the code corresponding to $\hat{\theta}(D)$, i.e. the code that encodes any D' using $L(D'|\hat{\theta}(D)) = -\log P(D'|\hat{\theta}(D))$ bits, only gives optimal compression for *some* data sequences (among which D). For most other data sequences $D' \neq D$, $\hat{\theta}(D)$ will definitely not be optimal: if we had been given such a different data sequence D' (also of length n) instead of D, then the code corresponding to $\hat{\theta}(D')$ rather than $\hat{\theta}(D)$ would give us the optimal compression. In general, coding D' using $\hat{\theta}(D)$ (i.e. using $L(D'|\hat{\theta}(D))$ bits) may be very inefficient.

Ideally, we would like to have a single code C_1 such that $L_{C_1}(D) = L(D|\hat{\theta}(D))$ for all possible D. However, such a code does not exist as soon as our model class contains more than one element – the reason being the familiar fact that, whatever code we

[4]Note that if $\mathbf{E}$ itself is either finite or compact, we may choose $\mathbf{E}_* = \mathbf{E}$.

use, only very few data sequences can receive short code lengths (Chapter 1, Section 1.1.1). It therefore makes sense to define the *regret* $\mathcal{R}_C(\cdot|\mathcal{M})$ of a code C relative to a class $\mathcal{M}$ as the function that, for each D of length n, gives the excess code length relative to the code length based on the ML estimator:

$$\mathcal{R}_C(D|\mathcal{M}) = L_C(D) - L(D|\hat{\theta}(D)) \quad (\text{where } \hat{\theta}(D) \in \Gamma) \tag{2.5}$$

We define the *stochastic complexity code* $C_{sc}(\cdot|\mathcal{M})$ as the code that minimizes the worst-case[5] regret. The worst-case regret is

$$\max_{D \in \mathbf{E}^n} \{\mathcal{R}_C(D|\mathcal{M})\} \tag{2.6}$$

The conditions on $\mathbf{E}$ ensure that this maximum exists. In general, there may exist several codes achieving the worst-case regret but they will share the same code lengths $L_{sc}(\cdot|\mathcal{M})$. The code length $L_{sc}(D|\mathcal{M})$ is called the *stochastic complexity* of data D with respect to model class $\mathcal{M}$:

Definition 2.8 [**stochastic complexity**] *Let $\mathcal{M} = \{P(\cdot|\theta) \mid \theta \in \Gamma\}$ be a probabilistic model class over $\mathbf{E}$ such that conditions* **C1**-**C3** *hold. Let $n > 0$ and let $\mathcal{L}$ be the set of functions $L : \mathbf{E}_*^n \to \mathbf{R}$ which satisfy the Kraft Inequality $\sum_{x^n \in \mathbf{E}_*^n} 2^{-L(x^n)} \leq 1$. Let $D \in \mathbf{E}^n$. The* stochastic complexity of data D relative to model class $\mathcal{M}$ *is the code length $L_{sc}(D|\mathcal{M})$ defined by*

$$L_{sc}(\cdot|\mathcal{M}) := \arg\min_{L' \in \mathcal{L}} \max_{\substack{x^n \in \mathbf{E}_*^n \\ \hat{\theta}(x^n) \in \Gamma}} \{L'(x^n) - L(x^n|\hat{\theta}(x^n))\} \tag{2.7}$$

In the definition above, the set $\mathcal{L}$ is the set of all functions L which can be interpreted as (idealized) code lengths.

The regret $\mathcal{R}_{sc}(D|\mathcal{M}) = L_{sc}(D|\mathcal{M}) - L(D|\hat{\theta}(D))$ must be equal for all D of length n (otherwise it would be possible to construct a code C' with a smaller worst-case regret). We can therefore define $K_{sc}(n|\mathcal{M}) = \mathcal{R}_{sc}(D|\mathcal{M})$ and write

$$L_{sc}(D|\mathcal{M}) = L(D|\hat{\theta}(D)) + K_{sc}(n|\mathcal{M}) \tag{2.8}$$

Whenever the model class $\mathcal{M}$ is clear from the context, we write $L_{sc}(D)$ and $K_{sc}(n)$ rather than $L_{sc}(D|\mathcal{M})$ and $K_{sc}(n|\mathcal{M})$.

The Trade-off The term $L(D|\hat{\theta}(D))$ in (2.8) is called the *goodness-of-fit* term. It reflects for each D how well D is fitted by the model in the class that fits D best. As a function of n, $K_{sc}(\cdot|\mathcal{M})$ is called the *model cost* of $\mathcal{M}$ or equivalently the *complexity* term. It measures the extra code length needed to encode the data due to the richness of $\mathcal{M}$. The stochastic complexity thus embodies a trade-off between fit and complexity. To see this for the polynomial example, let us compare the polynomials in Figure 1 on page x of the introduction: let $\mathcal{M}_i$ be the class of all polynomials of the i-th degree.

[5]The reasons why we are interested in the worst-case over all D will be discussed in Section 2.4.2, page 39.

Let $\hat{\theta}_i(D)$ be the ML model for data D within class $\mathcal{M}_i$. By equations 1.12 and 1.13 (page 21), the difference in the first terms of $L_{sc}(D|\mathcal{M}_1)$ and $L_{sc}(D|\mathcal{M}_{12})$ is *equal* to the difference in the squared errors on D of the best-fitting models in these classes. We have

$$\begin{aligned} &L_{sc}(D|\mathcal{M}_1) - L_{sc}(D|\mathcal{M}_{12}) = \\ &\quad L(D|\hat{\theta}_1(D)) - L(D|\hat{\theta}_{12}(D)) + K_{sc}(n|\mathcal{M}_1) - K_{sc}(n|\mathcal{M}_{12}) = \\ &\qquad \Delta\mathrm{ER}_{sq} + \Delta\text{complexity} \end{aligned}$$

$K_{sc}(n|\mathcal{M}_1)$ will be much *smaller* than $K_{sc}(n|\mathcal{M}_{12})$: $\mathcal{M}_{12}$ contains well-fitting models for *many many more* data sequences than $\mathcal{M}_1$. Since every code can only give a short code length to very few data sets, $K_{sc}(n|\mathcal{M}_{12})$ must be much larger than $K_{sc}(n|\mathcal{M}_1)$. We give an explicit value for $K_{sc}(n|\mathcal{M}_{12})$ in Section 2.6.

Formula (2.8) is reminiscent of the two-part code (1.6) which also contains an error term and a complexity term. The difference lies in the description methods used; while the two terms in (1.6) were based on an encoding of the data in which some hypothesis θ was encoded *explicitly*, the two terms in (2.8) are arrived at by looking directly for the code lengths that are, in a certain sense, as small as possible. The result is that the total length as given by (2.8) will in general be shorter than that given by the two-part code. However, as we discuss in Section 2.6, asymptotically the difference between the two code lengths will be very small.

The Stochastic Complexity Distribution Since $L_{sc}(\cdot|\mathcal{M})$ is a code length function, we can map it to a probability distribution $P_{sc}(\cdot|\mathcal{M})$ over $\mathbf{E}_*^n$ such that for all D of length n, $L_{sc}(D|\mathcal{M}) = -\log P_{sc}(D|\mathcal{M})$. We call $P_{sc}(\cdot|\mathcal{M})$ the *stochastic complexity distribution with respect to* $\mathcal{M}$. Just like $C_{sc}(\cdot|\mathcal{M})$ is the code that gives the shortest possible code length to those data sets for which there exists a well-fitting model in $\mathcal{M}$, $P_{sc}(\cdot|\mathcal{M})$ is the distribution that gives as much probability as possible to those data sets for which there exists a good-fitting model in $\mathcal{M}$. From (2.8) we have (dropping $\mathcal{M}$ from the conditionals whenever $\mathcal{M}$ is understood from the context):

$$L_{sc}(D) = -\log P_{sc}(D) = -\log P(D|\hat{\theta}(D)) + K_{sc}(n) \tag{2.9}$$

Hence we can write $P_{sc}(D) = P(D|\hat{\theta}(D))/F(n)$, where $K_{sc}(n) = \log F(n)$. If $\mathbf{E}_*$ is finite, then the sum $S(n) = \sum_{D \in \mathbf{E}_*^n} P(D|\hat{\theta}(D))$ is finite too. $F(n) \geq S(n)$, otherwise P_{sc} is not a probability distribution. Since, by definition, L_{sc} minimizes the worst-case regret (2.6), P_{sc} must maximize

$$\min_{x^n \in \mathbf{E}_*^n} \{P(x^n)/P(x^n|\hat{\theta}(x^n))\}.$$

From this it follows that $F(n) = S(n)$, and therefore we can write:

$$P_{sc}(D) = \frac{P(D|\hat{\theta}(D))}{\sum_{D \in \mathbf{E}^n} P(D|\hat{\theta}(D))} \tag{2.10}$$

If $\mathbf{E}$ is continuous, the normalizing sum gets replaced by the corresponding integral. Our regularity conditions **C1**-**C3** ensure that the integral is well-defined and finite.

2.4 Interpretation of the SC Distribution as a Single Hypothesis - Applications of SC

Let $P_{sc} = P_{sc}(\cdot|\mathcal{M})$. In light of the seemodel discussion in Section 2.2, we may regard P_{sc} as a *hypothesis*, i.e. a *single* model: just like any other hypothesis, P_{sc} defines for every possible realization of the data how well that data is explained by P_{sc} - so why not look at P_{sc} as if it where a hypothesis itself? It may then be regarded as the hypothesis that 'summarizes' all individual hypotheses in $\mathcal{M}$. Note however that only in special cases will P_{sc} itself be a member of $\mathcal{M}$.

In analogy to the case for single models (see page 34), we use the terminology 'the code length of D when D is encoded with the help of $\mathcal{M}$' to denote the code length of D when D is encoded using the code $C_{sc}(\cdot|\mathcal{M})$.

A Simpler Definition of Stochastic Complexity Once we have made the conceptual step of interpreting P_{sc} as defining a single hypothesis, we can rephrase the verbal definition of stochastic complexity in a much simpler way.

> The stochastic complexity of data D relative to model class $\mathcal{M}$ is the description length of D when D is encoded with the help of class $\mathcal{M}$.

2.4.1 Applications of SC

The stochastic complexity cannot directly be used to find the best *single* hypothesis $H \in \mathcal{M}$; encodings of data using the code $C_{sc}(\cdot|\mathcal{M})$ do not contain substrings that code for a particular $H \in \mathcal{M}$, so there is no way in which $L_{sc}(\cdot|\mathcal{M})$ tells one what the optimal single H is. If the goal is to identify such a single hypothesis, then the 2-part code of Chapter 1 remains the code of choice. But if we are interested solely in *prediction* or *model class selection*, then stochastic complexity can be fruitfully applied (see Yamanishi [167] for more applications). The case of prediction will be treated in detail in Chapter 6. Here we briefly sketch the case of model class selection.

MDL Principle for Model Class Selection Given data D and two model classes $\mathcal{M}_1$ and $\mathcal{M}_2$, we should prefer model class $\mathcal{M}_1$ if and only if the stochastic complexity of D with respect to $\mathcal{M}_1$ is smaller than the stochastic complexity of D with respect to $\mathcal{M}_2$, i.e. if

$$L_{sc}(D|\mathcal{M}_1) < L_{sc}(D|\mathcal{M}_2) \tag{2.11}$$

If $\mathcal{M}_1$ has a smaller stochastic complexity than $\mathcal{M}_2$, then apparently it captures more of the regularity in the data and as such should be preferred. The larger the difference, the higher the confidence that D is better modeled using $\mathcal{M}_1$ than using $\mathcal{M}_2$. We can use this definition to find out whether, for example, the given data is better modeled by the class of second degree polynomials or by the class of third degree polynomials. More interestingly, we can also compare *completely different model classes* to each other. For example, we can decide whether the class of 3rd-degree polynomials or the class of backpropagation neural networks [72] with 5 hidden units is better for the data at hand.

One can extend the definition of stochastic complexity to model classes that do not have a maximum number of parameters [128, 129]. Having done this, one can even use stochastic complexity to decide whether the data is better modeled by the class of *all* polynomials or by the class of *all* backpropagation neural networks. We note that in traditional statistics, this kind of inductive inference problem can be dealt with only in *ad hoc* ways [128].

2.4.2 Discussion

The definition of stochastic complexity which we gave in the previous section was a very 'pessimistic' one: we minimized the worst-case regret $\mathcal{R}(D)$, where the worst-case was taken over *all* data of length n. One reason for taking the worst-case regret is that it leads to Equation 2.8 (page 36): it precisely reflects for all D how many bits are needed to encode D on the basis of the model in $\mathcal{M}$ that fit D best. Hence it captures the intuitive idea of 'coding with the help of $\mathcal{M}$'. One may nevertheless ask whether there is no other code C' which for some, very few, data sequences gives a very long code length while giving a shorter code length than C_{sc} to all others. If one takes a more traditional point of view and assumes that the data is actually *generated* by one of the distributions in the model class, then such a code C' may be preferable over C_{sc}. One may, in fact, define the code C' that gives the shortest worst-case *expected* regret, where the expectation is taken over a distribution $P^* \in \mathcal{M}$ that is assumed to generate the data and the worst-case is taken over all distributions in the class $\mathcal{M}$ (rather than all data D as in our definition). Rissanen [129], using results from Clarke and Barron [26, 27] , shows that asymptotically the code C' which achieves this minimum worst-case expected regret is (under some regularity conditions on $\mathcal{M}$) *identical* to C_{sc}. This provides an important additional justification of using the code C_{sc} as a basis of stochastic complexity.

2.5 Former Definition of Stochastic Complexity

It turns out that if we *average* over all models in a given model class, we obtain something that is very close to the stochastic complexity. Assume, just for the moment, that our class $\mathcal{M}$ has a finite number of elements, all of which are probabilistic models $P(\cdot|\theta)$. In this case we may define a new probability distribution $P_{av}(\cdot|\mathcal{M})$ as a *mixture* of all the models in the class:

$$P_{av}(D|\mathcal{M}) = \sum_{\theta \in \Gamma} P(D|\theta) \times W(\theta) \tag{2.12}$$

Here $W(\theta)$ is a probability distribution over the models in $\mathcal{M}$ that is introduced for normalization. If we let $W(\theta) > 0$ for all $\theta \in \Gamma$, then for large samples D, $P_{av}(D|\mathcal{M})$ will be approximately equal to $P_{sc}(D|\mathcal{M})$. More precisely, whereas $-\ln P_{av}(D|\mathcal{M})$ and $-\ln P_{sc}(D|\mathcal{M})$ both grow linearly in the size n of D, there difference is bounded by some constant independent of n. Intuitively, the reason for this is that the models in the class that give the highest probability to data D automatically contribute the most to the probability $P_{av}(D|\mathcal{M})$. If D is large, then the contribution of $W(\theta)$ to each term in the summation becomes negligible compared to the contribution of $P(D|\theta)$.

For model classes indexed by parameters ranging over a continuous domain the sum gets replaced by an integral and we obtain

$$P_{av}(D|\mathcal{M}) = \int_{\theta \in \Gamma} P(D|\theta) w(\theta) d\theta \tag{2.13}$$

Here $w(\theta)$ must be some *prior density* over Γ, usually just called a 'prior'. Distribution (2.13) plays a central role in Bayesian statistics [17] where it is called the *marginal distribution* or *evidence*. Rissanen gave the verbal definition of stochastic complexity in 1986 but did not connect it to formula (2.10) until 1996. Until that time, $-\log P_{av}(D|\mathcal{M})$ served as the working definition of stochastic complexity, since it was the closest thing to his verbal definition which he could come up with - just how close depends on the prior w as we will show in the next section. Formally, we define $L_{av,w}(D|\mathcal{M}) = -\log \int P(D|\theta)w(\theta)d\theta$ as the *mixture approximation (with prior w)* to stochastic complexity.

2.6 Asymptotic Expansion of SC

In this section we give an explicit formula for $L_{sc}(x^n|\mathcal{M})$, which will allow us to quantify much more precisely the exact trade-off involved in formula (2.8).

As proved by Rissanen [129], for model classes that satisfy certain regularity conditions, we get the asymptotic expansion of $L_{sc}(x^n|\mathcal{M})$ given in formula (2.14) below. The precise regularity conditions can be found in [129]. Essentially, they state that the CLT (Central Limit Theorem) should hold for the maximum likelihood estimators for all elements in the model classes. Generally speaking, this is the case for model classes which consist of i.i.d. probabilistic models but also, for example, for some non-i.i.d. classes like the class of Markov models.

Let $\mathcal{M}_k$ be a class of probabilistic models parameterized by a set $\Gamma \subset \mathbf{R}^k$. We have, for all $n > 0$ and all x^n with $\hat{\theta}(x^n)$ in the interior of Γ:

$$L_{sc}(x^n|\mathcal{M}) = -\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log\int\sqrt{|I(\theta)|}d\theta + o(1) \tag{2.14}$$

where the integral goes over all $\theta \in \mathcal{M}$ and $\lim_{n\to\infty} o(1) = 0$.

The first term measures the goodness-of-fit of the data using the best-fitting model in the class. From (2.8) we see immediately that the model cost $K_{sc}(n|\mathcal{M})$ is equal to the sum of the second, third and fourth term. The second term measures the part of this complexity term that is due to the number of parameters k; note that it grows linearly in k and logarithmically in n. The third term involves the determinant $|I(\theta)|$ of the *(expected) Fisher information matrix* $||I(\theta)||$; the definition of this matrix is given in Chapter 6. It measures the part of the complexity term that is due to local geometrical properties of the model space.

To get an idea of what this notion means, note that if we have a model class consisting of an enormous number of hypotheses, all of which, however, are the same, then the model class is really not that complex at all. For example, we may have a parameter set $\Gamma = \mathbf{R}^2$ but all parameters are names for the same probabilistic model P. Hence in reality, the model class contains only one model with many different names and it is not complex at all. The regularity conditions needed for (2.14) to hold rule out this situation, but they do not rule out the following possibility: there may be regions in a (continuous) parameter space containing only extremely *similar* models in the sense that $P(\cdot|\theta_1) \approx P(\cdot|\theta_2)$ for all θ_1 and θ_2 in the region. In other regions of the space, $P(\cdot|\theta_1)$ and $P(\cdot|\theta_2)$ may be much further apart even though the regions are equally large. Intuitively, such a region is more 'dense': it contains more distinguishable, 'truly different' models. The term $\log \int \sqrt{|I(\theta)|} d\theta$ corrects for this phenomenon.

While the first two terms in (2.14) grow with n the third term does not. Hence, for very large n, it can be safely neglected. The model class selection criterion which arises if we indeed neglect it has, somewhat confusingly, been called the *MDL Model Selection Criterion* [128]. We see that this criterion amounts to the following: For given data D and competing model classes $\mathcal{M}_1, \ldots, M_m$, we should pick the $\mathcal{M}_k$ which minimizes $-\log P(D|\hat{\theta}_k(D)) + \frac{k}{2} \log n$ where $\hat{\theta}_k(D)$ is the maximum likelihood estimator of D within class $\mathcal{M}_k$. This criterion can be used as a first approximation, but since the neglected term does depend on k and can vary quite a lot for different k, one really needs a lot of data to be able to safely neglect it.

In cases where the integral in this term diverges (for example, when the class of normal distributions is used as the model class), Rissanen [129] gives a correction to the formula (2.14).

How close do the approximations get? Both the two-part code and the code based on P_{av} are good approximations to the stochastic complexity. If the model class $\mathcal{M}_k$ satisfies suitable regularity conditions similar to those needed for (2.14) to hold, one can prove that, for all x^n with $\hat{\theta}(x^n)$ in the interior of the parameter space, the two-part code length $L_{2\text{-}p}(x^n|\mathcal{M})$ approaches $L_{sc}(x^n|\mathcal{M})$ to within a constant:

$$L_{2\text{-}p}(x^n|\mathcal{M}) = -\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2} \log n + O(1) \qquad (2.15)$$

and hence becomes a better and better approximation as n increases. In Chapter 7 we describe a modification to two-part codes (introduced in [129]) which removes their inherent redundancy that was noted in Section 2.1. For this improved 2-part code, we get the same asymptotic expansion as (2.14). We call such a code 'asymptotically perfect', since, for all x^n, the difference in description length of data x^n between this code and the stochastic complexity code C_{sc} goes to 0 as n goes to infinity. The mixture $L_{av,w}(x^n|\mathcal{M})$ can - once more under similar regularity conditions - be rewritten exactly as (2.15). A special case is obtained if we take the prior w to be *Jeffreys' prior* [78, 17]. Jeffreys' prior is given[6] by $\pi(\theta) \propto \sqrt{|I(\theta)|}$ where $|I(\theta)|$ is the determinant of the Fisher information matrix that also appeared in Equation 2.14. Balasubramanian [9] showed

[6]The symbol $\propto$ denotes proportionality: $\pi(\theta) \propto \sqrt{|I(\theta)|}$ means that there exists a constant c such that for all $\theta \in \Gamma$: $\pi(\theta) = c\sqrt{|I(\theta)|}$.

that the asymptotic expansion of $L_{av,\pi}(x^n|\mathcal{M})$ is given by (2.14). Hence the evidence approximation of stochastic complexity using Jeffreys' prior is asymptotically perfect.

Using the code based on Jeffreys' prior has some advantages over the code $C_{sc}(\cdot|\mathcal{M})$, as we will discuss in Chapter 6. Therefore, it can be regarded as an alternative definition rather than approximation of stochastic complexity [129].

2.7 A Reinterpretation in Terms of Money

In this section, we show that the MDL Principle can be obtained in a rather different way: minimizing description length of a sequence of outcomes turns out to be equivalent to sequentially placing proportional bets on the outcomes in an optimal manner. This reinterpretation forms an additional justification of the identification of hypotheses with code lengths; it also allows us to view the stochastic complexity in a different way.

Proportional Betting Imagine you take part in a betting game. You are asked to place bets on possible outcomes. The set of possible outcomes is $\mathbf{E} = \{e_1, \ldots, e_k\}$. The odds in the game are k-to-1 on all outcomes e_i. This means that you are offered a ticket for e_i at the price of 1 Euro. If you buy the ticket and the outcome of the game is indeed e_i, then you receive k Euroes. Upon any other outcome $e_j \neq e_i$, you receive nothing and your investment is lost. We consider an 'idealized' game in that you are allowed to buy any number f of tickets, where f may be every rational number. If you buy f tickets for outcome e_i, then we say that you 'place a bet of f on outcome e_i'.

You are allowed to bet on several different outcomes at the same time. This implies that the bets are 'fair' in the following sense: suppose your start out with a capital f. By equally dividing f over the possible outcomes (i.e. placing a bet of f/k on all e_i) your capital after the game will always be equal to your starting capital. Hence it is always possible to place a 'safe bet' so that even if you are completely ignorant about the process generating the outcomes, you are guaranteed not to lose anything.

Notice that this setup is very similar to the game that is actually offered to you at the horse races [30]; the only essential difference is that in the actual horse race, the odds will be different for different horses, set up in such a way that they allow the book maker to make money.

Now suppose you have a probability distribution P over the k possible outcomes. You want to take part in the game described above such that the amount you bet on outcome e_i is proportional to $P(e_i)$. You decide to invest f Euroes, so your bet on e_i is $fP(e_i)$. We refer to this way of dividing your investments as *proportional betting.* Now your remaining capital after playing the game is clearly $kfP(x)$ where $x \in \mathbf{E}$ is the outcome that actually took place.

Next suppose the game involves a sequence of outcomes $x^n = (x_1, \ldots, x_n) \in \mathbf{E}^n$, and you have a probability distribution $P : \mathbf{E}^n \to [0, 1]$. Playing the game as before, you now place bets on every different $x^n \in \mathbf{E}^n$ proportionally to $P(x^n)$. Since there are k^n possible outcomes, the odds are 1-to-k^n and your end capital will then be

$$f \cdot k^n \cdot P(x^n) \tag{2.16}$$

where x^n is the sequence that actually occurred. But you may also play another game consisting of n sub-games. In the first sub-game, you divide all your capital among bets on the first outcome, x_1. The amount you put on outcome $e \in \mathbf{E}$ is $f \cdot P\{X_1 = e\} = \sum_{x_2,\dots,x_n} P(ex_2 \dots x_n)$. After this first sub-game, your remaining capital is $f \cdot kP(x_1)$ where x_1 is the outcome that actually occurred. You now invest all remaining capital in bets on the second outcome x_2 using the *conditional* probability distribution $P(x_2|x_1) = P(x_1x_2)/P(x_1)$. Hence

$$P\{X_2 = e|X_1 = x_1\} = \frac{\sum_{x_3,\dots,x_n \in \mathbf{E}^{n-2}} P(x_1 e x_3 \dots x_n)}{\sum_{x_2,x_3,\dots,x_n \in \mathbf{E}^{n-1}} P(x_1 x_2 x_3 \dots x_n)}$$

After the second outcome, your remaining capital is $f \cdot k^2 P(x_1)P(x_2|x_1)$ where x_1 and x_2 are the two outcomes that actually occurred. Clearly, if you continue betting on x_i using the conditional distribution $P(x_i|x_1, \dots x_{i-1})$ where $x_1, \dots, x_{i-1}$ are the first $i - 1$ outcomes that actually occurred, then after the n-th and last sub-game your capital is:

$$fk^n \Pi_{i=1}^n P(x_i|x^{i-1}) = fk^n \frac{P(x_1)}{1} \frac{P(x_1x_2)}{P(x_1)} \frac{P(x_1x_2x_3)}{P(x_1x_2)} \cdots \frac{P(x^n)}{P(x^{n-1})} = fk^n P(x^n) \tag{2.17}$$

This coincides with (2.16) and we see that both ways of playing the game are equivalent! As shown below, this allows us to interpret the MDL Principle in an alternative way. We can identify 'predicting that x will happen with probability p' with 'placing a bet on x proportional to p'.

Gambling Interpretation of Stochastic Complexity Let $\mathcal{M}$ be a class of probabilistic models over a discrete space $\mathbf{E}$. Consider the ML estimator $\hat{\theta}(x^n)$. Suppose we predict the x_i sequentially by placing a bet on x_1 proportional to $P(x_1|\hat{\theta}(x^n))$, a bet on x_2 proportional to $P(x_2|x_1, \hat{\theta}(x^n))$, etc. By Equation (2.17) our end capital is proportional to $P(x^n|\hat{\theta}(x^n))$. Had we played the same game using any other $\theta' \neq \hat{\theta}(x^n)$, our end capital would have been proportional to $P(x^n|\theta')$ which by definition is lower than $P(x^n|\hat{\theta}(x^n))$. Hence if we had predicted all the x_i on the basis of the ML estimator $\hat{\theta}(x^n)$, our end capital would have been maximized.

Unfortunately, we can only know the optimal predictor $\hat{\theta}(x^n)$ after the fact: we only know what *would have been* the optimal predictions after the data has been made available to us. From a 'minimax' viewpoint, the best way to predict is then to use a predictor P_{opt} which we can construct a priori (before seeing any data) and which is always as close as possible to the predictor that turns out to be optimal a posteriori. We proceed to show that P_{opt} is identical to the stochastic complexity distribution $P_{sc} = P_{sc}(\cdot|\mathcal{M})$ as defined on page 37. Using predictor (probability distribution) $\mathcal{P}$, the worst possible data that may appear for that predictor is the data which maximizes the proportion between the end capital arrived at using the a-posteriori optimal $P(\cdot|\hat{\theta}(x^n))$ and the end capital arrived at using $\mathcal{P}$, given by

$$\max_{x^n} \frac{P(x^n|\hat{\theta}(x^n))}{\mathcal{P}(x^n)} \tag{2.18}$$

P_{opt} is the distribution which minimizes this worst-case discrepancy. Comparing Equation (2.18) to Equation (2.7) and rewriting probabilities as code lengths, we see that P_{opt} and P_{sc} coincide.

Logarithmic Loss Let $\mathbf{E} = \{e_1, \ldots, e_k\}$ and let P be a probability distribution over $\mathbf{E}$. Suppose we predict that e_1 happens with probability $P(e_1)$, e_2 happens with probability $P(e_2)$ etc. Then some $x \in \mathbf{E}$ arrives and we are charged with a *loss* that measures the discrepancy between our predictions $P(e_1), P(e_2), \ldots$ and the actual outcome x. The *logarithmic loss* is defined as $\text{LOSS}_{lg}(x) = -\log P(x)$. Note that the higher the probability assigned to the actual outcome, the lower the logarithmic loss. In the statistical literature, this loss function is often used as a 'generic' loss function for probabilistic prediction [127]. By the arguments above, we see that we can regard the logarithmic loss both as the code length needed to encode x or as the logarithm of the factor by which our capital is multiplied if we bet proportionally on the e_i according to P.

Four Ways of Looking at MDL Summarizing, we see that we can alternatively interpret the goal of MDL as minimizing description length, maximizing probability, maximizing end capital in proportional betting, or, finally, minimizing logarithmic loss.

Predictive MDL We have seen that the stochastic complexity can be approximated by the two-part code length and by a 'mixture' approximation. At this point we should mention that there is a very important third way of approximating it which has its roots in the gambling interpretation. It is called 'predictive MDL' and was introduced independently with somewhat different motivations by Rissanen [127] and Dawid[7] [36, 37]. For details we refer to Rissanen [128].

2.8 MDL and Bayesian Statistics

Some authors use this approach so that they can use Bayesian methods in disguise without being ridiculed by their anti-Bayesian colleagues.
Wray Buntine on MDL

At this point we have discussed all main ingredients of MDL: the two-part codes, the stochastic complexity and the idea of viewing all model classes as 'probabilistic'. It is now instructive to compare MDL to Bayesian Statistics [17, 38, 18, 77]. Bayesian Statistics and MDL are closely related. Some people have even – wrongly – claimed that, philosophical differences notwithstanding, the two are really the same for all practical purposes. This claim probably stems from the fact that the mathematical formulas used in applications of MDL and Bayesian statistics coincide in two important cases:

[7] Strictly speaking, it can be seen as a special case of Dawid's 'prequential assessment' when used with the logarithmic loss.

Evidence and Stochastic Complexity In Bayesian statistics we always use a prior distribution w for all elements in the chosen model class $\mathcal{M}$. We can then simply calculate the *conditional* probability of the data D given model class $\mathcal{M}$ as

$$P(D|\mathcal{M}) = \int_{\theta \in \Gamma} P(D|\theta) w(\theta) d\theta$$

which coincides with our P_{av} as given by Equation (2.13). Bayesian model class selection [163] works as follows: if we are given some data D and we have to decide whether it is better explained by model class $\mathcal{M}_1$ or by model class $\mathcal{M}_2$, then we choose the class $\mathcal{M}_i$ for which the *evidence* $P(D|\mathcal{M}_i)$ is highest.

If we are willing to use P_{av} with prior w as the 'mixture approximation' to stochastic complexity (see Section 2.5) then, because of the monotonicity of '$-$' and log, this is equivalent to choosing the class with the lowest stochastic complexity.

MAP and 2-part MDL Bayesians often approximate the evidence $P(D|\mathcal{M})$ by $P(D|\breve{\theta})$, $\breve{\theta}$ being the *maximum a posteriori (MAP) model* for D. This is the model that has 'maximal probability in light of the data':

$$\breve{\theta} = \arg\max_{\theta \in \mathcal{M}} P(\theta|D) = \arg\max_{\theta \in \mathcal{M}} \frac{P(D|\theta)w(\theta)}{P(D)} = \arg\max_{\theta} P(D|\theta)w(\theta) \tag{2.19}$$

where the second equality follows from Bayes' rule [17] and the third follows from the fact that $P(D)$ does not change if we vary θ.

We know that for probabilistic model classes, the MDL estimator θ_{mdl} is given by minimizing $-\log P(D|\theta) + L_{C_1}(\theta)$. By the arguments of Section 1.2.3 in Chapter 1, there exists a probability distribution w such that for all D, $-\log w(D) = L_{C_1}(\theta)$. For this w we have:

$$\theta_{\mathrm{mdl}} = \arg\min_{\theta \in \mathcal{M}} \{-\log P(D|\theta) + [-\log w(\theta)]\} \tag{2.20}$$

Comparing (2.19) and (2.20) we see that they are equivalent!

What, then, is the essential difference between the two approaches? Though it remains somewhat hidden in the two examples above, the aim of Bayesian statistics is usually to make inferences and decisions based on data D that *maximize expected utility.* Here the 'utility' of an inference or decision is some function depending on the problem domain. According to MDL, rather than maximizing expected utility we should base inferences on minimizing code length (how to deal with 'decisions' will be explained in Chapter 4). Summarizing:

> Roughly speaking, the aim of MDL is to compress data as much as possible; the aim of Bayesian statistics is to maximize expected utility.

MDL's focus on compression reveals itself in several ways. First, the definition of stochastic complexity $L_{sc}(\cdot|\mathcal{M})$ does not depend on any particular prior distribution. This is how it should be, since it is defined with respect to D and $\mathcal{M}$ and should thus depend only on D and $\mathcal{M}$ and nothing else. If we approximate $L_{sc}(\cdot|\mathcal{M})$ using $L_{av,w}(\cdot|\mathcal{M})$, we should pick the prior with which we obtain code lengths that are as short as possible in the worst-case. Under suitable conditions, this will be Jeffreys' prior as introduced in Section 2.6.

In the same vein, Rissanen [129] shows that the two-part code length given by (1.5) is suboptimal only: there exists a somewhat more efficient version of the two-part code length that *cannot* be reinterpreted in MAP terms (see Chapter 7). There are different sub-schools of Bayesian Statistics and the proponents of each propose different priors for different reasons [17]; usually however, the purpose of such priors does not seem to be to compress as much as possible.

Since the influence of the prior distribution is typically almost negligible except for very small sample sizes, MDL and Bayesian methods will often lead to similar results in practice. However, 'non-typical' cases where the prior has a large influence even for relatively large sample sizes do occur in practice [129]. In such cases MDL and Bayesian methods may lead to different results.

No Expectations Most Bayesians do not seem to hesitate to take *expectations* over priors (see Chapter 7, Section 7.2 for an example). According to MDL, a prior should be used only as a tool to arrive at short descriptions. Therefore it is *not* what we would call a 'generating' probability: the hypothesis θ can certainly not be viewed as 'being drawn according to prior w'. MDL holds the view that taking expectations over non-generating probabilities should be done with care and is meaningful only in certain special cases (see Chapter 4).

MML and Prequential Analysis indexMinimum Message Length Principle o complicate the situation further, there are two branches of Bayesian statistics (or at least, branches of statistics that have their roots in the Bayesian view), which are particularly closely related to MDL. These are (1) Dawid's 'Prequential Analysis' [36, 37] and (2) Wallace's Minimum *Message* Length (MML) Principle [161, 162]. The MML Principle combines Bayesian ideas with data compression; it is discussed at length in Chapter 7. Prequential analysis and MDL share the view that probability distributions are to be seen as *models* of data and not as something 'according to which data are actually distributed'. The relationship between Prequential analysis and MDL is investigated by Dawid in [37].

The differences between Bayesian methods and MDL become more pronounced in the idealized setting where we allow the coding to be done by a universal programming language. This situation has been analyzed in detail by Vitányi and Li [160].

2.9 Conclusion and Outlook

In chapters 1 and 2 we have provided a general overview of the MDL Principle. Summarizing, we can state that the following views are fundamental to MDL:

1. Inductive inference should be done on the basis of a code, or, more generally, a description method.

2. The shorter such a code length assigns to the given data D, the better we have captured the regularities in D.

3. Models should always be viewed as, or be associated with, codes.

4. Classes of models $\mathcal{M}$ can be 'summarized' by a single code: the *stochastic complexity code* based on $\mathcal{M}$.

We discussed several ways of approximating the stochastic complexity; we provided a reinterpretation of 'description methods' in terms of 'betting strategies' and we compared MDL to Bayesian statistics.

However, there are still two unresolved issues we mentioned in passing but did not go into further: in Chapter 1, we argued that overly simple models, or, more generally, models that allow for compression but are nevertheless not in any sense 'true', can nevertheless be useful in predicting or making decisions with respect to future data. This claim still has to be verified. The other issue concerns Section 2.2 in the present chapter, where we connected non-probabilistic model classes to probabilistic ones in a somewhat arbitrary manner (Equation 2.3, page 32). Indeed, as we will show in Chapter 5, Example 5.1, there is something inherently arbitrary about the procedure used in Section 2.2. In Chapter 5 we propose a new idea to deal with both issues raised here. But in order to do so, we need to introduce the concept of *Maximum Entropy* and relate it to MDL. This will be the subject of the next chapter.

Chapter 3

Introduction to Maximum Entropy

This chapter discusses the Maximum Entropy Principle and shows how it is related to MDL. In sections 3-3.6 we give a brief and basic introduction to maximum entropy. We start by informally introducing maximum entropy, giving several examples of its use and relevance. This is followed in sections 3.2-3.4 by formal definitions of entropy, relative entropy, maximum entropy distributions and classes of these distributions. Section 3.5 discusses the 'concentration phenomenon' which lies at the heart of the maximum entropy method. Ever since their introduction, the appropriateness of maximum entropy principles has been fiercely debated. Section 3.6 reviews some of the issues in this debate. In Section 3.7 we show that Maximum Entropy can be seen as a special case of the MDL Principle. The chapter ends with a brief conclusion. sectionMotivation It is a dream of Artificial Intelligence to build intelligent robots. These should be capable of making rational decisions on the basis of incomplete and partially unreliable information coming from their sensors and their database. If a robot's knowledge were represented by a probability distribution (where the probability of e indicates a robot's degree of belief in proposition e), then it could use *decision theory* [17] to arrive at rational decisions. This is only possible if the probability distribution is complete, assigning a probability to each conceivable event e. Where do we get these probabilities from? Some probabilities may be calculated from past experience; but for most of them there is not enough past experience to determine them with any degree of correspondence to the 'real' probabilities. The Maximum Entropy Principle provides a means to arrive at probabilities in the presence of partial knowledge: given certain constraints on a probability distribution, it constructs a *single* probability distribution over all possible outcomes. Once such a single distribution has been constructed, all the tools of decision theory and probability theory become available to make decisions and predictions.

Such a means of arriving at a single distribution from partial knowledge has applications in many other fields. Within the field of AI, we should mention *expert systems* [114] and automated natural language processing [16]. Some successful applications of Maximum Entropy outside of AI are in protein modeling [89], psychological modeling [113], stock market analysis [32] and analysis of genetic algorithms [140]; for a much longer list, see [144]. We will now first discuss maximum entropy in an informal manner.

3.1 Informal Introduction to Maximum Entropy

Let $\mathbf{E}$ be a sample space and let $\phi_1, \ldots, \phi_n$ be functions on $\mathbf{E}$ with ranges $\mathbf{U}_1, \ldots, \mathbf{U}_n$. For simplicity we assume that all $\mathbf{U}_i$'s are intervals in $\mathbf{R}^1$. We are given a finite (possibly empty) set $\mathcal{C}$ of constraints regarding the *expected values* of the functions ϕ_i:

$$\mathcal{C} = \{E[\phi_1(X)] = t_1, \ldots, E[\phi_m(X)] = t_m\} \tag{3.1}$$

where the t_i are values in the interior of $\mathbf{U}_i$.

We ask the following question: if the only knowledge we have about a probability distribution P are the constraints given by (3.1), what is then our best guess for P?

It is of course not well-defined what a 'best guess' really is. A possibility is to look for the P that allows for the best possible prediction of future data. Another option is to search for the P that is in some sense, 'the most likely', given the constraints. A third option is to assume that, at several stages, new information about P will be made available. The aim would then be to pick a P according to a scheme that converges as quickly as possible to the 'true' P according to which the data is actually drawn.

According to some authors, all these aims coincide, and are realized by adopting the *P that maximizes the entropy subject to the given constraints.* The entropy of a distribution P is the expectation of $-\log P(X)$. It is a measure of the amount of randomness or 'disorder' inherent in P. Since $\log P(x)$ determines the goodness-of-fit of data x under model P, the entropy of P may be interpreted as minus the expected goodness-of-fit under P. A distribution with high entropy has a low expected goodness-of-fit. Adopting the probability distribution with the maximum entropy is called the Maximum Entropy Principle. It has been introduced by Jaynes in 1957 [73] but has its roots in the work of Boltzmann and Gibbs on statistical mechanics [77]. Our first example shows that Maximum Entropy is a generalization of Laplace's Principle of Insufficient Reason [91] which we introduced on page xi.

Example 3.1 Let $\mathbf{E} = \{1, \ldots, k\}$. As will be verified shortly, the distribution over $\mathbf{E}$ without any constraints (except $\sum_{i \in \mathbf{E}} P(i) = 1$) with maximum entropy is given by $P(i) = 1/k$ for all $i \in \mathbf{E}$; the entropy of this distribution is

$$E[-\log P(X)] = \sum_{i=1}^{k} P(i)[-\log P(i)] = -k \cdot \frac{1}{k} \log \frac{1}{k} = \log k$$

The expected log-likelihood $E[\log P(X)]$ is $-\log k$. Intuitively, data drawn according to a more skewed distribution over $\mathbf{E}$ will, with high probability, be more regular than data drawn according to P. Indeed the entropy of P decreases (and the expected log-likelihood increases) as $P(i)$ drifts away from $1/k$, and in the limit for $P(i) \to 1, P(j) \to 0$ $(j \neq i)$ the entropy becomes 0.

Before moving on to formal definitions, we give a few more examples of 'MaxEnt', as it is usually abbreviated:

Example 3.2 Let $\mathbf{E} = \{a, b, c\}$. We are given the constraint $P(a) = 1/2$. To see that it is of the form demanded by Equation 3.1, note that $P(a) = p$ is equivalent to $E[\mathcal{I}(X = a)] = p$, where $\mathcal{I}$ is the *indicator function* defined as follows:

$$\mathcal{I}(x = e) = \begin{cases} 1 & \text{if } x = e \\ 0 & \text{if } x \neq e \end{cases} \tag{3.2}$$

For fixed e, $\mathcal{I}(x = e)$ is a function of x only. In such a case we write $\mathcal{I}_e(x)$ instead of $\mathcal{I}(x = e)$ whenever this does not give rise to any confusion.

In the case of constraint $P(a) = 1/2$, the maximum entropy distribution is given by $P(a) = 1/2, P(b) = P(c) = 1/4$.

Example 3.3 [**independence**] Let the bias of a coin be p and the bias of a second coin be q. Let 1 stand for 'heads' and 0 for 'tails'. The maximum entropy distribution over the joint space $\mathbf{E} = \{0, 1\}^2$ is given by $P(11) = pq, P(10) = p(1 - q), P(01) = (1 - p)q, P(00) = (1 - p)(1 - q)$. We see that this is the distribution corresponding to two independent coins.

Example 3.4 [**uniform and normal distribution**] Let $\mathbf{E} = [a, b]$. Otherwise no constraints are given. The maximum entropy distribution is the uniform distribution over $\mathbf{E}$. As another example, let $\mathbf{E} = (-\infty, \infty)$. Let the constraints $E[X] = \mu$ and $\text{VAR}[X] = \sigma^2$ be given. The maximum entropy distribution under these constraints is the normal distribution with mean μ and variance σ^2.

Summarizing the previous examples, *complete ignorance* about the relative likelihood of k outcomes is translated into a *uniform* distribution on these outcomes. Complete ignorance about dependencies between random variables X and Y is translated into a distribution that renders X and Y independent.

3.2 Entropy and Relative Entropy

Preliminaries and Notation Let P be a probability distribution over a sample space $\mathbf{E}$. In the discrete case, the expectation $E_P[\phi(X)]$ of a function of the data ϕ is given by $\sum_{x \in \mathbf{E}} P(x)\phi(x)$. If $\mathbf{E}$ is continuous, then $E_P[\phi(X)] = \int_{x \in \mathbf{E}} f(x)\phi(x)dx$ where $f(x)$ is the density function of P. Whenever we use an indexed probability distribution like, say, P_i, we use $E_i[\phi(X)]$ as short for $E_{P_i}[\phi(X)]$. We define the *support* of a distribution P over sample space $\mathbf{E}$ as the set $\{x \in \mathbf{E} \mid P(x) > 0\}$. We say that P has *full* support if the support coincides with $\mathbf{E}$.

It will be convenient to change our unit of measurement and work in 'nats' in stead of 'bits' throughout the remainder of this chapter. Hence in this chapter, we use $L_P(x)$ as an abbreviation of $-\ln P(x)$, the code length (measured in nats) of x obtained by using the code based on P. In case $\mathbf{E}$ is continuous, we adopt the same conventions as in Chapter 1, Section 1.2. Specifically, $-\ln P(X)$ should then be read as $-\ln f(x)$ where f is the density function of P.

Whenever we write $P(x_1, \ldots, x_n)$ for $x_i \in \mathbf{E}$, we are referring to the product distribution: $P(x_1, \ldots, x_n) = \Pi_{i=1}^{n} P(x_i)$.

Definition 3.5 *The* entropy *of a distribution P is*

$$\mathcal{H}(P) = E_P[L_P(x)] \tag{3.3}$$

The relative entropy *or* Kullback-Leibler (KL) Divergence *between distributions P and Q is given by*

$$D(P||Q) = E_P[L_Q(x) - L_P(x)] = \sum_{x \in \mathbf{E}} P(x) \ln \frac{P(x)}{Q(x)}$$

The entropy has a direct interpretation as the expected number of nats needed to encode an outcome that is distributed according to P, using the code L_P based on P.

If we code the data using the code based on a distribution Q rather than on P, then the expected number of nats needed to encode an outcome drawn according to P will be $E_P[-\ln Q(x)]$. Hence the KL divergence has a direct interpretation as the expected difference in the number of nats needed to encode such an outcome using the code based on P versus the code based on Q. The following result is of fundamental importance (see [30] for a proof):

Theorem 3.6 (Information Inequality) *For any two probability distributions P and Q defined over the same space* $\mathbf{E}$,

$$D(P||Q) \geq 0 \tag{3.4}$$

with equality holding if and only if $P = Q$.

We see that if data are actually distributed according to P, then the optimal (in the sense of minimizing expected code length) code for the data is the code based on P: every other code will yield a larger expected code length. The difference in the expected number of nats needed can be interpreted as a kind of 'distance' between the distributions P and Q. Note however that in general, $D(P||Q) \neq D(Q||P)$ so the KL divergence is not really a distance in the usual mathematical sense of the word.

3.3 Maximum Entropy Distributions

Preliminaries We are given the sample space $\mathbf{E}$ and a set of constraints $\mathcal{C}$ pertaining to an otherwise unknown probability distribution P. We assume $\mathcal{C}$ to be of the form given by Equation 3.1 on page 50. We often abbreviate Equation 3.1 to

$$E[\phi(X)] = t$$

where $\phi(X)$ stands for the vector $(\phi_1(X), \ldots, \phi_m(X))$ and t stands for the vector $(t_1, \ldots, t_m)$. In this way, the set of constraints $\mathcal{C}$ can be equivalently seen as a single constraint on the vector-valued function ϕ. We write $\mathcal{C}(\phi, t)$ to denote the specific constraint $E[\phi(X)] = t$. We say a probability distribution *satisfies* constraint $\mathcal{C}(\phi, t)$

iff $E_P[\phi(X)] = t$. The set of probability distributions P over $\mathbf{E}$ satisfying $\mathcal{C}(\phi, t)$ is denoted by $\mathbf{M}(\phi, t)$: $\mathbf{M}(\phi, t) = \{P \mid E_P[\phi(X)] = t\}$.

In most practical applications, we are not really given expected values but rather measured *average* values of functions over large sets of data. Let $\overline{\phi(x)}^n$ be the average value of ϕ over n observations:

$$\overline{\phi(x)}^n = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i)$$

For a given constraint $\mathcal{C}(\phi, t)$, we define the corresponding *empirical constraint* $\mathcal{C}_e(\phi, t)$ as follows:

$$\mathcal{C}_e(\phi, t) = \{\overline{\phi_1(x)}^n = t_1 \wedge \ldots \wedge \overline{\phi_m(x)}^n = t_n\} = \{\overline{\phi(x)}^n = t\} \tag{3.5}$$

In this case $\mathbf{C}^n(\phi, t)$ denotes the set of data of length n for which the constraint $\mathcal{C}(\phi, t)$ holds:

$$\mathbf{C}^n(\phi, t) = \{x^n \mid \overline{\phi(x)}^n = t\}.$$

3.3.1 Explicit Expression for the MaxEnt Distribution

We are given a constraint $\mathcal{C}(\phi, t)$ for some function $\phi = (\phi_1, \ldots, \phi_m)$ with range $\mathbf{U} = \mathbf{U}_1 \times \ldots \times \mathbf{U}_m$. We look for the distribution P_{me} that maximizes the entropy $\mathcal{H}$ as given in Equation 3.3 under the constraint $\mathcal{C}(\phi, t)$:

$$P_{me} = \arg \max_{P \in \mathbf{M}(\phi, t)} \mathcal{H}(P) \tag{3.6}$$

We verify below that in the discrete case the maximum entropy distribution P_{me} is given by

$$P_{me}(x) = \frac{1}{Z(\beta)} e^{-\beta^T \cdot \phi(x)} \tag{3.7}$$

where $\beta = (\beta_1, \ldots, \beta_m) \in \mathbf{R}^m$ is a set of m parameters; β^T is the transpose of β (and hence $\beta^T \cdot \phi(x)$ denotes the inner product $\sum_{i=1}^m (\beta_i \cdot \phi_i(x))$. The quantity $Z(\beta)$ is used as a normalization factor:

$$Z(\beta) = \sum_{x \in \mathbf{E}} e^{-\beta^T \cdot \phi(x)} \tag{3.8}$$

For all n, $x^n \in \mathbf{E}^n$, $P_{me}(x^n)$ is given by the product distribution of P_{me}: $P_{me}(x^n) = \Pi_{i=1}^n P_{me}(x_i)$.

(3.7) only holds under certain mild regularity conditions. In particular, it must be the case that the sum (integral) in (3.8) converges and that, in the integral case, when differentiating (3.8) we can swap the order of differentiation and integration. It is straightforward to show that the following conditions are sufficient to ensure that there exists a unique value of β such that (3.7) defines a probability distribution for which the constraint $E[\phi(X)] = t$ holds.

Conditions for Existence of Maximum Entropy Distributions
Let $\phi = (\phi_1, \ldots, \phi_m)$ be a function with domain $\mathbf{E}$ and range $\mathbf{U} = \mathbf{U}_1 \times \ldots \times \mathbf{U}_m$. Let $t = (t_1, \ldots, t_m) \in \mathbf{U}$. We require the constraint $E[\phi(X)] = t$ to be such that for $i = 1, \ldots, m$:

C1 $\mathbf{U}_i$ is the smallest interval in $\mathbf{R}$ such that $\forall x \in \mathbf{E} : \phi_i(x) \in \mathbf{U}_i$.

C2 If $\mathbf{E}$ is continuous, then ϕ_i is continuous. More precisely, if $\mathbf{E} \subseteq \mathbf{R}^k$, then ϕ_i is the restriction to domain $\mathbf{E}$ of some continuous function $\psi : \mathbf{R}^k \to \mathbf{R}$.

C3 In the discrete case $\mathbf{E}$ contains a finite number of elements. In the continuous case, $\mathbf{E}$ can be written as $\mathbf{E}_1 \times \ldots \mathbf{E}_l$ for some $l \geq 1$, where for each $\mathbf{E}_j$ with $1 \leq j \leq l$:

1. $\mathbf{E}_j$ is a closed interval in $\mathbf{R}$
   or
2. $\mathbf{E}_j = \mathbf{R}$ and there exist $\alpha > 0$ and $C \in \mathbf{R}$ such that

$$\forall x_1, \ldots, x_l : \phi_i((x_1, \ldots, x_j, \ldots, x_l)) \geq |x_j|^\alpha - C.$$

C4 t_i lies in the interior of $\mathbf{U}_i$.

These conditions are sufficient but by no means necessary to ensure the existence of a maximum entropy distribution. However, they cover all the cases needed in this thesis (condition **C3**.2 may seem a bit counterintuitive; it is added to make sure that, if in the evaluation of (3.8) we have to integrate over $\mathbf{R}$, the integral converges). Henceforth, whenever we speak of a maximum entropy distribution for function $\phi : \mathbf{E} \to \mathbf{U}$ and constraint $E[\phi(X)] = t$ we tacitly assume **C1**-**C4** to hold.

To see that P_{me} indeed maximizes the entropy subject to the constraint $\phi(x) = t$, let Q be any distribution other than P_{me} satisfying the constraint and notice that:

$$\begin{aligned}\mathcal{H}(Q) = E_Q[-\ln Q(X)] < E_Q[-\ln P_{me}(X)] = E_Q[\beta^T \phi(X) + \ln Z(\beta)] \\ = \beta^T E_Q[\phi(X)] + \ln Z(\beta) \overset{(1)}{=} \beta^T t + \ln Z(\beta)\end{aligned} \quad (3.9)$$

where the inequality follows from the information inequality (3.4) and (1) follows from the fact that we defined Q to satisfy the constraint and hence $E_Q[\phi(X)] = t$. On the other hand,

$$\begin{aligned}\mathcal{H}(P_{me}) = E_{me}[-\ln P(X)] = E_{me}[\beta^T \phi(x) + \ln Z(\beta)] \\ = \beta^T E_{me}[\phi(X)] + \ln Z(\beta) \overset{(1)}{=} \beta^T t + \ln Z(\beta)\end{aligned} \quad (3.10)$$

where (1) follows from the fact that, by definition, the maximum entropy distribution satisfies the constraint and hence $E_{me}[\phi(X)] = t$.

Together, (3.9) and (3.10) give that $\mathcal{H}(P_{me}) > \mathcal{H}(Q)$ for all $Q \neq P_{me}$ satisfying the constraint, which is what we had to prove.

The proper value of β can be obtained from the equality:

$$t_i = E_{me}[\phi_i(X)] = -\frac{\partial}{\partial \beta_i} \ln Z(\beta)$$

which is easy to check (see Proposition 3.9).

It can now easily be verified that the distributions mentioned in examples (3.1)-(3.4) are indeed the maximum entropy distributions for the respective constraints; see [30] for details. As an example, the 'empty' constraint, leading to the uniform distribution $P(x) = 1/k$, can always be written as $E[\mathbf{0}(X)] = 0$ where $\mathbf{0}(x)$ is the function that maps each x to 0. The optimal β for this constraint turns out to be 0 and (3.7) becomes the uniform distribution.

3.4 Maximum Entropy Model Classes

As will be seen in the next chapter, it is very useful to consider the class of all Maximum Entropy distributions that satisfy constraints of the same functional form. These classes of distributions play a fundamental role in statistics, where they are known as the 'exponential families'. Many of the model classes that are typically used in statistical practice are of this form; a few examples are the Bernoulli, multinomial, Normal, Beta- and Gamma-distributions.

The maximum entropy model class for a function $\phi : \mathbf{E} \to \mathbf{U}$ contains exactly the maximum entropy distributions for the (vector of) constraints $E[\phi(X)] = t$ for all t in the interior of $\mathbf{U}$. Formally,

Definition 3.7 *Let a function $\phi : \mathbf{E} \to \mathbf{U}$ be given. Let $P_{me,\phi}(\cdot|t)$ be the maximum entropy model for constraint $E[\phi(X)] = t$. The* maximum entropy model class $\mathcal{M}_{me}$ *for function ϕ is given by*

$$\mathcal{M}_{me} = \{P_{me,\phi}(\cdot|t) \mid t \in \mathrm{int}(\mathbf{U})\}$$

where $\mathrm{int}(\mathbf{U})$ *stands for the interior of* $\mathbf{U}$.

Note that if conditions **C1-C3** of page 54 hold for $\mathbf{E}$, $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$ and $\mathbf{U} = \mathbf{U}_1 \times \ldots \mathbf{U}_m$, then the class $\mathcal{M}_{me}$ for function ϕ is guaranteed to exist. Henceforth, whenever we speak of a maximum entropy class for a function ϕ we tacitly assume $\mathbf{E}$, ϕ and $\mathbf{U}$ to be such that **C1-C3** are satisfied.

Example 3.8 Let $\mathbf{E} = \{0, 1\}$. The maximum entropy model for constraint $P(X = 1) = \theta$ is the corresponding Bernoulli model $P(X = 1) = \theta$ (Chapter 1, Definition 1.7). This can be seen by substituting $\beta = \ln(1 - \theta) - \ln \theta$ in Equation 3.7. Hence the Bernoulli model class restricted to $0 < \theta < 1$ is the maximum entropy model class for $\mathbf{E} = \{0, 1\}$ and constraint $P(X = 1) = \theta$. Note that $\beta = 0$ corresponds to the uniform distribution $\theta = 1/2$. As $\beta \to \infty$, $\theta \to 0$ and as $\beta \to -\infty$, $\theta \to 1$: the smaller $|\beta|$, the closer the distribution is to uniform and hence the higher the entropy. This relation can be shown to hold for maximum entropy model classes in general, see Proposition 3.9.

How to parameterize $\mathcal{M}_{me}$ We can parameterize maximum entropy model classes in two useful ways. Note that the maximum entropy class $\mathcal{M}_{me}$ for function ϕ contains exactly one element for every $t \in \text{int}(\mathbf{U})$. Therefore we can identify the set of parameters Γ with $\text{int}(\mathbf{U})$. This way of characterizing models is called *mean-value parameterization* [82]. Alternatively, as we know from Section 3.3.1, each element of $\mathcal{M}_{me}$ may be written as in (3.7) for a particular value of β. Since, again by (3.7), $E_\beta[\phi(X)]$ is a continuous function of β, it follows that the class $\mathcal{M}_{me}$ can also be characterized as follows:

$$\mathcal{M}_{me} = \{P(\cdot|\beta) \mid E_\beta[\phi(X)] \in \text{int}(\mathbf{U})\}$$

where $P(\cdot|\beta)$ is given by (3.7) and extended to $x^n \in \mathbf{E}^n$ by taking the product distribution. This indexing of elements of $\mathcal{M}_{me}$ by the corresponding β is called the 'natural' parameterization [82].

Exponential Families and Maximum Entropy Model Classes A k-parameter *exponential family* is a family of probability distributions or densities that can be written in the form

$$P(x|\beta) = \frac{1}{Z(\beta)} e^{-\beta^T \cdot \phi(x)} h(x) \tag{3.11}$$

where $\phi(x) = (\phi_1(x), \ldots, \phi_k(x))$, $\beta \in \mathbf{R}^k$ and $Z(\beta) = \sum_{x \in \mathbf{E}} \exp(-\beta^T \phi(x)) h(x)$. $h(x)$ and $\phi(x)$ are functions defined for all $x \in \mathbf{E}$. The *natural parameter space* of an exponential family is given by

$$\Gamma_{nat} = \{\beta \in \mathbf{R}^k \mid Z(\beta) < \infty\}$$

An exponential family is said to be *full* if it contains a model for every $\beta \in \Gamma_{nat}$. The *dimension* of an exponential family is the dimension of its associated Γ_{nat}. An exponential family is said to be of *irreducible dimension* if there is no $(k-1)$-parameter exponential family expressing the same class of probability distributions.

From the point of view of measure theory, the function $h(x)$ may be absorbed in a dominating measure [82]. One can then drop the factor $h(x)$ from (3.11). We see that maximum entropy model classes and exponential families essentially coincide.

Some Useful Facts about Maximum Entropy Model Classes The following proposition lists some useful (and well-known) facts about maximum entropy/exponential model classes that will be used several times in the chapters to come.

Proposition 3.9 *Let $\mathcal{M}_{me}$ be a maximum entropy class for function $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$ with range $\mathbf{U} = \mathbf{U}_1 \times \ldots \times \mathbf{U}_m$. Let $\beta = (\beta_1, \ldots, \beta_m) \in \Gamma_{nat}$, where Γ_{nat} is the space of parameters in the natural parameterization of $\mathcal{M}_{me}$. Let $1 \leq i, j \leq m$. Then*

1. *The first two (central) moments of $P(\cdot|\beta)$ are determined by the first two derivatives of $Z(\beta)$:*

$$\begin{aligned}\frac{\partial}{\partial \beta_i} \ln Z(\beta) &= -E_\beta[\phi_i(X)] \\ \frac{\partial^2}{\partial \beta_i \partial \beta_j} \ln Z(\beta) &= \operatorname{cov}[\phi_i(X), \phi_j(X)] = \\ &= E[(\phi_i(X) - E[\phi_i(X)])(\phi_j(X) - E[\phi_j(X)])].\end{aligned}$$

2. *Let $\beta_1, \ldots, \beta_m$ all be fixed except β_i. Then*
   (a) *$E_\beta[\phi_i(X)]$ as a function of β_i is strictly decreasing.*
   (b) *If $\beta_i > 0$, then $\mathcal{H}(\beta)$ is a strictly decreasing function of β_i. If $\beta_i < 0$, then $\mathcal{H}(\beta)$ is a strictly increasing function of β_i.*

3. *The log-likelihood* $\ln P(x^n|\beta)$ *as a function of* β_i *is* concave, *reaching its maximum at the point where $E_\beta[\phi_i(X)] = \overline{\phi_i(x)}^n$. More generally:*

4. *Let E_θ stand for the expectation under the model $P(\cdot|\theta) \in \mathcal{M}_{me}$ defined by the mean-value parameterization (i.e. $E_\theta[\phi(X)] = \theta$). Assume that $\overline{\phi(x)}^n$ lies in the interior of* $\mathbf{U}$. *Then:*

$$E_{\hat{\beta}(x^n)}[\phi(X)] = E_{\hat{\theta}(x^n)}[\phi(X)] = \overline{\phi(x)}^n = \hat{\theta}(x^n) \tag{3.12}$$

5. *$\theta(\beta) := E_\beta[\phi(X)]$ as a function of β is a continuous bijection from Γ_{nat} to* $\operatorname{int}(\mathbf{U})$.

Proof: All of these properties are straightforward to verify by differentiation, and realizing that when we take derivatives of $Z(\beta)$ we are allowed to swap the order of differentiation and integration by our regularity conditions on ϕ. Otherwise, see [82], Chapter 1. □

3.5 The Concentration Phenomenon

The Maximum Entropy distributions[1] have a very special property: *almost all outcomes that satisfy a given empirical constraint have frequencies extremely close to the maximum entropy probabilities.* Jaynes [75] has termed this the 'concentration phenomenon'. It shows that, among all distributions that one may consider as candidates for having generated the actual outcomes, the maximum entropy distribution has a unique status. This is used by some as a justification of the maximum entropy principle (though others reject it, as will be discussed in the next section).

Below we show how the concentration phenomenon arises. It will allow us to prove some of the claims we have made in earlier chapters, and it will play an important

[1] At least in the case of discrete sample spaces.

role in Chapter 4. We only consider finite sample spaces $\mathbf{E} = \{1, \ldots, k\}$. Let P_{me} be the maximum entropy distribution for the distributional constraint $E[\phi(X)] = t$ that corresponds to the empirical constraint $\overline{\phi(x)}^n = t$. For simplicity, we make the following assumption: we assume that for each n, there exist x^n such that the empirical constraint $\mathcal{C}_e(\phi, t)$ holds. The arguments can be easily extended to the situation where this is not the case.

Notation Concerning Frequencies We first introduce some notation and terminology regarding frequencies. A tuple of positive rational numbers $\gamma = (\gamma_1, \ldots, \gamma_k)$ is called a *frequency distribution* if $\sum_{i=1}^n \gamma_i = 1$. For a given sequence x^n, we let n_i (where $1 \leq i \leq k$) stand for the number of times that outcome i occurs in x^n. Hence $\gamma_i = n_i/n$. We write $\gamma(x^n) = (\gamma_1(x^n), \ldots, \gamma_n(x^n)) = (n_1/n, \ldots, n_k/n)$ to denote the tuple of frequencies belonging to the set x^n.

Clearly, a sequence x^n satisfies constraint $\overline{\phi(x)}^n$ if and only if

$$\gamma_1(x^n)\phi(1) + \ldots + \gamma_k(x^n)\phi(k) = t. \tag{3.13}$$

Hence whether or not a sequence satisfies a constraint is completely determined by the frequency distribution γ. We say that γ *satisfies* constraint $\overline{\phi(x)}^n$ iff (3.13) holds.

The Concentration Phenomenon Let us define

$$\mathbf{G}^n(\gamma) = \mathbf{G}^n(\gamma_1, \ldots, \gamma_k) = \{x^n | \gamma(x^n) = \gamma\}.$$

In words, $\mathbf{G}^n(\gamma)$ stands for the set of sequences in $\mathbf{E}^n$ with frequencies γ. Observe that

$$|\mathbf{G}^n(\gamma)| = \binom{n}{n_1, \ldots, n_k} = \frac{n!}{n_1! \cdots n_k!} \tag{3.14}$$

Straightforward application of Stirling's approximation,

$$\ln n! = n \ln n - n + \ln \sqrt{2\pi n} + O(\frac{1}{n}), \tag{3.15}$$

together with (3.14) gives:

$$|\mathbf{G}^n(\gamma)| = e^{n\mathcal{H}(\gamma_1, \ldots, \gamma_k)} \cdot c(n) \tag{3.16}$$

Here $\mathcal{H}(\gamma_1, \ldots, \gamma_k)$ stands for the empirical entropy[2] of γ; $c(n) = e^{C/n}$ for some constant C and hence goes to 1 with increasing n.

Now define $\mathbf{C}^n = \{x^n \mid \overline{\phi(x)}^n = t\}$ as the set of data x^n for which the constraint $\overline{\phi(x)}^n$ holds. Let $\gamma = (\gamma_1, \ldots, \gamma_k)$ be two sets of frequencies that satisfy the given constraint (so that (3.13) holds for both γ and γ'). Note that both $\mathbf{G}^n(\gamma)$ and $\mathbf{G}^n(\gamma')$ are subsets of $\mathbf{C}^n$. Suppose $\mathcal{H}(\gamma) < \mathcal{H}(\gamma')$. By (3.16), the ratio between the number of

[2]The *empirical entropy* $\mathcal{H}(\gamma) = \mathcal{H}(\gamma_1, \ldots, \gamma_k)$ is defined analogously to the entropy for probability distributions: $\mathcal{H}(\gamma_1, \ldots, \gamma_k) = -\sum_{i=1}^k \gamma_i \ln \gamma_i$.

elements in $\mathbf{C}^n$ with frequencies γ and the number of elements in $\mathbf{C}^n$ with frequencies γ' decreases exponentially with n:

$$\frac{|\mathbf{G}^n(\gamma)|}{|\mathbf{G}^n(\gamma')|} \rightarrow \frac{\exp(n\mathcal{H}(\gamma))}{\exp(n\mathcal{H}(\gamma'))} = e^{-n\alpha} \text{ for some } \alpha > 0. \tag{3.17}$$

We denote by γ_{me} the tuple of frequencies that maximizes the empirical entropy $\mathcal{H}(\gamma)$ subject to the constraint $\overline{\phi(x)}^n = t$. By (3.17), as n increases, the number of elements in $\mathbf{C}^n$ with frequencies γ_{me} becomes exponentially larger than the number of elements in $\mathbf{C}^n$ with frequencies γ for every fixed $\gamma \neq \gamma_{me}$. This is the 'concentration phenomenon' [75]. It says that nearly all outcomes x^n satisfying constraint $\overline{\phi(x)}^n = t$ will have empirical frequencies $(n_1/n, \ldots, n_k/n)$ extremely close to the maximum entropy frequencies $\gamma_{me} = (\gamma_{me,1}, \ldots, \gamma_{me,k})$. These in turn are equal to the maximum entropy *probabilities* $(P_{me}(1), \ldots, P_{me}(k))$ for constraint $E[\phi(X)] = t$.

The concentration phenomenon enables us to prove some claims we made earlier: (3.16) shows that Equation 1.9 of Chapter 1, page 18 holds. Also, (3.17) shows the validity of the claim that the ratio of $\binom{n}{\gamma n}$ to 2^n goes to 0 exponentially fast for all γ except $\gamma = 1/2$ (page 7).

In the next section, we show how the concentration phenomenon may be used to justify the Maximum Entropy Principle.

3.6 Pros and Cons of MaxEnt

Jaynes first published about maximum entropy in 1957. The idea as a whole and its range of applicability has been a topic of fierce debate ever since. Perhaps the most impressing argument in favor of MaxEnt is the simple fact that it works very well in many applications [113, 16, 77]. However, we would like to know why. There are lots of arguments in favor of MaxEnt; but there are just as many against it. We will sketch some of the most well-known of these arguments. Before presenting them, we give Jaynes' example of the 'Brandeis Dice'. This is the standard example of a 'toy' MaxEnt application. Opponents and proponents of the principle alike use it to illustrate their arguments.

Example 3.10 [Brandeis Dice] Let a six-sided die be given, together with the additional information that the expected number of spots coming up in a throw is 4.5 rather than 3.5. We let p_i stand for $P_{me}(X = i)$. In other words, $\mathbf{E} = \{1, 2, 3, 4, 5, 6\}$ and $E[X] = \sum_{i=1}^{6} ip_i = 4.5$. Jaynes [74, 76] calculated the MaxEnt probabilities as follows:

$$(p_1, \ldots, p_6) = (0.05435, 0.07877, 0.11416, 0.16545, 0.23977, 0.34749) \tag{3.18}$$

The question is now, of course, how to interpret these probabilities.

3.6.1 What's good about MaxEnt

One can justify MaxEnt in many ways [77, 114, 115, 144, 153, 152]. Most of these justifications fall in either of two categories. We discuss these in turn.

Almost all outcomes are typical for the MaxEnt distribution The approaches belonging to the first category are all based on the concentration phenomenon we discussed in the previous section. We first show how the concentration phenomenon can be used as a direct justification of maximum entropy. The justifications work only for the case of finite sample spaces $\mathbf{E} = \{1, \ldots, k\}$. Let x^n be some sequence. Suppose that we are given a constraint $\overline{\phi(x)}^n = t$ for some large n. By the concentration phenomenon we know that the overwhelming majority of x^n that satisfy the constraint will have frequencies of occurrence $(\gamma_1, \ldots, \gamma_k)$ (where γ_i is the number of times outcome i occurs in x^n) extremely close to the maximum entropy probabilities $(P_{me}(1), \ldots, P_{me}(k))$. For lack of any more specific knowledge, it may now seem reasonable to infer as a 'best guess' that the frequencies really will be very close to the maximum entropy probabilities. Going one step further, one may also infer that the maximum entropy distribution is a 'best guess' as a model for the data.

This argument consists of two steps, both of which may be subject to criticism: the first step is the 'inductive principle' that the frequencies that can be realized in the greatest number of ways are a best guess of the actual frequencies. The second is that these frequencies may then be identified with probabilities. The first step can be seen as a straightforward extension of Laplace's Principle of Insufficient Reason. Hence *if* one accepts this principle, *then* one may be inclined to accept the first step. Namely, let us assume, in line with the Principle of Insufficient Reason, that all sequences x^n satisfying the given constraint are equally likely (hence uniformly distributed). Then the concentration phenomenon tells us that with overwhelming ***probability***, the frequencies will indeed be close to the maximum entropy probabilities.

Rationality Requirements In the other category of MaxEnt justifications, one postulates a number of axioms that should be satisfied by any rational inference procedure. A typical example would be an axiom expression 'permutation invariance': giving different names to outcomes should not affect the results of the inference procedure.

Shore and Johnson [144] formulated this 'permutation invariance' and several other, similarly innocuous-seeming 'rationality requirements'. They were able to show that the only procedure satisfying the requirements is to adopt the maximum entropy distribution for the given constraints. These results were later extended by other writers [115, 153] who gave even simpler and even more intuitive axioms and showed that only maximum entropy satisfies them. It should be noted though that Uffink [157, 158] demonstrates that some of the requirements may not necessarily be so 'rational' as it seems, and that in some of the approaches, there are hidden additional assumptions used in proving the unique status of MaxEnt.

3.6.2 A Problem with MaxEnt: Ex Nihilo Nihil

MaxEnt has been criticized on lots of different grounds by many different people [130, 49, 141, 116, 157, 158]. It would much too ambitious to try and list all objections here. Instead, we focus on a single, 'classic' one, that stands at the basis of many (though by no means all) of the other ones. Here it is:

If you are only given the information that P satisfies constraint $E_P[\phi(X)] = t$ for some ϕ and t, how then, in principle, can you conclude anything more than what follows logically from the constraint? By a 'logical consequence' we mean everything that can be derived from $E_P[\phi(X)] = t$ in conjunction with the axioms of probability theory (e.g. Kolmogorov's axioms [85]).

In the Brandeis dice example, how can you decide upon distribution (3.18)? The distribution with $p_4 = p_5 = 1/2$ and all other probabilities zero will satisfy the constraints just as well, but it will lead to completely different predictions! Never mind – from this point of view – that MaxEnt is the only inference procedure satisfying some rationality requirements, or that the frequencies corresponding to the p_i's can be realized in the greatest number of ways: the very *attempt* to arrive at a complete probability distribution from partial knowledge is flawed!

This argument goes back to Leslie Ellis ([44]; see also [157, 158]) who objected on the same grounds to Laplace's Principle of Insufficient Reason. In 1850 he coined the phrase 'ex nihilo nihil' (literally, 'nothing out of nothing'). We quote from [44]: 'Mere ignorance is no ground for any inference whatsoever. *Ex nihilo nihil.* It cannot be that because we are ignorant of the matter we know something about it.'

In Chapter 4, Section 4.2 we propose a 'weakened' version of Maximum Entropy that partially answers the 'Ex Nihilo Nihil' argument given above. But in order to do this, we first have to establish the relation between MDL and Maximum Entropy. This is done in the following section.

3.7 Maximum Entropy as a Special Case of MDL

In the previous sections we have introduced the Maximum Entropy Principle. We now relate it to the Minimum Description Length Principle. The connection between the two principles can be understood from two points of view. The first point of view is to assume that the given constraints pertain to some otherwise unknown distribution generating the data. In this setting, the code based on the maximum entropy distribution turns out to yield the shortest worst-case expected code length, where 'worst-case' is defined with respect to all generating distributions that satisfy the constraint. The second point of view is to make no distributional assumptions and to assume that one is given an empirical constraint concerning actual data; it turns out that in this case too, a close connection between MDL and Maximum Entropy exists. Here, we focus on the first point of view, essentially following [128]. The second point of view (no distributional assumptions) is explored in [46, 93].

3.7.1 The Maximum Entropy Distribution Minimizes the Maximum Expected Codelength

Let us assume then that data are actually generated by repeated sampling of some 'true' distribution P^* satisfying the constraint $E[\phi(X)] = t$. Let $\mathbf{M}(\phi, t)$ be the class of all probability distributions over $\mathbf{E}$ that satisfy this constraint. If we code the data

Let Q and R be two distributions over $\mathbf{E}$ satisfying $E[\phi(X)] = t$. Let P_{me} be the maximum entropy distribution for this constraint. We have:

$$\begin{aligned} \mathcal{H}(Q) &\overset{(1)}{=} E_Q[-\ln Q(X)] \overset{(2)}{\leq} E_Q[-\ln P_{me}(X)] \\ &\overset{(3)}{=} E_{me}[-\ln P_{me}(X)] \\ &\overset{(4)}{=} \mathcal{H}(P_{me}) \overset{(5)}{\leq} E_{me}[-\ln R(X)] \end{aligned} \tag{3.19}$$

If $Q \neq P_{me}$, inequality (2) becomes strict; if $R \neq P_{me}$, inequality (5) becomes strict.

(1) and (4) follow from the definition of entropy. (2) and (5) follow from the information inequality (3.4) on page 52. (3) follows from equations (3.9) and (3.10) on page 54. (1)-(5) imply $\mathcal{H}(Q) \leq \mathcal{H}(P_{me})$ which expresses the fact that P_{me} maximizes entropy. It is also implied that $E_Q[-\ln P_{me}(X)] \leq E_{me}[-\ln R(X)]$ which expresses the fact that P_{me} minimizes worst-case description length.

Figure 3.1: Maximum Entropy and Minimum Description Length

using the code based on P_{me}, the maximum entropy distribution for this constraint, then we see from Figure 3.1 (which summarizes equations (3.9) and (3.10) of page 54) that the expected code length of $x^n = (x_1, \ldots, x_n)$ becomes

$$E_{P^*}[-\ln P_{me}(X_1, \ldots, X_n)] = n\mathcal{H}(P_{me}). \tag{3.20}$$

This quantity is independent of the 'true' distribution P^*.

Let us now consider the special case in which the 'true' distribution is P_{me}. For every $R \neq P_{me}$, the code based on R satisfies (by Figure 3.1, Inequality (5)):

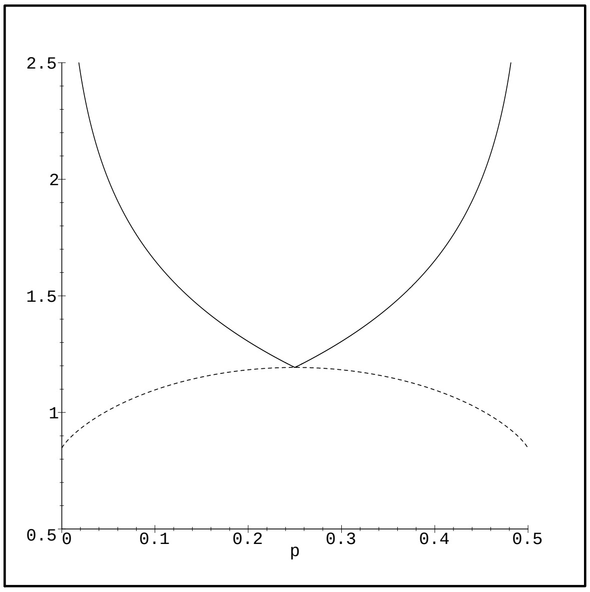
$$E_{me}[-\ln R(X_1, \ldots, X_n)] = n(\mathcal{H}(P_{me}) + \epsilon) \text{ for some } \epsilon > 0,. \tag{3.21}$$

and hence the extra code length needed to encode n outcomes due to using R rather than P_{me} is on the order of n. It follows that when the only information that is given is that the data are distributed according to a distribution P^* satisfying the given constraint, then P_{me}, the maximum entropy distribution for this constraint, minimizes *the worst-case expected code length* among all possible probability distributions over the same sample space:

$$P_{me} = \arg\inf_{P} \sup_{P^* \in \mathbf{M}(\phi,t)} E_{P^*}[-\ln P(X)] \tag{3.22}$$

This is shown graphically in Figure 3.2, where the specific example of the probability distributions over $\mathbf{E} = \{a, b, c\}$ are used to illustrate the general phenomenon.

Connection to MDL What is the relation between 'minimizing worst-case expected code length' and MDL? The strong law of large numbers [47] says that if data are



The figure depicts both the entropy and the worst-case expected description length of distributions P over $\{a, b, c\}$ satisfying the constraint $P(a) = 1/2$. The x-axis depicts $P(b)$ and ranges over all distributions for which the constraint holds. The dashed curve depicts $\mathcal{H}(P)$, the solid curve depicts $\max_{P^* \in \mathbf{M}(\phi,t)} E_{P^*}[-\ln P(X)]$. The curves touch at $P = P_{me}$ which gives both the maximum entropy and the minimum worst-case expected description length. According to MDL we should pick P_{me} because it minimizes the upper curve, *not* because it maximizes the lower curve. Note that

$$E_{P^*}[-\ln P(X)] = -P^*(a) \ln P(a) - P^*(b) \ln P(b) - P^*(c) \ln P(c).$$

The term $-P^*(a) \ln P(a)$ must be equal to $-1/2 \ln(1/2)$ for all P^* and P that satisfy the constraint. For $P(b) < 1/4$, the worst-case description length is reached for generating distribution $P^*(b) = 1/2; P^*(c) = 0$. For $P(b) > 1/4$, it is given by $P^*(b) = 0; P^*(c) = 1/2$. For $P(b) = P_{me}(b) = 1/4$, every P^* satisfying the constraints yields the same expected description length. The worst-case description length $\max_{P^* \in \mathbf{M}(\phi,t)} E_{P^*}[-\ln Q(X)]$ for distributions Q *not* satisfying the constraint is not depicted, but by equations 3.20 and 3.21 it is easily seen to be larger than $\max_{P^* \in \mathbf{M}(\phi,t)} E_{P^*}[-\ln P_{me}(X)] = E_{me}[-\ln P_{me}(X)]$.

Figure 3.2: The entropy vs. the worst-case expected description length.

drawn by repeated sampling according to a fixed distribution P^*, then, for every coding distribution P, with probability 1 the average code length $n^{-1} \sum_{i=1}^{n} -\ln P(x_i)$ will converge to its expected value $E_{P^*}[-\ln P(X)]$. Hence if we are willing to assume that the data are drawn according to some unknown distribution P^* satisfying the given constraint, and we have no idea as to whether there are any other constraints underlying the data, then modeling our data using (the code based on) P_{me} is in the spirit of the MDL Principle.

It would go too far to say though that the MDL Principle *tells* us to use this code. The reason is that the code based on P_{me} is not *necessarily* the best one to use in this scenario: there are many distributions P' for which, for all n, all $x^n \in \mathbf{E}^n$, $-\ln P'(x^n) \leq -\ln P_{me}(x^n) + c$ for an arbitrarily small constant c. If data are generated by independent sampling from some P^* satisfying the constraints, then the average code length of the code based on P' will converge to some value $L \leq E_{P^*}[-\ln P_{me}(X)]$. As an example, consider the distribution P' with $P'(x^n) = (1/2)P_{me}(x^n) + (1/2)P''(x^n)$ where $P''(x^n)$ is some completely different distribution. Then $-\ln P'(x^n) \leq -\ln P_{me}(x^n) + \ln 2$. If the data happens to be generated by a P^* such that $E_{P^*}[-\ln P''(X)] < E_{P^*}[-\ln P_{me}(X)]$ then the code based on P' will give (with P^*-probability 1) shorter average code lengths than the code based on P_{me}; if P^* is such that $E_{P^*}[-\ln P''(X)] \geq E_{P^*}[-\ln P_{me}(X)]$ then the average code length based on P' will (with P^*-probability 1) converge to the length based on P_{me}.

3.8 Conclusion and Outlook

We have introduced the Maximum Entropy Principle and shown its relation to MDL. In the next chapter we will discuss an important property of maximum entropy model classes; in Chapter 5 the connection between MDL and Maximum Entropy will be exploited to optimize MDL's trade-off between hypothesis complexity and goodness-of-fit.

Chapter 4

Safe Statistics

This Chapter introduces the notion of *reliable* inferences, leading to 'safe statistics'. Section 4.1 introduces reliable inferences in the context of maximum entropy model classes. Reliable inferences allow one to make good predictions and decisions regarding future data under a much wider variety of assumptions than do 'unreliable' inferences. This is extended in Section 4.2 where two forms of applying maximum entropy are distinguished: *safe* and *risky* maximum entropy. The distinction applies not only to maximum entropy, but to statistical procedures in general.

All this is done in preparation of Chapter 5, where 'safe statistics' will be put to use. Essentially, it will allow us to establish in what way we can and in what way we cannot use overly simple models (or, more generally, models that achieve compression but are not in any way related to the 'true' model generating the data). In order to keep the treatment as general as possible, in this chapter we abstract away from the specific statistical procedure used and do not focus specifically on MDL. In the next chapter 'safe statistics' will be connected to MDL.

4.1 You can trust Maximum Entropy models (in some respects)

This section introduces the central ideas of this chapter. We start by noting an extremely important property shared by all Maximum Entropy model classes. Let $\mathcal{M} = \{P_{me,\phi}(\cdot|\theta) \mid \theta \in \mathbf{U}\}$ be a class of maximum entropy distributions for function $\phi : \mathbf{E} \rightarrow \mathbf{U}$ with $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$. Let $x^n \in \mathbf{E}^*$ be an arbitrary data sequence such that $\overline{\phi(x)}^n$ lies in the interior of $\mathbf{U}$. The following proposition says that the expected value under $\hat{\theta} = \hat{\theta}(x^n)$ of linear transformations of ϕ is equal to their average value over x^n. In particular, this implies that the expected code length according to the maximum likelihood model $\hat{\theta}(x^n)$ is equal to the actual average code length achieved by $\hat{\theta}(x^n)$. More precisely,

Proposition 4.1 *Let $\mathcal{M}$, ϕ, x^n and $\hat{\theta}(x^n)$ be as above.*

For arbitrary $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_m) \in \mathbf{R}^{m+1}$*, let* $\psi(x) = \sum_{i=1}^{m} \alpha_i \phi_i(x) + \alpha_0$*. We have:*

$$\begin{aligned} E_{\hat{\theta}}[\psi(X)] &= \overline{\psi(x)}^n && (4.1) \\ E_{\theta}[\psi(X)] &\neq \overline{\psi(x)}^n \textit{ for all } \theta \in \mathbf{U} \textit{ with } \theta \neq \hat{\theta}. && (4.2) \end{aligned}$$

In particular this implies:

$$E_{\hat{\theta}}[-\ln P(X|\hat{\theta})] = -\frac{1}{n} \ln P(x^n|\hat{\theta}) \quad (4.3)$$

Proof: (4.1) and (4.2) are immediate from Proposition 3.9 (page 56) and the fact that both expectation and averaging are linear operations. To prove (4.3), observe that, from the natural parameterization (Equation 3.7, page 53) of a maximum entropy model $P(\cdot|\beta)$ one can see that there exists a β such that for all $x \in \mathbf{E}$, it holds that $-\ln P(x|\theta) = \beta\phi(x) + Z(\beta)$. Hence there exist α_0, α_1 such that for all $x \in \mathbf{E}$, $-\ln P(x|\theta) = \alpha_0 \phi(x) + \alpha_1$, which, by (4.1) and independence ($\sum_{i=1}^{n} \ln P(x_i) = \ln P(x^n)$) implies (4.3). □

This proposition has a very important consequence which is best introduced by means of an example.

Example 4.2 [Brandeis once more] As in Chapter 3, Example 3.10, consider a six-sided die. This time, let the sequence of throws x^n consist of $n/2$ 4's and $n/2$ 5's, but suppose we are only given the partial information that $\overline{x}^n = 4.5$. Let $\mathcal{M}_{me}$ be the maximum entropy model class for constraints $E[X] = t$ where $1 < t < 6$. By (4.1) and (4.2) the maximum likelihood model $P(\cdot|\hat{\theta})$ in the model class $\mathcal{M}_{me}$ is equal to the maximum entropy model P_{me} for the constraint $E[X] = 4.5$. The probabilities $P_{me}(X = i)$ were given on page 59; we repeat them for convenience:

$$(p_1, \ldots, p_6) = (0.05435, 0.07877, 0.11416, 0.16545, 0.23977, 0.34749)$$

Clearly, these probabilities do not coincide with the frequencies of $1, \ldots, 6$ in the sequence x^n. Still, it holds that

$$-\ln P(x^n|\hat{\theta}) = -\frac{n}{2} \ln p_4 - \frac{n}{2} \ln p_5 = E_{\hat{\theta}}[-\ln P(X^n|\hat{\theta})] = n\mathcal{H}(\hat{\theta})$$

where the second equality follows from (4.3) in the proposition above but can also be checked numerically. If we had used the model $\hat{\theta}$ all the time for prediction (for example, because we already knew from some previous data that $E[X] = 4.5$), and we had predicted using the logarithmic loss $\textsc{loss}_{lg}(e; \hat{\theta}) = -\ln P(e|\hat{\theta})$ (Chapter 2, Section 2.7), then our prediction error would have been *exactly equal to what we expected it to be* (namely $E_{\hat{\theta}}[-\ln P(X^n|\hat{\theta})]$) even though the *frequencies* of actual outcomes $(\gamma_1, \ldots, \gamma_6) = (0, 0, 0, 1/2, 1/2, 0)$ would have been very different from their expectations $E_{\hat{\theta}}[\mathcal{I}(X = 1)] = p_1, \ldots,$ $E_{\hat{\theta}}[\mathcal{I}(X = 6)] = p_6$. (here $\mathcal{I}$ denotes the indicator function defined as in Equation 3.2 on page 51).

From Chapter 2, Section 2.7 we know that the logarithmic loss with respect to a probability distribution P can be interpreted as a generic kind of loss function, but also as the number of bits needed to encode an outcome using the code based on P and as the accumulated loss in capital when betting proportionally on the basis of P.

It follows that, if we had placed bets on $x_1, \ldots, x_n$ proportionally on the basis of $\hat{\theta}$, the (logarithm of the) accumulated loss we would have made would have been exactly as large as we would have expected it to be on the basis of $\hat{\theta}$. This holds even though other aspects of x^n (like the frequencies) may be completely different from what we would expect them to be on the basis of $\hat{\theta}$. Note that the equality of expected and actual loss does not just hold for our particular x^n with $n/2$ 4's and $n/2$ 5's; rather, by (4.3) in Proposition 4.1 above it holds for *every* x^n with $\overline{x}^n = 4.5$. Hence it also holds for many less 'suspicious' samples for which $\gamma_i > 0$ for each $1 \leq i \leq 6$.

Reliable Prediction Apparently, some aspects of the data (code length, prediction loss in proportional betting, the average of x) can be *reliably* predicted on the basis of the maximum entropy model while others (for example, the frequencies of the outcomes) cannot. This is the central point of this chapter.

In general, we will be interested in what can be reliably predicted and what not from a model that is only partially correct. In the present example, we assumed that we were only given the information that $\overline{x}^n = 4.5$; therefore, we could not simply estimate the underlying probability distribution by setting the probabilities equal to the frequencies and we were forced to use a model that is potentially incorrect. In other examples, the whole sequence x^n may be available, but our model class does not contain a model that captures all of the regularity in the data. For example, in Chapter 1, Section 1.5, we showed the case of data being generated by a Markov chain with stationary distribution 1/3 that was modeled by a Bernoulli distribution. In such a situation, we would like to be able to infer that the optimal Bernoulli model for the data, $\hat{\theta} \approx 1/3$ can be reliably used to predict the frequency of 1s in future data but not to predict the frequency of 11 (two 1's in a row) (page 24).

We will now put the central idea of the example in more abstract terms. Suppose we are trying to learn the parameters of a probability distribution from a sample $x_1, \ldots, x_k$. We then use the values we found to predict future data, say $x_{k+1}, \ldots, x_n$. Clearly, the model in the class that fits best the future data is $\hat{\theta}(x_{k+1}, \ldots, x_n)$. By the arguments in Chapter 2, Section 2.2 this is also the model that, once it is given, allows us to code $x_{k+1}, \ldots, x_n$ using a minimal number of bits; finally, had we predicted $x_{k+1}, \ldots, x_n$ on the basis of $\hat{\theta}(x_{k+1}, \ldots, x_n)$ we would have obtained the minimal possible loss in the game of proportional betting described in Chapter 2, Section 2.7. We see that this model is optimal for data $x_{k+1}, \ldots, x_n$ in several different ways. We will call this 'model that will turn out to be the best' the *future-optimal model* and denote it by $\hat{\theta}_{\text{fut}}$.

Suppose now that for some reason or other, our guess $\ddot{\theta}$ based on $x_1, \ldots, x_k$ is equal to $\hat{\theta}_{\text{fut}} = \hat{\theta}(x_{k+1}, \ldots, x_n)$. In other words, by some coincidence we hit upon the model that will turn out to yield the best predictions of future data. Now if our model class is a maximum entropy model class for function $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$ and we use $\ddot{\theta} = \hat{\theta}_{\text{fut}}$ then not only will our prediction error (as measured in logarithmic loss) be minimal, the error will also be *exactly* equal to what we expect it to be: according to $\hat{\theta}_{\text{fut}}$, our expected error will be $E_{\hat{\theta}_{\text{fut}}}[-\ln P(X_{k+1}, \ldots, X_n | \hat{\theta}_{\text{fut}})]$. By Proposition 4.1

this coincides with the error $-\ln P(x_{k+1},\ldots,x_n|\hat{\theta}_{\text{fut}})$ which we will actually make. The proposition further implies that if we estimate the averages $\overline{\phi_1(x)},\ldots,\overline{\phi_m(x)}$, by $E_{\hat{\theta}_{\text{fut}}}[\phi_1(X)],\ldots,E_{\hat{\theta}_{\text{fut}}}[\phi_m(X)]$ respectively, then our estimates will be perfect.

Reasonable Inference Procedures In a more realistic situation, if given enough initial data $x_1,\ldots,x_k$ and a good procedure for estimating $\ddot{\theta}$ (for example, two-part code MDL, Chapter 1, Section 1.3), then $\ddot{\theta}$ may be quite close, but not equal to $\hat{\theta}_{\text{fut}}$. In Chapter 5.2, Section 5.3 (Lemma 5.14) it will be shown formally that, if $\ddot{\theta}$ is based on a large enough sample and data are generated by independent sampling according to some arbitrary distribution P^*, then under fairly general conditions on the model class $\mathcal{M}$, exactly this situation will occur with probability 1.

It is this situation that we will study in this section and the next: the results we present will only be useful if the assumption that the estimator $\ddot{\theta}$ will be close to $\hat{\theta}_{\text{fut}}$ is justified. To keep the discussion general, we will not restrict ourselves to MDL estimators. Rather, we will assume that we use some reasonable inference procedure that, for those sequences $x_1,x_2,\ldots$ that are such that there really is 'something to infer', is guaranteed to work well for large enough samples:

Definition 4.3 *Let $\mathcal{M}$ be a model class that is finitely parameterized by some Γ. Let $\mathcal{L}_{\mathcal{M}}$ be an estimation procedure that, for each n, $x^n \in \mathbf{E}^n$ outputs an estimator $\ddot{\theta}(x^n) \in \Gamma$. We call $\mathcal{L}_{\mathcal{M}}$* reasonable *if for every sequence $x_1,x_2,\ldots$ for which $\hat{\theta}(x^n)$ converges to some value, $\ddot{\theta}(x^n)$ converges to that same value; that is*

> if $\lim_{n\rightarrow\infty}\hat{\theta}(x^n)$ *exists and is equal to $\hat{\theta}_{fut}$ for some $\hat{\theta}_{fut} \in \Gamma$,*
> then $\lim_{n\rightarrow\infty}\ddot{\theta}(x^n)$ *must also exist and be equal to $\hat{\theta}_{fut}$.*

Hence if the model in $\mathcal{M}$ that maximizes the likelihood converges to some value $\hat{\theta}_{\text{fut}}$ then a reasonable estimator should converge to that same value. Note that we assume $\mathcal{M}$ to be finitely parameterized. It can then be seen by the argument in Chapter 1, Section 1.4.1 that the 2-part code MDL estimator and the ML estimator will converge to the same value, hence the MDL estimator is reasonable. Similarly, one can show that (if a reasonable prior is used) the Bayesian MAP estimator and of course, the ML estimator itself are 'reasonable'. Our definition of 'reasonable estimator' does *not* mean that we think the ML estimator is superior to the other estimators; we do not rule out the possibility that for finite n, a different estimator $\ddot{\theta}(x^n)$ may in some cases with high probability be closer to $\hat{\theta}_{\text{fut}}$ than $\hat{\theta}(x^n)$. The point is that *if* there exists a value $\hat{\theta}_{\text{fut}}$ to which $\hat{\theta}(x^n)$ converges *then* this value will in the limit for large n fit the data better than any other model in $\mathcal{M}$. For this reason we demand 'reasonable' estimators to converge to $\hat{\theta}_{\text{fut}}$.

Guaranteed Performance Since for every maximum entropy model class $\mathcal{M}_{me}$, the expectation $E_\theta[\phi(X)]$ is a continuous function of θ (Proposition 3.9), it follows by Proposition 4.1 that in a situation in which the estimate $\ddot{\theta}(x^k)$ based on x^k is close enough to the future-optimal $\hat{\theta}(x_{k+1},\ldots,x_n)$:

1. The expected value of (every linear combination of) the functions $\phi_1, \ldots, \phi_m$ under the estimator $\ddot{\theta}(x^k)$ will be close to the actual average value they will get on future data $x_{k+1}, \ldots, x_n$.

2. In particular, since the logarithmic loss is a linear combination of the functions $\phi_1, \ldots, \phi_m$, the expected value of logarithmic loss under $\ddot{\theta}$ will be close to the actual average logarithmic loss that will be achieved when predicting future data.

In order to get such a guaranteed performance, it is essential that

1. We use a maximum entropy model class, and

2. We only consider functions ψ that can be written as linear combinations of the ϕ_i.

In Chapter 5.2, Example 5.1 we shall see an example of a non-maximum entropy class for which it can happen that even though $\ddot{\theta}$ is a very good estimate of $\hat{\theta}_{\text{fut}}$ (in the sense that $\ddot{\theta}$ is very close or even equal to $\hat{\theta}_{\text{fut}}$), still $E_{\ddot{\theta}}[\psi(X)]$ is quite different from $\lim_{n\to\infty} \overline{\psi(x)}^n$ for some function $\psi(x)$ that we would like to estimate: the expectation of $\psi(X)$ based on $\ddot{\theta}$ may be quite different from the average value this function will attain on future data. (we have no proof though that this can happen for all functions ψ and all model classes $\mathcal{M}$ that are not maximum entropy classes; Rissanen ([128], page 112) claims that this is the case but provides no proof). Nevertheless, we shall see in the next chapter that if the functions to be predicted depend *themselves* on the model $\ddot{\theta}$ (in other words, ψ is a function of both x and $\ddot{\theta}$) then we can get around the maximum entropy requirement after all.

If we do use a maximum entropy model class $\mathcal{M}_{me}$ for ϕ but consider a function ψ that is not a linear combination of the ϕ_i, then we may get the same disagreement between expectations and averages once more (again, we have no formal proof that this can happen for *all* non-linear ψ and all $\mathcal{M}_{me}$). We have seen an instance of this in Example 4.2 at the beginning of this section. However, at least for discrete maximum entropy model classes, the disagreement between expectations and averages will only take place for an exponentially small fraction of the data: as we saw in the previous chapter (Section 3.5), for large enough n, nearly all outcomes for which $\overline{\phi(x)}^n = t$ will have frequencies (almost) equal to the probabilities of the MaxEnt distribution for the constraint $E[\phi(X)] = t$, which (Proposition 4.1) coincides with the maximum likelihood model in $\mathcal{M}_{me}$ for constraint $\overline{\phi(x)}^n = t$. One verifies immediately that for such sequences, we have $E_{\hat{\theta}}[\psi(X)] \approx \overline{\psi(x)}^n$ for *every* function $\psi : \mathbf{E} \to \mathbf{U}$, and not just for linear combinations of the ϕ_i. Nevertheless in many practical situations the actual sequence x^n will indeed be such that the frequencies are very different from the maximum entropy probabilities (see the next chapter; for a concrete example, see Example 5.24 in the Epilogue to part I, page 120).

In the next section we will construct a framework to formally deal with the notion of 'guaranteed performance'.

4.2 Reliable Decisions

It was shown in the previous section that if the estimate $\ddot{\theta}$ of the parameters of a distribution in a maximum entropy model class $\mathcal{M}_{me}$ is close to the future-optimal model $\hat{\theta}_{\mathrm{fut}}$, then we can reliably estimate averages of some functions over $\mathbf{E}$ while we cannot reliably estimate averages over some other functions. In realistic settings, we would not only like to estimate averages of functions but also, more generally, to arrive at optimal *decisions* concerning future data. How to make such decisions is the subject of decision theory [17, 39]. Below we first review some basic notions of decision theory. We will then define the notions of 'reliable' and 'unreliable' estimates and decisions.

4.2.1 Decision Theory

A standard concept in decision theory is the *loss function*. We will restrict ourselves here to loss functions $\text{LOSS} : \mathbf{E} \times \mathcal{D} \to \mathbf{R} \cup \{\infty\}$. Here $\mathbf{E}$ is a space of possible outcomes and $\mathcal{D}$ is a space of possible decisions. $\text{LOSS}(e; \delta)$

stands for the loss that is incurred if, in a situation where, after the *decision* δ has been made, the outcome e occurs. Decisions have to be made before the data occurs; hence we cannot simply make the decision that minimizes the loss for the actual outcome. Instead, we should make some decision that is likely to lead to a small loss. Let us assume that we model our data using a probability distribution P. P will often be arrived at on the basis of previously seen outcomes (data). We may now pick the δ^* that minimizes our *expected loss*:

$$\delta^* = \delta^*(\text{LOSS}, P) = \arg\min_{\delta} E_P[\text{LOSS}(X; \delta)] \tag{4.4}$$

We call $\delta^*(\text{LOSS}, P)$ the *optimal decision*[1] based on P and $\text{LOSS}(\cdot; \cdot)$.

Predicting is a Special Case of Deciding A 'prediction' can be seen as a special kind of a 'decision'. We consider two kinds of prediction in this thesis: *set-wise* and *point-wise* prediction. We have already encountered set-wise prediction in Chapter 2, Section 2.7. It amounts to dividing a unit investment over the space $\mathbf{E}$ of possible outcomes. Hence you have to assign each $e \in \mathbf{E}$ a weight $w(e)$ such that $\sum_{e \in \mathbf{E}} w(e) = 1$. The loss you incur if the actual outcome is e then depends on both e and the function w. In this case the decision space is the space of all possible weight assignments, which corresponds to the class of all possible probability distributions over $\mathbf{E}$. In Chapter 2, Section 2.7 we considered the logarithmic loss where $\text{LOSS}(e, w) = -\log w(e)$ (we interpreted w as a probability distribution there).

Pointwise prediction works somewhat differently: here we have to specify a single value $\hat{e} \in \mathbf{E}$ and we are then charged with a loss $\text{LOSS}(e, \hat{e})$ where e is the actual outcome. A loss function $\text{LOSS} : \mathbf{E} \times \mathbf{E} \to \mathbf{R} \cup \{\infty\}$ is also called a *distortion function*.

[1] The δ^* optimizing (4.4) is usually called the 'Bayes action' [17]. Strictly speaking, it is a concept belonging to Bayesian, not frequentist decision theory. However, it is also used outside of Bayesian statistics; see for example [166].

An example of a distortion function is the squared error function for a fixed value of x. It takes the form $\text{ER}(y|H,x) = (y - \hat{y})^2$ where $\hat{y} = H(x)$. Hence different values of H lead to different predictions $\hat{y}$ of y.

4.2.2 Reliable Decisions

We will now formally define the notions of reliable estimations and decisions. The following definition captures the idea of reliable estimation as explained on page 67. In the definition, $\text{int}(\mathbf{U})$ stands for the interior of the set $\mathbf{U}$.

Definition 4.4 *Let $\mathcal{M}$ be a class of probabilistic models over $\mathbf{E}$ parameterized by some Γ. Let $\psi : \mathbf{E} \to \mathbf{U}$ be a given function. If, for all n, all $x^n \in \mathbf{E}^n$ with $\hat{\theta}(x^n) \in \text{int}(\Gamma)$, we have*

$$E_{\hat{\theta}(x^n)}[\psi(X)] = \overline{\psi(x)}^n \tag{4.5}$$

and, moreover, $E_\theta[\psi(X)]$ is a continuous function of θ, then we say that averages of ψ can be reliably estimated *on the basis of $\mathcal{M}$. Otherwise we say that averages of ψ cannot be reliably estimated on the basis of $\mathcal{M}$.*

Note that we require $E_{\hat{\theta}}[\psi(X)]$ to be a continuous function of θ. As discussed on page 68, in realistic situations, we will not be able to find an estimator $\ddot{\theta}$ that is equal to the future optimal $\hat{\theta}_{\text{fut}}$; rather, we will have $|\ddot{\theta} - \hat{\theta}_{\text{fut}}| = \epsilon$ for some small ϵ; the continuity requirement makes sure that the estimates $E_{\ddot{\theta}(x^n)}[\psi(X)]$ will still be reasonably close to the average of ψ over future data.

The definition becomes slightly more involved in the decision-theoretic case. We first note that when the loss function LOSS is known from the context, we will write $\delta^*(x^n)$ as short for $\delta^*(\text{LOSS}, P(\cdot|\hat{\theta}(x^n)))$. Hence $\delta^*(x^n)$ is the decision that minimizes the expected loss where the expectation is under the maximum likelihood model for data x^n.

Definition 4.5 *Let $\mathcal{M}$ be as above. Let LOSS be a given loss function with domain $\mathbf{E} \times \mathcal{D}$. Let, for data $x^n \in \mathbf{E}^n$, $\delta^*(x^n)$ be the optimal decision based on Equation 4.4 for loss function LOSS and probability distribution $P(\cdot|\hat{\theta}(x^n))$. If, for all n, all $x^n \in \mathbf{E}^n$ with $\hat{\theta}(x^n) \in \text{int}(\Gamma)$, we have*

$$E_{\hat{\theta}(x^n)}[\text{LOSS}(X; \delta^*(x^n))] = \frac{1}{n} \sum_{i=1}^{n} \text{LOSS}(x_i; \delta^*(x^n)) \tag{4.6}$$

and, moreover, for all $\delta \in \mathcal{D}$, $E_\theta[\text{LOSS}(X; \delta)]$ is a continuous function of θ, then we say that LOSS can be reliably used on the basis of $\mathcal{M}$. Otherwise we say that LOSS cannot be reliably used on the basis of $\mathcal{M}$.

In other words: let x^n be arbitrary but fixed. Then the expected loss under $\hat{\theta}(x^n)$ on the basis of the decision $\delta^*(x^n)$ that minimizes this expected loss is equal to the average loss of $\delta^*(x^n)$ over x^n. So, if we base our decisions to predict $x_1, \ldots, x_n$ on an estimate $\ddot{\theta}$ that turns out to be such that $\ddot{\theta} \approx \hat{\theta}(x^n)$, and we always opt for the decision that minimizes expected loss, then our average loss over x^n will be close to the loss we expect to make based on our decisions (this will be illustrated in Example 4.11 on page 74).

Having defined what functions and loss functions can be reliably estimated (used), we proceed to define reliable *estimates* and *decisions*:

Definition 4.6 *Let $\mathcal{L}_{\mathcal{M}}$ be a reasonable estimation procedure based on a class $\mathcal{M}$ as in Definition 4.3. Let $\ddot{\theta}(x^n)$ be the estimator that is output by $\mathcal{L}_{\mathcal{M}}$ for given data x^n. The estimates $E_{\ddot{\theta}(x^n)}[\psi(X)]$ of $\overline{\psi(x)}$ are called reliable if and only if the average of ψ can be reliably estimated on the basis of $\mathcal{M}$. The decisions $\delta^*(\text{LOSS}, \ddot{\theta}(x^n))$ are called reliable iff* LOSS *can be reliably used on the basis of $\mathcal{M}$.*

Suppose you issue the estimate $\ddot{\theta}(x^k)$ for data $x_1, \ldots, x_k$. As explained on page 67, the essence of 'reliability' is that *if* future data $x_{k+1}, \ldots, x_n$ is such that $\ddot{\theta}(x^k) = \hat{\theta}(x_{k+1}, \ldots, x_n)$, *then* your estimate $E_{\ddot{\theta}(x^k)}[\psi(X)]$ is equal to $\overline{\psi(x)}$, the average being taken over $x_{k+1}, \ldots, x_n$. This explains the important point that, while on the one hand there is no reason that $\ddot{\theta}(x^k)$ should itself be the maximum likelihood estimator, on the other hand the definition of 'a function ψ that can be reliably estimated' does make use of the maximum likelihood estimator. For *present* data we allow every estimator that has a reasonable possibility of being close to the maximum likelihood estimator over *future* data.

The following two propositions are now straightforward:

Proposition 4.7 *Let $\mathcal{M}_{me}$ be the maximum entropy model class for a function $\phi : \mathbf{E} \rightarrow \mathbf{U}$ with $\phi(x) = (\phi_1(x), \ldots, \phi_m(x))$. Let ψ be a function that can be written as a linear combination of the ϕ_i. Then*

1. *Averages of the function ψ can be reliably estimated on the basis of $\mathcal{M}_{me}$.*
2. *Let P_{me} be the maximum entropy distribution for given constraint $\overline{\phi(x)}^n = t$. The estimates $E_{me}[\psi(X)]$ of $\overline{\psi(x)}$ are reliable.*

Proof: Item (1) is immediate from Proposition 4.1. By item (1). and Definition 4.6 the estimates $E_{\hat{\theta}(x^n)}[\psi(X)]$ of $\overline{\psi(x)}$ are reliable. Since the maximum entropy distribution P_{me} for given constraint $\overline{\phi(x)}^n = t$ coincides with the maximum likelihood model $P(\cdot|\hat{\theta}(x^n))$ (Proposition 3.9), the estimates $E_{me}[\psi(X)]$ of $\overline{\psi(x)}$ are also reliable. □

Proposition 4.8 *Let $\mathcal{M}_{me}$ and ϕ be as above. Let $\text{LOSS}_{lg}(x, \theta) := -\log P(x|\theta)$ be the logarithmic loss function. Then:*

1. *$\delta^*(x^n) = \delta^*(\text{LOSS}_{lg}, P(\cdot|\hat{\theta}(x^n))) = \hat{\theta}(x^n)$ ('the optimal weight assignment over $\mathbf{E}$ when expectation is taken over $\hat{\theta}(x^n)$ coincides with $\hat{\theta}(x^n)$').*
2. *LOSS_{lg} can be reliably used for prediction on the basis of $\mathcal{M}_{me}$.*

Proof: We first prove (1). We have to find the probability distribution w over $\mathbf{E}$ minimizing $E_{\hat{\theta}(x^n)}[-\log P(X|w)]$. By the Information Inequality 3.4 (Chapter 3, page 52), this expression is minimized for $w = \hat{\theta}(x^n)$. Hence $\delta^*(x^n) = w = \hat{\theta}(x^n)$. This proves (1). Now for (2). By (1). and Proposition 4.1 we have $\text{LOSS}_{lg}(y; \delta^*(x^n)) = \text{LOSS}_{lg}(y; \hat{\theta}(x^n)) = -\log P(y|\hat{\theta}(x^n)) = \sum_{i=1}^m \alpha_i \phi_i(y) + \alpha_0$ for some $(\alpha_0, \ldots, \alpha_m)$. This is a linear combination of the ϕ_i. Hence, by Proposition 4.7, averages of $\text{LOSS}_{lg}(y; \delta^*(x^n))$ can be reliably estimated. Then by definitions 4.4 and 4.6 LOSS_{lg} can be reliably used on the basis of $\mathcal{M}_{me}$. □

Summary We summarize the meaning of 'reliable': *if* $\ddot{\theta}$ is a *good* estimate in the sense that it will turn out to be close to $\hat{\theta}_{\text{fut}}$, then decisions which are 'reliable' according to Definition 4.4 are guaranteed to give a loss over future data that is close to the $\ddot{\theta}$- expected loss over future data; for decisions which are 'unreliable' we have no such guarantee.

4.2.3 What we can and cannot do

It seems that, if we restrict ourselves to making only provably reliable decisions, then we can hardly conclude *anything* about what decisions to take. However, there are at least three situations in which we can still do something useful:

Proportional Betting In the case of finite $\mathbf{E}$, we can place bets on outcome e_i proportionally to $P(e_i|\ddot{\theta})$. In Chapter 2, Section 2.7 we showed this to be equivalent to predicting with the logarithmic loss function. Hence by Proposition 4.8 proportional betting is reliable: the logarithm of our accumulated capital after n bets will be close to our expectation of the logarithm of our accumulated capital after n bets.

Indicator Functions Consider the case of finite $\mathbf{E} = \{1, \ldots, k\}$ where the function ϕ is the vector of all indicator functions

$$\phi(x) = (\mathcal{I}(x = 1), \ldots, \mathcal{I}(x = k)).$$

In this situation there are no strong restrictions, since we can reliably predict future values of all functions ψ with domain $\mathbf{E}$. The reason is simply that *all* functions of discrete data can be written as linear combinations of indicator functions. In the Brandeis example, $\phi(x) = x = \sum_{i=1}^6 i\mathcal{I}(x = i)$ which is clearly a linear combination of indicator functions. From this example we immediately see how to write an arbitrary function on a finite domain as a linear combination of indicator functions.

Moment Constraints The first two moments ($E[X]$ and $E[X^2]$) determine some interesting aspects of a probability distribution. If $\mathbf{E}$ is continuous and $\phi = (x, x^2)$, then we can reliably estimate $\phi_1(x) = x$ and $\phi_2(x) = x^2$. A good estimate $\ddot{\theta}$ is then guaranteed to have $E_{\ddot{\theta}}[X]$ and $E_{\ddot{\theta}}[X^2]$ close to $E_{\hat{\theta}_{\text{fut}}}[X]$ and $E_{\hat{\theta}_{\text{fut}}}[X^2]$ respectively. Using *only* these first two moments, we may use for example Chebyshev's inequality to get bounds on the probability that an outcome is more than k standard deviations away from the mean $E_{\ddot{\theta}}[X] \approx E_{\hat{\theta}_{\text{fut}}}[X]$.

There are also many things we *cannot* do if we want to make only reliable estimates and decisions. Roughly stated, *we cannot add and multiply probabilities any more in all the situations where we used to.*

Below we give several examples of what is possible and what is not with reliable estimates and decisions.

Example 4.9 [some probabilities can be added, some cannot] For a six-sided die, suppose we are given that $\gamma_1 = \overline{\mathcal{I}(x=1)}^n$ and $\gamma_2 = \overline{\mathcal{I}(x=2)}^n$. These are the observed frequencies of 1s and 2s in n throws. In this case P_{me} is given by $P_{me}(1) = \gamma_1, P_{me}(2) = \gamma_2, P_{me}(3) = \ldots = P_{me}(6) = (1-\gamma_1-\gamma_2)/4$. P_{me} is also the ML distribution for the sample x^n with respect to the maximum entropy model class $\mathcal{M}_{me}$ of function $\phi(x) = (\phi_1(x), \phi_2(x)) = (\mathcal{I}(x=1), \mathcal{I}(x=2))$. Let us regard P_{me} as an estimate to be used for predicting future data. As $\mathcal{I}(x = 1 \vee x = 2) = \mathcal{I}(x = 1) + \mathcal{I}(x=2)$, we see that $\mathcal{I}(x = 1 \vee x = 2)$ is a linear combination of the functions ϕ_i. By Proposition 4.7 we can reliably use P_{me} to predict that the frequency of the event of throwing a number of spots less than 3 will be $E_{me}[\mathcal{I}(X = 1 \vee X = 2)] = \gamma_1 + \gamma_2$. Since $\mathcal{I}(x = 3)$ cannot be written as a linear combination of ϕ_1 and ϕ_2, we cannot use this proposition to prove that it can be reliably predicted. Indeed, it cannot: the sequence x^n may of course be such that $\gamma_3 \neq P_{me}(3)$.

Example 4.10 [if $P(A)$ can be reliably predicted, then so can $P(\neg A)$] Suppose the indicator function $\mathcal{I}(A)$, where A is some proposition regarding the data, is a linear function of ϕ and hence can be reliably predicted if we use the maximum entropy model class $\mathcal{M}_{me}$ for ϕ. Then $\neg A$, the negation of A, has as indicator function $1 - \mathcal{I}(A)$ and can therefore also be reliably predicted.

Example 4.11 [decision theory] Let $\mathbf{E} = \{0, 1\}$ and let $x^n \in \mathbf{E}^n$. We model x^n by the Bernoulli model class, which (Example 3.8, page 55) is equivalent to the maximum entropy model class for the function $\phi(x) = x$. Let n_0 be the number of 0s and n_1 be the number of 1s in x^n. Let $\hat{\theta}(x^n) = \gamma = n_1/n$, i.e. the frequency of 1s in x^n is γ. Suppose for some reason or other we are in possession of a perfect estimate $\ddot{\theta} = \hat{\theta}(x^n)$ which we will use to predict the x^n. Note that every function ψ over $\mathbf{E}$ can be written as $\psi(x) = ax + b$ which is a linear combination of $\phi(x)$. Every loss function $\text{LOSS}(x;\delta)$ can, for each fixed δ, be thought of as a function over $\mathbf{E}$. It follows, by Proposition 4.7 that we can make reliable predictions against every loss function. We can also directly show this as follows: Let LOSS be an arbitrary loss function such that the optimal decision $\delta^* = \arg\min_{\delta} E_{\ddot{\theta}}[\text{LOSS}(X;\delta)]$ exists. At each time step $1 \leq i \leq n$ we take decision δ^*. Our total loss after n decisions will be

$$\begin{aligned}\sum_{i=1}^{n} \text{LOSS}(x_i;\delta^*) &= n_0 \text{LOSS}(0;\delta^*) + n_1 \text{LOSS}(1;\delta^*) \\ &= n((1-\ddot{\theta})\text{LOSS}(0;\delta^*) + \ddot{\theta}\text{LOSS}(1;\delta^*)) = nE_{\ddot{\theta}}[\text{LOSS}(X;\delta^*)],\end{aligned}$$

where the second equality follows from the fact that $\ddot{\theta}$ was defined as $\ddot{\theta} = \hat{\theta}(x^n) = n_1/n$. Hence the loss we expect to make based on our optimized decision will be equal to the loss we will actually make.

This result generalizes to $\mathbf{E} = \{1, \ldots, k\}$ and the class $\mathcal{M}$ of multinomial distributions over $\mathbf{E}$. Why then don't we always use this class in case there are k possible outcomes per trial? The answer is given once again by the MDL principle: $\mathcal{M}$ contains $k-1$ parameters - if k is large, we would need an awful lot of data to obtain accurate estimates for all of these (note that what we call 'reliable' only means that *if* the estimate of θ is close to $\hat{\theta}_{\text{fut}}$, *then* predictions of some functions of the data will be reliable; if we have too many parameters, then we need a lot of data in order for the premise of this implication to be true. It may then often be more reasonable to use a simpler (in the sense of less parameters) model class; the price we pay is that not all our decisions are reliable any more. Obviously, on the basis of a small data sample we cannot reliably predict a complex phenomenon.

Example 4.12 [non-i.i.d. data] We continue the example above. While all functions of *single* outcomes could be reliably estimated based on $\ddot{\theta} = \hat{\theta}(x^n) = \gamma$, we *cannot* reliably estimate properties of the data like 'the frequency of two ones in a row'. We may be tempted to say that this will be $P(X_{i+1} = 1 | X_i = 1, \ddot{\theta}) \cdot P(X_i = 1 | \ddot{\theta}) = \gamma^2$. But this is *un*-reliable. For example, the data may be generated by a Markov Chain P^* with stationary distribution $P(\cdot|\ddot{\theta})$, while $P^*(X_{i+1} = 1 | X_i = 1)$ is highly different from $P^*(X_{i+1} = 1 | X_i = 0)$. In such a case, for large n, with high probability $\hat{\theta}(x^n) \approx 1/3$ while the frequency of two one's in a row is, with high probability, different from $1/9$. We saw an example of this already in Chapter 1, Section 1.5.

If we look at the indicator function for two ones in a row: $\phi(x_{i+1}, x_i) = \mathcal{I}(x_{i+1} = x_i = 1)$, we see that it has domain $\mathbf{E}^2$ rather than $\mathbf{E}$. Therefore Proposition 4.7 does not apply and, as expected, we cannot prove that the function can be reliably estimated. One could extend the whole formalism by allowing constraints and functions pertaining to more than one outcome, but we will not pursue that idea further.

The previous examples notwithstanding, it seems clear that the requirement of making only reliable decision and predictions will often restrict us too much. In the next chapter we shall see how to partially get around this problem.

4.3 Safe and Risky Statistics

In this section we explore what happens if, based on an estimate $\ddot{\theta}(x^k)$, one restricts oneself to making only reliable predictions and decisions for future data $x_{k+1}, \ldots, x_n$. This leads to a distinction between 'safe' and 'risky' statistics. In safe statistics, only provably reliable predictions and decisions are allowed on the basis of $\ddot{\theta}(x^n)$). In risky statistics the estimate $\ddot{\theta}(x^n)$ is used as if data were actually generated according to it.

We start by treating the special case of maximum entropy. Restricting this principle so that only reliable inferences can be drawn makes it very weak but it nicely illustrates the general idea. Afterwards, we discuss safe statistics in general.

4.3.1 Safe and Risky Maximum Entropy

Let a constraint $\overline{\phi(x)}^n = t$ be given and let P_{me} be the maximum entropy distribution for this constraint. In almost all applications of maximum entropy, one deals with

Let $\overline{\phi(x)}^n = t$ be a given constraint. Let P_{me} be the maximum entropy distribution for this constraint and let $\mathcal{M}_{me}$ be the maximum entropy model class for function ϕ.
We distinguish between two forms of applying the Maximum Entropy formalism:

Safe MaxEnt We use the MaxEnt distribution P_{me} *only* to make provably reliable decisions and predictions, where 'reliable' is defined with respect to the class $\mathcal{M}_{me}$ (Definition 4.4; Proposition 4.7).

Risky MaxEnt We use the MaxEnt distribution as if data were actually distributed according to it.

Figure 4.1: Safe vs. Risky Maximum Entropy

empirical constraints of this form rather than constraints regarding expected values of unknown probability distributions. We can then view the adoption of P_{me} as a model for data x^n as being equivalent to maximum likelihood estimation of x^n using the maximum entropy model class $\mathcal{M}_{me}$ for ϕ as the model class. Since n will usually be large, P_{me} will also be hardly distinguishable from other reasonable (Definition 4.3) estimators based on $\mathcal{M}_{me}$.

While the examples of the previous section show that 'reliable' decisions do give us the possibility to predict *some* aspects of future data using this P_{me}, we may still want more; sometimes, we can also do more. For example, we may have additional reasons for assuming that all sequences of data satisfying the constraints are about equally likely to occur; in that case, by the concentration phenomenon (Chapter 3, sections 3.5 and 3.6), the frequencies $(\gamma_1, \ldots, \gamma_k)$ will with very high probability almost coincide with the maximum entropy probabilities $(P_{me}(1), \ldots, P_{me}(k))$. Since the frequencies are the averages of the indicator functions $(\mathcal{I}(x=1), \ldots, \mathcal{I}(x=k))$, in this 'typical' case the average over x^n of every function ψ over $\mathbf{E}$ will be close to its expected value under P_{me} and P_{me} will serve as a good basis for all predictions and/or decisions concerning the data. In cases where we know that the data are subject to additional constraints, but we just do not know which, no data sequences satisfying the known constraint but not the unknown constraints can arise. In that case, the concentration phenomenon is not applicable and we cannot guarantee that $\overline{\psi(x)}^n = E_{me}[\psi(X)]$ except for functions ψ that are linear combinations of the ϕ. If we are interested in estimating the average of a function $\psi(x)$ that cannot be written as a linear combination of the $\phi(x)$, it may still be a reasonable *guess* to predict that $\overline{\psi(x)}^n = E_{me}[\psi(X)]$ – but this prediction has a fundamentally different status from a 'reliable' prediction.

This consideration leads us to identify two forms of maximum entropy: a *safe* and a *risky* form. They are defined in Figure 4.1. Several people, including Jaynes, have hinted at this difference [77, 24] but, as far as we know, nobody has ever formalized it before.

Ex Nihilo Nihil (almost) Restricting oneself to 'safe' maximum entropy takes the sting out of the principle, as we now show. The price to pay is that with 'safe' maximum entropy we can hardly make any non-trivial inferences any more.

Assume that the constraint $\overline{\phi(x)}^k = t$ is given and that we use the maximum entropy distribution P_{me} for this constraint only to make 'safe' estimates regarding future data $x_{k+1}, \ldots, x_n$.

By Definition 4.4, we are then only allowed to estimate the average value of functions ψ that provably satisfy the following property:

$$\textit{if } \frac{1}{k}\sum_{i=k}^{n} \phi(x_i) \approx \frac{1}{n-k}\sum_{i=k+1}^{n} \phi(x_i) \text{ then } E_{me}[\psi(X)] \approx \frac{1}{n-k}\sum_{i=k+1}^{n} \psi(x_i) \tag{4.7}$$

In words, *if* future data (almost) behaves in accordance with the given constraint $\overline{\phi(x)}^k$ over present data, *then* the average of $\psi(x)$ must be (almost) equal to our estimate of it.

The only thing that is needed for the estimates of averages of functions ψ satisfying (4.7) to be accurate is that the average of $\phi(x)$ over future data will be approximately equal to t, its average over present data. This is still an assumption, or perhaps more aptly 'inductive principle' of course. However, for large k and large $n-k$ it is an assumption that is fundamental to make predictions based on i.i.d. model classes work at all: 'the future should behave similarly to the past'. This inductive principle will be acceptable to many statisticians who have difficulties accepting the maximum entropy principle in general. That is the good news. The bad news is that we can hardly make any non-trivial inferences at all: the only functions ψ that can be provably reliably predicted are linear combinations of the ϕ. The only non-obvious linear combination of the ϕ is the codelength function $-\log P(\cdot)$, which allows us to reliably use P_{me} in proportional betting (Section 4.2.3).

4.3.2 Safe and Risky Statistics

The notions of 'safe' and 'risky' can be extended to parametric statistical estimation in general - whether two-part code MDL, Bayesian MAP estimation or any other sufficiently reasonable procedure is used. Let $\mathcal{L}_{\mathcal{M}}$ be some reasonable statistical inference procedure (Definition 4.3), that on input x^n outputs $\ddot{\theta}(x^n)$. Following Figure 4.1, we may call estimates, decisions and predictions based on $\ddot{\theta}(x^n)$ that are provably reliable 'safe statistics'; we may call them 'risky' otherwise. The crucial difference between safe and risky statistics is that the assumptions that have to be made about 'how the world works' are much weaker in the case of safe statistics than in the case of risky statistics.

To get a clearer idea of 'safe statistics', let us look at a function ψ over $\mathbf{E}$ that cannot be reliably estimated. In that case, the following holds: there exist x^n and y^n in $\mathbf{E}^n$ such that

$$\hat{\theta}(x^n) = \hat{\theta}(y^n) \text{ but } \frac{1}{n}\sum_{i=1}^{n} \psi(x_i) \neq \frac{1}{n}\sum_{i=1}^{n} \psi(y_i)$$

Hence, while x^n and y^n share the same optimal model in the class, using the function ψ one is able to 'detect' that x^n and y^n are really different. Therefore, the value $\hat{\theta} = \hat{\theta}(x^n) = \hat{\theta}(y^n)$ simply does not give sufficient information to determine the average of ψ (it is, in other words, not a sufficient statistic for ψ). This means that we can informally rephrase 'safe statistics' as follows:

> **Safe Statistics** Use an estimator $\ddot{\theta} = \ddot{\theta}(x^k)$ for given data x^k only to predict those properties of future data $x_{k+1}, \ldots, x_n$ that are shared by *all* $x_{k+1}, \ldots, x_n$ for which $\ddot{\theta} \approx \hat{\theta}(x_{k+1}, \ldots, x_n)$.

We end this section by analyzing 'safe' statistics from two more points of view:

Safe Statistics and Data Compression As argued in Chapter 1, Section 1.1.3, there can be no practically applicable statistical inference procedure which, for every data sequence D, outputs a model that captures *all* regularities in D (in the sense that it maximally compresses D using a universal computer language as a code). This means that it will always be possible that the model for data D as found by a practical instantiation of two-part MDL (or any other reasonable inference procedure) does not capture *all* (computable) regularity in the data. We already discussed this in Chapter 1, Section 1.5 where we modeled non-i.i.d. data using the Bernoulli model class. We saw that this leads to unreliable predictions as soon as one tries to estimate aspects of the data like the frequency of '11'. 'Safe statistics' allows us to use an overly simple model with less danger. The only thing that is still required from the data generating process is that the estimate $\ddot{\theta}(x^n)$ is close to $\hat{\theta}_{\text{fut}}$, the maximum likelihood model for future data. We emphasize at this point that this is still a requirement that may not hold in practice. Nevertheless, safe statistics can be seen as a safeguard against the conclusion that an estimate captures *all* the regularity in the data: it ensures that we predict only those aspects of the data on the basis of estimate $\ddot{\theta}$ that are shared by *all* x^n for which $\ddot{\theta} = \hat{\theta}(x^n)$, even 'special' x^n that can be compressed further on the basis of another, more complex model.

Classical Statistics is Risky Let ϕ be a given function over sample space $\mathbf{E}$. It is useful to consider the case in which data are generated by repeated sampling of some distribution P^*. We will assume that $E_{P^*}[\phi(X)] = t$ for some t in the range of ϕ. Let $\mathcal{M}$ be a model class that is finitely parameterized by some Γ. Under fairly general conditions (as we shall prove in Chapter 5, Section 5.3), there exists a model $\tilde{\theta} \in \Gamma$ such that

$$\tilde{\theta} = \arg\min_{\theta \in \Gamma} E_{P^*}[-\log P(X|\theta)] \tag{4.8}$$

Hence $\tilde{\theta}$ maximizes the expected log-likelihood. By the law of large numbers, as $n \to \infty$, we obtain $\overline{\phi(x)}^n \to t$ and, if $\mathcal{M}$ is regular enough, also $\hat{\theta}(x^n) \to \tilde{\theta}$ with probability 1 (the latter to be shown in Chapter 5, Lemma 5.14). But it may very well be possible that $E_{\tilde{\theta}}[\phi(x)] \neq E_{P^*}[\phi(x)]$ and hence $E_{\hat{\theta}(x^n)}[\phi(X)] \not\to \overline{\phi(x)}^n$. In such a case, with high probability we end up with a model that gives a wrong idea of the average of ϕ. There are two ways to avoid this: the first is to adhere to 'safe statistics' and to use as functions ϕ only those functions whose averages we can reliably predict on the basis of our model class $\mathcal{M}$; it is clear that the problem can then not occur. The second way is a kind of cheating: simply declare that the discrepancy mentioned will never happen by making the additional *assumption* that $P^* = P(\cdot|\theta^*)$ for some $\theta^* \in \Gamma$ ('the true model is in the class'). Then by the Information Inequality (Equation 3.4, page 52), $\tilde{\theta} = \theta^*$. In this case, we can apply the law of large numbers to obtain that for all functions ϕ over $\mathbf{E}$, $E_{\hat{\theta}(x^n)}[\phi(X)] \to \overline{\phi(x)}^n$ with probability 1.

The assumption that data are actually generated by a model $P^* = P(\cdot|\theta^*)$ in the class $\mathcal{M}$ under consideration is fundamental to much of classical statistics (consider, for example, classical hypothesis testing [6]). Unfortunately, it can hardly ever be justified [128]. In our view, it is a form of 'risky' statistics. The alternative assumption that data are generated by i.i.d. sampling from *some* distribution - not necessarily contained in one's model class - is, of course, still an assumption that is difficult to justify, but it is nevertheless much weaker. Whereas in classical statistics, one freely draws conclusions that strictly speaking only follow (with high probability) if one's model class contains the true distribution, in safe statistics one only draws conclusions that already follow (with high probability) under the much weaker assumption that the data is generated by i.i.d. sampling from *some* distribution such that the minimizing $\tilde{\theta}$ in (4.8) exists.

What to use: safe, risky, or in between? We are not saying that one should only use 'safe' statistics. There is nothing wrong in using risky statistics, as long as one is aware that one is then implicitly making some additional assumptions. It can also be the case that one has additional knowledge about the 'data generating machinery' that make some of the 'risky' predictions less risky. For example, if the data consists of repeated trials of some physical experiment (like dice throwing), it is reasonable to assume that the outcomes are truely (almost) independent. Let $\mathcal{M}_{me}$ be once more the maximum entropy class for $\phi(x) = x$ and let P_{me} be the maximum entropy model for $\overline{x}^n = t$. Our knowledge of physics allows us to predict that the event $\{X_i = a, X_{i+1} = b\}$ will occur with probability $P_{me}(X_i = a, X_{i+1} = b) = P_{me}(X_i = a)P_{me}(X_{i+1} = b)$ even though this is, strictly speaking, not a safe inference (see Example 4.12).

4.4 Conclusion and Outlook

We have introduced a distinction between 'reliable' and 'unreliable' inferences, leading to 'safe' and 'risky' statistics. In the next chapter we will put the distinction to use.

Chapter 5

How to Make Predictions Reliable

According to the MDL Principle, models of data are always probabilistic: if a class of non-probabilistic models is used to model the data at hand, it is first mapped to a corresponding probabilistic class (Chapter 2, Section 2.2). On the other hand, as was repeatedly stressed in chapters 1 and 2, these models are to be interpreted as *codes* for the data - not as traditional probability distributions according to which data are generated. This raises the question of what conclusions (predictions) about future data can and what conclusions cannot be drawn on the basis of such 'probabilistic' models. The question becomes all the more difficult if we acknowledge, in line with the MDL philosophy, that our models will always be partially wrong – even if they allow us to substantially compress the data.

Main Concepts and Results of this Chapter In this chapter, we identify conditions under which a probabilistic model, inferred from the data, can be used to reliably predict future data *even if that model is really a probabilistic representation of a non-probabilistic model and/or if the model is wrong.* We show that given a model class $\mathcal{M}$ with a fixed number of parameters ($\mathcal{M}$ is not necessarily probabilistic) and an error function ER, we can turn $\mathcal{M}$ into a probabilistic version $\langle\mathcal{M}\rangle_{\text{ER}}$ that is essentially equivalent to $\mathcal{M}$ except that it leads to 'reliable' estimates of the error function (where 'reliable' is defined as in Definition 4.4 of the previous chapter). We call $\langle\mathcal{M}\rangle_{\text{ER}}$ the *entropification* of $\mathcal{M}$. Entropification stands at the basis of the main results of this chapter (theorems 5.16-5.18) which can be summarized as follows:

1. Under the assumption that the data are i.i.d. according to an essentially arbitrary P^*, we can infer from a large enough data set D, with high probability, a model $\tilde{\theta}$ in $\langle\mathcal{M}\rangle_{\text{ER}}$ that is
   (a) the optimal model in $\langle\mathcal{M}\rangle_{\text{ER}}$ for predicting future data against error function ER: among the models in $\langle\mathcal{M}\rangle_{\text{ER}}$, $\tilde{\theta}$ minimizes the 'true' expected error $E_{P^*}[\text{ER}(Y|\tilde{\theta}, X)]$.

(b) can be 'reliably' used, since it gives a truthful impression of its own performance in the sense that $E_{\tilde{\theta}}[\text{ER}(Y|\tilde{\theta}, X)] = E_{P^*}[\text{ER}(Y|\tilde{\theta}, X)]$).

Essentially, this means that whenever the assumption that data are i.i.d. can be justified and the function ER according to which errors will be measured is known, the model $\tilde{\theta}$ can be used (1) to arrive at optimal predictions (relative to the model class $\langle \mathcal{M} \rangle_{\text{ER}}$) of future data against ER and (2) as an accurate estimator of how good these predictions will be - even if $\mathcal{M}$ is a wrong ('misspecified') model class that does not contain any model that is similar to the 'true' P^*.

While $\tilde{\theta}$ can be inferred from data by many statistical inference procedures (not necessarily MDL), the 'entropification' of $\mathcal{M}$ turns out to yield additional results when combined with MDL, leading to the other two important results of this chapter:

2. Entropification removes an inherent arbitrariness in MDL's trade-off between error and model complexity that occurs if non-probabilistic model classes are used (Section 5.4)

3. Entropification allows us to associate codes with non-probabilistic model classes in an optimal manner, in the sense that the 'worst-case expected codelength' is minimized (Proposition 5.22).

Preliminary Status The material presented in this chapter has a very preliminary character; it could certainly be rephrased in a simpler, clearer and more compact fashion. Nevertheless, we decided to include it in this thesis, since it does serve to justify our view of probability distributions as codes, which, on one hand, cannot be used to predict just any aspect of the data, but which, on the other hand, can be used to reasonably predict some aspects of the data even if they are very different from the 'true' distribution generating the data.

Organization of the Chapter We will start (Section 5.1) with a motivating example. In Section 5.2 we formally introduce the concept of 'entropification', present some of its basic properties and give some examples of its use. Section 5.3 presents our main results (item 1 in the listing above). In Section 5.4 we show how entropification can be used in the context of MDL.

Attached to the chapter is an Epilogue to Part I of this thesis in which we argue that 'entropification' resolves the problematic issues concerning MDL that were mentioned at various places in Chapters 1–3.

5.1 An Introductory Example

In the next section we will formally introduce the concept of 'entropification'. In this section we present an introductory example.

Example 5.1 Consider once more the fitting of polynomials. Let data $D = (x^n, y^n)$ be given. In a non-probabilistic approach to this problem, we would use some algorithm that, for each D, when input D, outputs a polynomial $\ddot{H}$ that it regards as an optimal

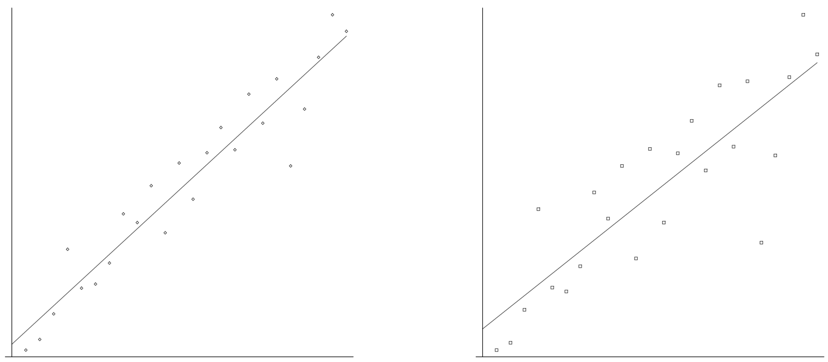


Figure 5.1: Two sets of points with the same least-squares first-degree polynomial. The distances between $f(x_i)$ and y_i in the picture on the right are exactly twice as large as those in the picture on the left.

hypothesis for D. Such a polynomial $\ddot{H}$ in itself does not give any information on how good it will be on future data, and this can be problematic. For example, imagine that a company uses some sophisticated tool to infer $\ddot{H}$ from lots of data, and then sells $\ddot{H}$ to a client so that the client can use it to predict future data ($\ddot{H}$ may, for example, be a model for some data from the stock exchange and the client may use it as a guideline for future investments). If the company only gives $\ddot{H}$ to the client, then the client has no means of knowing how well $\ddot{H}$ actually *will* predict future data. This can be most easily demonstrated if we imagine that the model class $\mathcal{M}$ is restricted to the class of first-degree polynomials. In Figure 5.1 we show two sets of points that have the same optimal (in the sense of minimizing squared error) first degree polynomial $\widehat{H}$. Let us denote by D_1 the set of points depicted on the left of Figure 5.1 and by D_2 the set of points depicted on the right. Assuming that the company uses a reasonable (see Chapter 4, Definition 4.3) method to infer the best polynomial, it will infer a polynomial reasonably close to $\widehat{H}$ for both data sets depicted. However, if future data behaves like present data, then in the case of D_1, $\widehat{H}$ will be a much better predictor than in the case of D_2. The client (who has not seen the 'training' data) would probably like to know how good the hypothesis $\ddot{H} \approx \widehat{H}$ is before he decides whether to buy it or not; but $\ddot{H}$ does not reveal this information. Therefore, the client may rather want the company to sell a tuple $(\ddot{H}, \ddot{\sigma}^2)$ where $\ddot{\sigma}^2$ is some reasonable estimate of the error $\ddot{H}$ will make on future data. In this way, he will get a *reliable* impression of the performance of $\ddot{H}$.

When using two-part code MDL, we do not estimate a polynomial for data D directly. Rather, we turn each polynomial $H \in \mathcal{M}$ into a probability distribution $P(\cdot|H)$ and we look for the polynomial H_{mdl} minimizing $-\log P(D|H) + L(H)$. Specifically, in

Chapter 1, Section 1.4 and Chapter 2, Section 2.2 we showed how to change $\mathcal{M}$ to a corresponding class of (conditional) probability distributions $\mathcal{M}_{pr}$ such that each polynomial H is mapped to a corresponding $P(\cdot|H,\cdot)$ with for all $D = (x^n, y^n)$:

$$-\log P(y^n|H, x^n) = \mathrm{ER}_{sq}(y^n|H, x^n) + K \tag{5.1}$$

We showed that the $P(\cdot|H,\cdot)$ achieving this for us is a product distribution of Gaussians with density given by:

$$\begin{aligned} P(y^n|H, x^n) &= \prod_{i=1}^{n} P(y_i|H, x_i) \text{ where} \\ P(y|H, x) &= \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y - H(x))^2}{2\sigma^2}). \end{aligned} \tag{5.2}$$

with the variance σ^2 given by $\sigma^2 = (2 \ln 2)^{-1}$. The advantage of turning each H into a probability distribution is that we can express the error H makes on D in terms of bits and can therefore add to it a complexity term which is also expressed in bits.

Now, as we saw above, the optimal *polynomial* H_{mdl} for data D tells us nothing on how good it will be in predicting future data. However, the probabilistic version $P(\cdot|H_{\mathrm{mdl}},\cdot)$ does make such a statement, for it defines an expected error over future data $\mathrm{ER}_1 = E_{P(\cdot|H_{\mathrm{mdl}})}[\mathrm{ER}_{sq}(Y|H_{\mathrm{mdl}}, X)]$. We may be tempted to use ER_1 as an estimate of the error we will make when predicting future data. But this cannot work, since we have for all $H \in \mathcal{M}$:

$$E_{P(\cdot|H,\cdot)}[\mathrm{ER}_{sq}(Y|H, X)] = \sigma^2 \tag{5.3}$$

(this is a standard fact about regression; see [6]). Returning to Figure 5.1, if we are asked to use $P(\cdot|H_{\mathrm{mdl}},\cdot)$ to make an estimate of the average squared error we will make on future data, we would answer $\sigma^2 = (2 \ln 2)^{-1}$ both in the case where H_{mdl} is based on D_1 and in the case where H_{mdl} is based on D_2. Clearly, the answer will necessarily be false for at least one of the two sets D_1 or D_2 (and in general, for almost all D for which H_{mdl} is the optimal two-part code model).

We could solve this problem by *never* using the optimal polynomial H_{mdl} in its probabilistic form $P(\cdot|H_{\mathrm{mdl}},\cdot)$ and instead supply it, just as the non-MDL estimate $\ddot{H}$ we referred to above, by a separately obtained estimate $\ddot{\sigma}^2$ of the error that H_{mdl} will make on future data. But this is not in line with the general MDL Principle (Chapter 2, Section 2.1), which says that we should hunt for the model that gives optimal compression of our data, and hence best captures the regular features of the data. The reason is that the estimate $\ddot{\sigma}^2$ can be used to further compress the data: it gives additional information about the data over and above the information contained in H_{mdl}. Therefore, we should rather try to find a single model for the data which combines the regularities expressed by H_{mdl} and those expressed by $\ddot{\sigma}^2$. This is the first central idea of this chapter. It can be achieved as follows: instead of mapping each $H \in \mathcal{M}$ to a *single* probability distribution $P(\cdot|H,\cdot)$ with a fixed value of σ^2, we will map it to a whole class of distributions $P(\cdot|H,\sigma^2,\cdot)$ (one for each $\sigma^2 > 0$) where $P(\cdot|H,\sigma^2,\cdot)$ is the conditional normal distribution given by (5.2) with variance

σ^2. The resulting class of models contains, for each $H \in \mathcal{M}$, the probabilistic models $P(\cdot|H, \sigma^2, \cdot)$ for all $\sigma^2 > 0$. By (5.3), $E_{P(\cdot|H,\sigma^2,\cdot)}[\text{ER}_{sq}(Y|H,X)] = \sigma^2$. Hence, the models $P(\cdot|H, \sigma^2, \cdot)$ restricted to fixed H all correspond to a different expected squared error of H. We can now use two-part code MDL (but, if we like, also another method) to find the optimal probabilistic model $(H_{\text{mdl}}, \sigma^2_{\text{mdl}})$ for the data. This probabilistic model *does* give a reliable idea of its own performance over future data. Formally, let $H \in \mathcal{M}$ be arbitrary and let $\hat{\sigma}^2_H$ stand for the maximum likelihood estimator of D for this H. That is, among all models (H, σ^2) with this fixed H it maximizes the likelihood of D. The following is a standard result about regression (also easy to verify): for all n, all $D = (x^n, y^n)$ and for every $x \in \mathbf{E}_x$,

$$E_{(H,\hat{\sigma}^2_H)}[\text{ER}_{sq}(Y|H,X)|X = x] = \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} \text{ER}_{sq}(y_i|H, x_i)$$

In words, for every $H \in \mathcal{M}$, according to the maximum likelihood estimate $\hat{\sigma}^2_H$, the expected squared error of predicting Y given that $X = x$ is independent of x and equal to the average squared error over the training data. This implies that for each fixed H, the average error of H can be *reliably* estimated. Here 'reliable' is defined as in Definition 4.4 of Chapter 4. The advantage of reliable estimation has been amply discussed in the previous chapter (see page 73). Roughly, it implies that *if* future data is similar to the training data, *then* the expected value of the error according to an estimate based on the training data will be close to the actual average error obtained when this estimate is used to predict future data. In our example, it implies that, if $(\ddot{H}, \ddot{\sigma}^2)$ is estimated from data D_1 by a reasonable (Chapter 4, Definition 4.3) method, then $\ddot{\sigma}^2$ will be approximately equal to the average error of $\ddot{H}$ over D_1; if it is estimated from D_2, then $\ddot{\sigma}^2$ will be approximately equal to the average error of $\ddot{H}$ over D_2.

One can use the reliability of error estimates to prove (as we will in Section 5.3) that if the data are *independently* generated according to an essentially arbitrary distribution P^*, then every reasonable inference procedure will, for large enough D, output a model $P(\cdot|\ddot{H}, \ddot{\sigma}^2, \cdot)$ such that the following holds: let, among all $H \in \mathcal{M}$, $\widetilde{H}$ be the model that minimizes the 'true' expected error $E_{P^*}[\text{ER}_{sq}(Y|H,X)]$ and let $\tilde{\sigma}^2 = E_{P^*}[\text{ER}_{sq}(Y|\widetilde{H},X)]$ be the 'true' expected error of this optimal $\widetilde{H}$. Then, with probability 1, (1) $\ddot{H}$ is arbitrarily close to the optimal model $\widetilde{H}$ and (2) $\ddot{\sigma}^2$ itself is arbitrarily close to $\tilde{\sigma}^2$.

What is new?

It is well-known that modeling errors using the normal distribution with varying σ^2 works even when the errors are not truely normally distributed [20], so this far, there is nothing really new here. Our own contribution lies in the fact that we consider the general case of (almost) *arbitrary* error functions ER and model classes $\mathcal{M}$. In the next section, we will give a recipe of how, given $\mathcal{M}$ and ER, one can define a new probabilistic model class $\langle\mathcal{M}\rangle_{\text{ER}}$ that has some special properties. We call the model class $\langle\mathcal{M}\rangle_{\text{ER}}$ the *entropification* of $\mathcal{M}$ with respect to ER. $\langle\mathcal{M}\rangle_{\text{ER}}$ is constructed from $\mathcal{M}$ by adding a single extra real-valued parameter β as part of the hypotheses: models in $\langle\mathcal{M}\rangle_{\text{ER}}$ are indexed by parameters $\theta = (H, \beta)$ for some $H \in \mathcal{M}$ and $\beta \in \mathbf{R}$. If

(H, β) is inferred from data D, then the β associated with H can be interpreted as a reliable estimate of the error H will make on future data. β will also determine the entropy of the model (H, β), hence the name 'entropification'. We give three examples corresponding to three often used error functions.

If $\mathcal{M}$ is a class of continuous functions and ER is the squared error, then $\langle \mathcal{M} \rangle_{\text{ER}}$ turns out to be equivalent to the class $\{P(\cdot|H, \sigma^2, \cdot) \mid H \in \mathcal{M}; \sigma^2 > 0\}$ where $P(\cdot|H, \sigma^2, \cdot)$ is as given by (5.2). This coincides with the case considered above: if (H, σ^2) is inferred from D, then σ^2 can be interpreted as an estimate of the squared error H will make on future data.

If $\mathcal{M}$ is a class of concepts (functions from $\mathbf{E}_x$ to $\mathbf{E}_y = \{0, 1\}$) and ER is the 0/1-error (see Chapter 2, Example 2.4), then, as we will see in Example 5.3, $\langle \mathcal{M} \rangle_{\text{ER}}$ is equivalent to a class of distributions $\{P(\cdot|H, \theta, \cdot) \mid H \in \mathcal{M}; 0 < \theta < 1\}$ where

$$E_{P(\cdot|H,\theta,\cdot)}[\text{ER}(Y|H, X)] = \theta$$

θ can be interpreted as the probability that $H(X) \neq Y$. If (H, θ) is inferred from D, then θ can be interpreted as an estimate of the 0/1-error that H will make on future data.

If $\mathcal{M}$ is a class of probabilistic models $\{P(\cdot|\eta) \mid \eta \in \Gamma\}$, and ER is the logarithmic error (Chapter 2, Section 2.7), then (roughly) $\langle \mathcal{M} \rangle_{\text{ER}}$ will turn out to be equivalent to a class $\{P(\cdot|\eta, \beta) \mid \eta \in \Gamma; \beta \in \mathbf{R}\}$; an explicit formula for the distributions $P(\cdot|\eta, \beta)$ will be given later (Example 5.4). If (η, β) is inferred from D, then β can be interpreted as an estimate of the logarithmic error that η will make on future data.

Other Error Functions The possibility of making estimates of errors ER reliable by using the model class $\langle \mathcal{M} \rangle_{\text{ER}}$ raises the following interesting question: what happens if we use a model (H, β) in the class $\langle \mathcal{M} \rangle_{\text{ER}}$ to predict against other error functions $\text{ER}' \neq \text{ER}$? The models (H, β) we infer from data D are still, in the first place, to be interpreted as codes rather than classical probability distributions. Therefore, when $\text{ER}' \neq \text{ER}$, it is not a priori clear whether it makes any sense to use $E_{P(\cdot|H,\beta,\cdot)}[\text{ER}'(Y|H, X)]$ as an estimate of the average error H will make over future data. It turns out that such an estimate (of ER') can still be seen as a 'reasonable guess', but it has a status fundamentally different (weaker) from the estimate of ER. This will be seen in the Epilogue to this chapter (page 119).

The Goals of Entropification: reliable estimates of errors + applications in MDL Summarizing, 'entropification' will serve as a generic means to make estimates of errors reliable. This fact will be used in Section 5.3 to prove our main results, which we already summarized at the beginning of this chapter (items 1(a) and 1(b), page 82). It turns out that, in connection with MDL, entropification serves several other purposes besides making estimates of errors reliable; these were also already summarized on page 82. They will be treated in Section 5.4.

Now that we have given a first idea of 'entropification' and have mentioned its main goals, we proceed to define the concept formally.

5.2 Entropification of a Model Class

In this section we formally introduce the concept of 'entropification' and give some examples of its use. This will be done in (sub-) sections 5.2.1 and 5.2.2. Section 5.2.3 presents some of the basic properties of entropification and Section 5.2.4 summarizes the essentials of these properties.

5.2.1 Preliminaries

We will assume throughout the remainder of this chapter that all error functions ER considered are sufficiently regular. More precisely, let $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$, and let $\text{ER} : \mathbf{E} \times \mathcal{M} \rightarrow \mathbf{R}$ be an error function defined as in Chapter 2 on page 31. Our assumption is that, for all fixed $H \in \mathcal{M}$, $\text{ER}(y|H, x)$ considered as a function of x and y can be used to define a maximum entropy model class. Specifically, we assume that for all $H \in \mathcal{M}$, the function ϕ_H over $\mathbf{E}_x \times \mathbf{E}_y$ defined by $\phi_H(x, y) := \text{ER}(y|H, x)$ satisfies conditions **C1**-**C3** of page 54. This ensures that the maximum entropy model class for $\phi_H(x, y)$ exists. But we need something stronger than merely the guaranteed existence of this class, as we will now explain.

Conditional and Unconditional Case Throughout this chapter, we consider two cases. In the first case, the hypothesis class $\mathcal{M}$ contains models relating to $\mathbf{E}_y$ and not $\mathbf{E}_x$ (for example, each $H \in \mathcal{M}$ is itself a probabilistic model over $\mathbf{E}_y$ or each H is a relation over $\mathbf{E}_y$ not involving $\mathbf{E}_x$). In such a case $\mathbf{E}_x$ does not really play any role and we could have equally well set $\mathbf{E} = \mathbf{E}_y$. In this situation, the entropification of a model class $\mathcal{M}$ with respect to error function $\text{ER} : \mathbf{E} \times \mathcal{M} \rightarrow \mathbf{R}$ is the class of probabilistic models containing, for each $H \in \mathcal{M}$, the class of distributions $P(\cdot|H, \beta)$ defined by

$$P(y|H, \beta) = \frac{1}{Z_H(\beta)} \exp(-\beta \text{ER}(y|H)) \tag{5.4}$$

where $Z_H(\beta)$ is a normalizing factor:

$$Z_H(\beta) := \sum_{y \in \mathbf{E}_y} \exp(-\beta \text{ER}(y|H)) \tag{5.5}$$

and β ranges over all $\beta \in \mathbf{R}$ for which $P(y|H, \beta)$ is well-defined. As can be immediately seen from equations 3.7 and 3.8 (Chapter 3, page 53), the distributions (5.4) are formally equivalent to maximum entropy distributions.

The formal definition of entropification we give below unifies the unconditional case with the more complicated *conditional* or *supervised* case. In this case, we assume that the outcomes $x \in \mathbf{E}_x$ do play a role, and we are interested in the conditional version of (5.4), $P(y|H, \beta, x) = Z_{H,x}^{-1}(\beta) \exp(-\beta \text{ER}(y|H, x))$ with $Z_{H,x}(\beta) := \sum_{y \in \mathbf{E}_y} \exp(-\beta \text{ER}(y|H, x))$. However, all our results will only hold if the resulting distributions are still 'essentially' maximum entropy distributions. For this reason, we must additionally assume that the following conditions hold:

C4 $\mathbf{E}_x$ is either finite or compact.

C5 ER is such that for each fixed H and each fixed $\beta \in \mathbf{R}$, $Z_{H,x}(\beta)$ is either equal for all $x \in \mathbf{E}_x$ or it diverges for all $x \in \mathbf{E}_x$.

C5 turns out to hold for most error functions ER of interest. These include (as will be shown formally in Proposition 5.11) error functions like the squared and 0/1-error. **C5** allows us to drop the subscript x in $Z_{H,x}(\beta)$ and write, for arbitrary $x \in \mathbf{E}_x$:

$$Z_H(\beta) = \sum_{y \in \mathbf{E}_y} \exp(-\beta \text{ER}(y|H,x)). \tag{5.6}$$

In the remainder of this chapter, we tacitly assume ER to satisfy conditions **C1**-**C3** of page 54 and conditions **C4** and **C5** listed here.

5.2.2 Formal Definition of Entropification

We are now ready to define entropification formally.

Definition 5.2 (Entropification) *Let $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$. The entropification of a model class $\mathcal{M}$ with respect to error function* ER : $\mathbf{E} \times \mathcal{M} \rightarrow \mathbf{R}$ *is the class of (conditional) probabilistic models*

$$\langle \mathcal{M} \rangle_{\text{ER}} = \{P(\cdot|\theta,\cdot) \mid \theta = (H,\beta); H \in \mathcal{M}; \beta \in \Gamma_{nat}(H)\} \tag{5.7}$$

Here $P(\cdot|\theta,\cdot) = P(\cdot|H,\beta,\cdot)$ is a conditional probabilistic model defined as follows:

1. *For each $x \in \mathbf{E}_x$, $P(\cdot|\theta,x) = P(\cdot|H,\beta,x)$ is a probability distribution over $\mathbf{E}_y$ defined by*

$$P(y|H,\beta,x) = \frac{1}{Z_H(\beta)} \exp(-\beta \text{ER}(y|H,x)) \textit{ for all } y \in \mathbf{E}_y. \tag{5.8}$$

   Here $Z_H(\beta)$ is as in (5.6).

2. *For all $(x^n,y^n) \in \mathbf{E}^n$, $P(y^n|H,\beta,x^n) = \prod_{i=1}^n P(y_i|H,\beta,x_i)$.*

For each H, the set of β such that $P(\cdot|H,\beta,\cdot) \in \langle \mathcal{M} \rangle_{\text{ER}}$ is given by

$$\Gamma_{nat}(H) = \{\beta \mid Z_H(\beta) < \infty\}.$$

If $\mathbf{E}$ is continuous, the sum in $Z_H(\beta)$ gets replaced by the corresponding integral. $Z_H(\beta)$ acts as normalizing constant.

Codelengths based on models in $\langle \mathcal{M} \rangle$; first interpretation of β The code corresponding to $P(\cdot|H,\beta,.)$ leads to the following code lengths (expressed in nats):

$$L(y^n|H,\beta,x^n) = -\ln P(y^n|H,\beta,x^n) = \beta \sum_{i=1}^n \text{ER}(y^n|H,x^n) + n \ln Z_H(\beta) \tag{5.9}$$

We see that the code length of y^n given H,β,x^n contains an error term and a 'uniform' term $n \ln Z_H(\beta)$ that grows linearly in n and is equal for all y^n. This shows that β

can be interpreted as determining how strongly the error should be weighed in the code length corresponding to hypothesis (H, β). The extreme case $\beta = 0$ corresponds, for each x_i, to the uniform distribution over all outcomes in $\mathbf{E}_y$. For fixed H, in the limit for $\beta \to \infty$, the probability under (H, β) of an outcome y^n given x^n with $\text{ER}(y^n|H, x^n) > 0$ becomes 0.

Obtaining β; second interpretation of β We can estimate β by any statistical means, for example, by two-part code MDL (Chapter 1). In that case, we would pick as optimal hypothesis for given data D the $\theta_{\text{mdl}} = (H_{\text{mdl}}, \beta_{\text{mdl}})$ that minimizes the sum of (5.9) and $L_C(H_{\text{mdl}}, \beta_{\text{mdl}})$, the description length of the (extended) hypothesis. In general, if the tuple (H, β) is estimated from data D, then β can be interpreted as (a monotonic transform of) an estimate of the error H will make on future data, as will be made clear in the examples below. This gives a second interpretation of β; we will encounter even a third. These interpretations will be summarized in Section 5.2.4.

Example 5.1, continued Let $\mathcal{M}$ be a class of continuous functions $H : \mathbf{E}_x \to \mathbf{E}_y$. From the definition of entropification (Definition 5.2) we can see (by substituting $\beta = (1/2\sigma^2)$) that $\langle\mathcal{M}\rangle_{\text{ER}_{sq}}$, the entropification of $\mathcal{M}$ with respect to the squared error, is equivalent to the model class that supplies $\mathcal{M}$ with a normal error distribution of arbitrary variance $\sigma^2 > 0$. Formally,

$$\langle\mathcal{M}\rangle_{\text{ER}_{sq}} = \{P(\cdot|H, \sigma^2, \cdot) \mid H \in \mathcal{M}; \sigma^2 > 0\} \tag{5.10}$$

with $P(\cdot|H, \sigma^2, \cdot)$ as given by (5.2) (page 84). In this case, the value of $Z_H(\beta)$ is independent of H and finite for all $\beta > 0$. The parameter space will be $\Gamma_{nat}(H) = \{\beta|\beta > 0\}$ independently of H, corresponding to all variances $\sigma^2 = 1/(2\beta) > 0$. Note that, by Equation 5.3, $E_{P(\cdot|H,\sigma^2,\cdot)}[\text{ER}_{sq}(Y|H, X)] = \sigma^2$. Hence when the tuple (H, β) is inferred from data, then β, which determines σ^2, can be seen as an estimate of the expected squared error of H.

Example 5.3 [concept learning and Bernoulli parameters] Let $\mathcal{M}$ be a class of concepts over $\mathbf{E} = \mathbf{E}_x \times \{0, 1\}$ and let ER_{01} be the 0/1-error function (Chapter 2, Example 2.4). Let the observational data $D = (x^n, y^n)$. Let $\langle\mathcal{M}\rangle_{\text{ER}_{01}}$ be the entropification of $\mathcal{M}$ with respect to error function ER_{01}. Let $\langle H\rangle_{\text{ER}_{01}} = \{P(\cdot|H, \beta) \mid \beta \in \Gamma_{nat}(H)\}$ be the restriction of $\langle\mathcal{M}\rangle_{\text{ER}_{01}}$ to models with fixed $H \in \mathcal{M}$. Substituting $\beta := \ln(1 - \theta) - \ln\theta$ in Definition 5.2, we find that $\langle H\rangle_{\text{ER}_{01}}$ is just the class of Bernoulli models containing one model for each possible probability of error (see also Chapter 3, Example 1.1):

$$\begin{aligned} E_\beta[\text{ER}_{01}(Y|H, X)] = P_\beta\{\text{ER}_{01}(Y|H, X) = 1\} = P_\beta\{H(X) \neq Y\} = \\ = \frac{1}{Z(\beta)} e^{-\beta \cdot 1} = \theta \end{aligned} \tag{5.11}$$

and

$$\begin{aligned} E_\beta[|1 - \text{ER}_{01}(Y|H, X)|] = P_\beta\{\text{ER}_{01}(Y|H, X) = 0\} = P_\beta\{H(X) = Y\} = \\ = \frac{1}{Z(\beta)} e^{-\beta \cdot 0} = 1 - \theta \end{aligned} \tag{5.12}$$

It follows that the class $\langle\mathcal{M}\rangle_{\text{ER}_{01}}$ can equivalently be parameterized as $\langle\mathcal{M}\rangle_{\text{ER}_{01}} = \{P(\cdot|H,\theta,\cdot) \mid H \in \mathcal{M}, 0 < \theta < 1\}$, such that, if $\text{ER}_{01}(y^n|H,x^n) = k$, then

$$P(y^n|H,\theta,x^n) = \theta^k(1-\theta)^{n-k} \tag{5.13}$$

This expresses that the probability of error of H is equal to θ for each observation, independently of any other observations. (5.11) shows that, if (H,β) is inferred from data D, then β (which determines θ) can be interpreted as an estimate of the expected 0/1-error of H, which is just the probability that H misclassifies D. Just as in the previous example, β serves to estimate the expected (in this case, 0/1-) error.

MDL is usually applied to concept classes in a way that does not involve entropification [122, 84]. In Section 5.4, Example 5.21 we show that the 'traditional' way of applying MDL to a concept class $\mathcal{M}$ is essentially equivalent to applying MDL to the probabilistic class $\langle\mathcal{M}\rangle_{\text{ER}}$, thus reconciling the two views.

Example 5.4 [Entropification of probabilistic model classes] What happens if we try to entropify a probabilistic model class $\mathcal{M}$? For simplicity, we only consider the case where $\mathcal{M} = \{P(\cdot|\eta) \mid \eta \in \Gamma_{\mathcal{M}}\}$ is a class of i.i.d. probabilistic models over $\mathbf{E}_y$. Similarly, we consider only error functions $\text{ER} : \mathbf{E}_y \times \mathcal{M} \to \mathbf{R}$. The values of x_i are therefore irrelevant and $\mathcal{M}$ consists of full rather than conditional probability distributions. Definition 5.2 is seen to simplify in this case to

$$\begin{aligned}\langle\mathcal{M}\rangle_{\text{ER}} &= \{P(\cdot|(\eta,\beta)); \eta \in \Gamma_{\mathcal{M}}; \beta \in \Gamma_{nat}(\eta)\} \text{ where}\\ P(y|(\eta,\beta)) &= \frac{1}{Z_\eta(\beta)}\exp(-\beta\text{ER}(y|\eta)),\\ P(y^n|(\eta,\beta)) &= \prod_{i=1}^{n} P(y_i|(\eta,\beta))\end{aligned} \tag{5.14}$$

A natural error function for probabilistic models is the logarithmic error $\text{ER}_{lg}(y^n|\eta) = -\sum_{i=1}^n \ln P(y_i|\eta)$; see Chapter 2, Section 2.7. Using the logarithmic error, we obtain:

$$P(y|\eta,\beta) = \frac{1}{Z_\eta(\beta)}\exp(\beta\ln P(y|\eta)) = \frac{P(y|\eta)^\beta}{\sum_{y\in\mathbf{E}_y} P(y|\eta)^\beta} \tag{5.15}$$

We now consider two cases:

$\mathcal{M}$ is an exponential family If $\mathcal{M}$ is a full exponential family (Chapter 3, Section 3.4) then $\langle\mathcal{M}\rangle_{\text{ER}_{lg}} = \mathcal{M}$ as can easily be seen from substituting Equation 3.11 on page 56 into (5.15). If $\mathcal{M}$ is an exponential family that is not full and that contains a model for some $\beta \neq 0$, then entropification serves to make it full (to see this, check Equation 3.11 once more).

We see that full exponential families, and hence 'full' maximum entropy model classes, are closed under entropification.

$\mathcal{M}$ is not an exponential family This case is more interesting. Many useful probabilistic model classes are not of the exponential form; as a simple example, consider hidden Markov Models [123, 125]. For such model classes, entropification can nevertheless be useful, for two reasons: (1) it leads to reliable estimates of the logarithmic error in the sense of Definition 4.4 (this fact will be put to use in Section 5.3.4); and (2), using $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$ instead of $\mathcal{M}$ can often lead to additional compression of the data when data is encoded using the MDL 2-part code (as will be discussed in Section 5.4.2).

5.2.3 Properties of Entropification

In this subsection we present some useful properties of entropified model classes $\langle\mathcal{M}\rangle_{\mathrm{ER}}$. These properties will be used in the proofs of our main results in Section 5.3.

The key to proving all the properties is that for each fixed $H \in \mathcal{M}$, the subclass of models $\langle H\rangle_{\mathrm{ER}}$ containing (H, β) for all $\beta \in \Gamma_{nat}(H)$ (that is, $\langle H\rangle_{\mathrm{ER}} := \{(H, \beta) \mid (H, \beta) \in \langle\mathcal{M}\rangle_{\mathrm{ER}}\}$) is essentially (though not strictly) a maximum entropy model class (compare Definition 5.2 with Definition 3.7 on page 55). The reason that the correspondence is not strict is that $\langle H\rangle_{\mathrm{ER}}$ is a class of *conditional* models. This leads to some technical, but not essential complications in proving the properties. In order to focus on what is really essential, we moved some of the proofs of these properties to an appendix to this chapter (Appendix 5.6, page 111).

The Properties

Throughout the remainder of this subsection, we assume that we are given a sample space $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$, a model class $\mathcal{M}$ and an error function ER.

Briefly, we will show that (1) even though the distributions indexed by (H, β) are conditional, one can define entropy and expectation of error with respect to these distributions; (2) entropification leads to 'reliable' estimates of error (in the sense of Definition 4.4); (3) for fixed H, the models (H, β) are all, in an important sense, equivalent, but they differ in that they all have different entropy; and (4) there exists a particularly well-behaved class of error functions which we will call 'simple'. Below we show all properties in detail. In stating them, we need to use the maximum likelihood estimator for *fixed hypothesis* H and similarly, for *fixed* β, which we now define:

Definition 5.5 *Let $D = (x^n, y^n) \in \mathbf{E}^n$. The maximum likelihood estimator of D for* fixed H *with respect to $\langle\mathcal{M}\rangle_{\mathrm{ER}}$, denoted by $\hat{\beta}(D|H)$, is (if it exists) given by*

$$\hat{\beta}(D|H) = \arg\max_{\beta \in \Gamma_{nat}(H)} \{P(y^n|H, \beta, x^n)\} \tag{5.16}$$

The maximum likelihood estimator of D for fixed β *with respect to $\langle\mathcal{M}\rangle_{\mathrm{ER}}$, denoted by $\widehat{H}(D|\beta)$, is (if it exists) given by*

$$\widehat{H}(D|\beta) = \arg\max_{H \in \mathcal{M}} \{P(y^n|H, \beta, x^n)\} \tag{5.17}$$

Notational Conventions

The first property we show allows us to simplify notation. Recall that we assume throughout this chapter that $Z_H(\beta)$ does not depend on x (page 88). As will be shown in Appendix 5.6 this implies that for fixed H and β, the expectation of the error under (H, β) is independent of the given x. The same holds for the entropy. Formally, we let $E_{(H,\beta)}$ denote expectation under the model $P(\cdot|H, \beta, \cdot) \in \langle\mathcal{M}\rangle_{\text{ER}}$. Then for all $(H, \beta) \in \langle\mathcal{M}\rangle_{\text{ER}}$ and all $x_1, x_2 \in \mathbf{E}_x$, we have

$$E_{(H,\beta)}[\text{ER}(Y|H,X)|X = x_1] = E_{(H,\beta)}[\text{ER}(Y|H,X)|X = x_2] \tag{5.18}$$

Also, for all $x_1, x_2 \in \mathbf{E}_x$, the entropy $\mathcal{H}(P(\cdot|H, \beta, \cdot))$ satisfies:

$$\mathcal{H}(P(\cdot|H, \beta, x_1)) = \mathcal{H}(P(\cdot|H, \beta, x_2)) \tag{5.19}$$

(5.18) and (5.19) will be proven formally in Appendix 5.6. They imply that the expectation $E_{(H,\beta)}[\text{ER}(Y|H,X)]$ over the conditional model (H, β) supplied with an arbitrary distribution P_x over $\mathbf{E}_x$ does not depend on P_x. This allows us to write $E_{(H,\beta)}[\text{ER}(Y|H,X)]$ instead of $E_{(H,\beta)}[\text{ER}(Y|H,X)|X = x]$. Similarly, we will write $\mathcal{H}(H, \beta)$ instead of $\mathcal{H}(P(\cdot|H, \beta, x))$.

Entropification and Reliable Estimates of Errors

We proceed to show that entropification leads to reliable estimates of error. This will be the key to proving the theorems on entropification which we prove in the next section. In Example 5.1 we discussed why 'reliability' is a desirable property.

Proposition 5.6 [reliability] *Let $D = (x^n, y^n)$. For each $H \in \mathcal{M}$, $E_{(H,\beta)}[\text{ER}(Y|H,X)]$ as a function of β is continuous. Moreover,*

$$E_{(H,\hat{\beta}(D|H))}[\text{ER}(Y|H,X)] = \frac{1}{n}\sum_{i=1}^{n} \text{ER}(y_i|H, x_i)$$

This proposition will be proven in Appendix 5.6. It shows that, for each model $(H, \hat{\beta}(D|H))$, its expected error over future data is equal to its average error over the given data. By Definition 4.4 (Chapter 4), this implies that for each $H \in \mathcal{M}$, the average error $\text{ER}(y|H, x)$ can be reliably estimated on the basis of the restriction of the class $\langle\mathcal{M}\rangle_{\text{ER}}$ to models containing this specific H.

The Name 'Entropification'

We now state two properties that explain why we have chosen the name 'entropification': let $H \in \mathcal{M}$ be arbitrary but fixed. The models (H, β) in $\langle\mathcal{M}\rangle$ are, for all $\beta \in \Gamma_{nat}(H)$ except $\beta = 0$, partially equivalent to H as stand-alone in the sense that they leave the ordering (in terms of goodness-of-fit) that they impose on the data unchanged: the ordering with respect to the original error function equals the new ordering with respect to the logarithmic error. Yet the models (H, β) are all different in the sense that they all have different entropies. We show both properties below.

Proposition 5.7 *Let $H \in \mathcal{M}$ and let x^n, y^n and z^n be such that* $\text{ER}(y^n|H, x^n) > \text{ER}(z^n|H, x^n)$. *Then for all $\beta \in \Gamma_{nat}(H)$ with $\beta > 0$,*

$$-\ln P(y^n|H, \beta, x^n) > -\ln P(z^n|H, \beta, x^n).$$

while for all $\beta < 0$, $-\ln P(y^n|H, \beta, x^n) < -\ln P(z^n|H, \beta, x^n)$.

Proof: Immediate from Definition 5.2. □

Hence for each H, entropification either leaves unchanged or reverses the ordering in terms of goodness-of-fit that H imposes on the data: for every β, the ordering with respect to $\text{ER}(\cdot|H, x^n)$ is identical or reversed to the ordering with respect to the code length (or 'logarithmic error') $-\ln P(\cdot|H, \beta)$.

For the second property mentioned above, recall that $\mathcal{H}(H, \beta)$ stands for the entropy of the model $P(\cdot|H, \beta, x)$ (for arbitrary x) restricted to single outcomes in $\mathbf{E}_y$.

Proposition 5.8 *For all $\beta \in \Gamma_{nat}(H)$ with $\beta > 0$, the entropy $\mathcal{H}(H, \beta)$ is a strictly decreasing function of β. For all $\beta < 0$, $\mathcal{H}(H, \beta)$ is a strictly increasing function of β.*

The proposition is proven in Appendix 5.6. Together with Proposition 5.7 it tells us that for fixed H and varying β, the compound models (H, β) can all be seen as 'versions' of H with different entropies.

Entropy and Expected Error

Suppose $\beta > 0$. As β increases, the entropy $\mathcal{H}(H, \beta)$ decreases. One may expect that, with decreasing entropy (and thus decreasing 'inherent disorder'), the expected error $E_{(H,\beta)}[\text{ER}(Y|H, X)]$ also decreases. This relation indeed holds, but only if $\beta > 0$ (the case $\beta < 0$ will be explored briefly in Section 5.3.3): if $\beta < 0$, then (as we saw above), the entropy is an increasing function of β while the expected error remains a decreasing function of β. In general, let, for fixed H, $\mathbf{U}_H$ be the smallest (possibly unbounded) interval in $\mathbf{R}$ such that $\forall (x, y) \in \mathbf{E} : \text{ER}(y|H, x) \in \mathbf{U}_H$. Then

Proposition 5.9 *$E_{(H,\beta)}[\text{ER}(Y|H, X)]$ is a strictly decreasing function of β. For each t in the interior of $\mathbf{U}_H$ there exists a unique value of β such that $E_{(H,\beta)}[\text{ER}(Y|H, X)] = t$.*

The proposition is proven in Appendix 5.6.

Simple Error Functions

Some error functions, among which the squared error and the 0/1-error, turn out to have a useful additional property which automatically makes them satisfy our regularity conditions for error functions and which makes sure that entropification leaves the relative ordering of hypotheses in terms of goodness-of-fit for given data unchanged. We call such hypotheses *simple*:

Definition 5.10 *If* ER *is such that for all $H_1, H_2 \in \mathcal{M}$ and all β,*

$$Z_{H_1}(\beta) = Z_{H_2}(\beta),$$

where $Z_H(\beta)$ is defined as in Equation 5.6, then we call ER *a* simple *error function for $\mathcal{M}$.*

The two error functions we have encountered earlier are both simple, as shown by the following proposition:

Proposition 5.11 *Let $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$, let $\mathcal{M}$ be an arbitrary class of functions $H : \mathbf{E}_x \to \mathbf{E}_y$. If $\mathbf{E}_y = \mathbf{R}$, then the squared error function ER_{sq} is simple for $\mathcal{M}$. If $\mathbf{E}_y = \{0,1\}$, then the 0/1-error function ER_{01} is simple for $\mathcal{M}$.*

Proof: In the squared error case, $Z_{H,x}(\beta) = \int_{y \in \mathbf{E}_y} \exp(-\beta \text{ER}_{sq}(y|H,x))dy = \int \exp(-\beta(y - H(x))^2)dy = \sqrt{\pi/\beta}$ which does not depend on either H or x. The case of the 0/1-error is analogous. $\Box$

For model classes entropified with simple error functions we can drop the subscript from $Z_H(\beta)$ and simply write $Z(\beta)$. By calculating the entropy $\mathcal{H}(H,\beta)$ of the model (H,β) it follows immediately that this entropy depends only on β and not on H.

Simple error functions have an additional important property which is dual to the property expressed by Proposition 5.7. Whereas in that proposition, we showed that the ordering in goodness-of-fit imposed on data by a hypothesis (H,β) is identical for all β (up to their sign), the present result shows that in the case of simple error functions, a reverse property holds too: the ordering in goodness-of-fit imposed on hypotheses (H,β) by given data D is identical for all β (up to their sign):

Proposition 5.12 *Let ER be a simple error function for $\mathcal{M}$ and let $D = (x^n, y^n)$. For each $\beta_1, \beta_2 \in \Gamma_{nat}$ with $\beta_1, \beta_2 > 0$ or $\beta_1, \beta_2 < 0$ and each $H_1, H_2 \in \mathcal{M}$ we have:*

$$P(y^n|H_1,\beta_1,x^n) > P(y^n|H_2,\beta_1,x^n) \quad \Rightarrow \quad P(y^n|H_1,\beta_2,x^n) > P(y^n|H_2,\beta_2,x^n)$$

and, in particular, if $\beta_1, \beta_2 > 0$ and there exists a single H that minimizes the empirical error $\text{ER}(y^n|H,x^n)$, then:

$$\widehat{H}(D|\beta_1) = \widehat{H}(D|\beta_2) = \arg\min_{H \in \mathcal{M}} \{\text{ER}(y^n|H,x^n)\} \tag{5.20}$$

Hence the H in the tuple (H,β) that maximizes the likelihood is independent of β (except for the sign of β) and (if $\beta > 0$) is equal to the $H \in \mathcal{M}$ that minimizes the empirical error.

Proof: Immediate from instantiating $P(y^n|H_i,\beta_i,x^n)$ using Definition 5.2 and the fact that $Z(\beta)$ does not depend on H. $\Box$

We have now established all basic properties of entropification. We will use these properties to prove the main results of this chapter, concerning its behaviour in the i.i.d. case. Before we do this, we give a summary of the possible interpretations of β.

5.2.4 Interpretations of β

The parameter β as part of the hypothesis (H,β) can be interpreted in the following ways:

1. β determines the expected error $E_{(H,\beta)}[\text{ER}(Y|H,X)]$ (Proposition 5.9); hence ...

2. ... when (H, β) is inferred from data D, β serves as an estimate of the error H will make on future data.

3. β determines the entropy $\mathcal{H}(H, \beta)$ (Proposition 5.8): the closer $|\beta|$ to 0, the larger $\mathcal{H}(H, \beta)$.

4. β determines how strongly the error $\text{ER}(y^n|H, x^n)$ is weighed in the code based on $P(\cdot|H, \beta, \cdot)$, which has lengths $L(y^n|H, x^n) = \beta\text{ER}(y^n|H, x^n) + n \ln Z_H(\beta)$ (Equation 5.9): the closer $|\beta|$ to 0, the closer P is to the uniform distribution.

The last two items show that β can be interpreted as a kind of 'noise' level, measuring for each fixed H *the apparent randomness of the data with respect to hypothesis* H (we use the word 'apparent' because a small value of β does not mean that the data are random in any general sense; it only means that H *does not give very much information about the data*). From the polynomial example we see by the substitution $\beta = 1/(2\sigma^2)$ that small β means large variance σ^2 and hence a large amount of 'noise' with respect to H. From the concept learning example (Example 5.3) we see by the substitution $\beta = \ln(1 - \theta) - \ln(\theta)$ that a hypothesis H together with a small (absolute) value of β expresses the extended hypothesis that the probability that H makes a mistake is θ, where θ is close to $1/2$ (uniform). $\beta = 0$ cancels the effect of the hypothesis altogether and makes $P(\cdot|H)$ equal to the uniform distribution.

β and temperature The idea to turn an error function $\text{ER}(\cdot|\cdot)$ into a class of probability distributions $P(\cdot|\cdot, \beta) = Z(\beta)^{-1} \exp(-\beta\text{ER}(\cdot|\cdot))$ is actually not new: it is common practice in Statistical Mechanics [155, 69] where the error function is called 'energy function' and in stead of a parameter β one uses a parameter T (called a 'temperature') satisfying $\beta = 1/T$. Such 'energy functions' and 'temperatures' are frequently used outside of a purely physical context. As far as we know, the role of the temperature T is somewhat different from our β since T is not treated as a parameter to be estimated from data.

5.3 Entropification and Generating Distributions

In this section we present our main results concerning entropification. We study the behaviour of entropified model classes when data are independently distributed according to some unknown 'true' distribution P^*. Roughly, it is shown that with an entropified model class $\langle \mathcal{M} \rangle_{\text{ER}}$, if given enough data, we can find the model in $\langle \mathcal{M} \rangle_{\text{ER}}$ with the smallest expected prediction error under P^*. Additionally, this model will provide a correct estimate of the average prediction error over future data that it will achieve; hence the model gives a good impression of 'how good it really is' when errors are measured by ER. The important thing is that this model that is both optimal and 'reliable' will be found *even if* P^* *is not contained in* $\langle \mathcal{M} \rangle_{\text{ER}}$. Below we state a technical lemma that is needed in our theorems, which follow in sections 5.3.1-5.3.5. The proof of the lemma is quite involved and is not directly related to entropification. Therefore it has been deferred to an appendix (Appendix 5.7, page 114). The proofs of the theorems are directly related to the main properties of entropification, so they can be found in this section. These theorems concern the general case (Section 5.3.2), the case of simple error functions (Section 5.3.3), the case of the logarithmic error (Section 5.3.4) and the case of the squared error (Section 5.3.5). The lemma and the theorems all assume that the data are i.i.d. according to some unknown but fixed probability distribution P^*. We have to impose some mild conditions on P^*. These amount to there being some 'window' (i.e. a bounded set containing more than one element) within which all data will fall. The reason is that otherwise the required expectation $E_{P^*}[\text{ER}(Y|H,X)]$ may not exist. Here is a formal definition of this condition:

Definition 5.13 (Regularity Condition for the True Distribution)
Let a sample space $\mathbf{E} = \mathbf{E}_1 \times \ldots \times \mathbf{E}_m$ be given for $m \geq 1$. Whenever in the following we speak of a 'true', or 'generating' distribution P^*, we assume P^* to be a distribution over $\mathbf{E}_{P^*} = \mathbf{E}_{P^*,1} \times \ldots \times \mathbf{E}_{P^*,m}$ with full support such that for $1 \leq i \leq m$, (a) $\mathbf{E}_{P^*,i} \subseteq \mathbf{E}_i$ (b) $\mathbf{E}_{P^*,i}$ contains more than one element and (c) if $\mathbf{E}_i$ is continuous, then $\mathbf{E}_{P^*,i}$ is compact.

We now state our technical lemma. Essentially, it says the following: if $\mathcal{M}$ is compactly parameterized, then the average code length of x^n based on the maximum likelihood model for x^n converges (with probability 1) to the expected code length based on the model in the class that minimizes this expected code length. This holds if data are i.i.d. according to some P^* satisfying Definition 5.13. Note that P^* is *not* required to be a member of $\mathcal{M}$.

Lemma 5.14 *Let $\mathcal{M} = \{P(\cdot|\theta) \mid \theta \in \Gamma\}$ be a class of i.i.d. probabilistic models over sample space $\mathbf{E}$ that is finitely parameterized by $\Gamma \subset \mathbf{R}^k$ where Γ is compact. Let the data be i.i.d. according to some P^* satisfying Definition 5.13. Then the following minima exist for all n, all $x^n \in \mathbf{E}^n$:*

$$\begin{aligned} \hat{L}(x^n) &:= \min_{\theta \in \Gamma}\{-\ln P(x^n|\theta)\} \\ \widetilde{L}(P^*) &:= \min_{\theta \in \Gamma}\{E_{P^*}[-\ln P(X|\theta)]\} \end{aligned}$$

We have with P^*-probability 1:

$$\lim_{n\to\infty} \frac{1}{n}\hat{L}(x^n) = \widetilde{L}(P^*)$$

The proof is given in Appendix 5.7. Lemma 5.14 is the key to the theorems which will be proven in the next subsections.

5.3.1 The Theorems

This subsection gives an informal overview of the theorems we are about to prove. We start by defining an analogue of the definition of 'reliable' (Definition 4.4) for the setting where some true (i.i.d.) distribution is assumed to exist.

Definition 5.15 *Let the data be i.i.d. according to some distribution P^*. Let P be some given probabilistic model over $\mathbf{E}$ and let $\psi : \mathbf{E} \to \mathbf{U}$ be some given function. We call P* reliable with respect to ψ under P^* *if*

$$E_P[\psi(X)] = E_{P^*}[\psi(X)].$$

A model P that is reliable with respect to ψ under P^* is (with probability 1) guaranteed to give a correct impression of the average $\overline{\psi(x)}^n$ for large n: by the law of large numbers, $\overline{\psi(x)}^n \to E_{P^*}[\psi(X)] = E_P[\psi(X)]$ as n increases, with P^*-probability 1.

Informal Discussion of the Theorems Let $\langle \mathcal{M} \rangle_{\text{ER}}$ be a model class entropified with respect to an error function ER. Let the data be i.i.d. according to some arbitrary P^* (not necessarily in $\langle \mathcal{M} \rangle_{\text{ER}}$). The main point of *Theorem 5.16* is that for each $H \in \mathcal{M}$, there exists a unique $\tilde{\beta}_H$ such that (1) $E_{(H,\tilde{\beta}_H)}[\text{ER}(Y|H,X)] = E_{P^*}[\text{ER}(Y|H,X)]$ (hence $(H,\tilde{\beta}_H)$ is reliable with respect to ER under P^*), and (2), $\hat{\beta}(D|H)$, the maximum likelihood estimator for fixed H (Definition 5.5), converges with P^*-probability 1 to $\tilde{\beta}_H$. Hence for each H, a reliable estimate of its performance can, with probability 1, be obtained.

If the error function ER is *simple* (Definition 5.10), then in addition, the stronger *Theorem 5.17* applies. Its essence is (roughly) that the maximum likelihood estimator $(\widehat{H},\hat{\beta})$ converges with P^*-probability 1 to the model $(\widetilde{H},\tilde{\beta})$ where $\widetilde{H}$ is the optimal model in $\mathcal{M}$, minimizing the 'true' expected error $E_{P^*}[\text{ER}(Y|\widetilde{H},X)]$, and $\tilde{\beta}$ is such that $E_{(\widetilde{H},\tilde{\beta})}[\text{ER}(Y|\widetilde{H},X)] = E_{P^*}[\text{ER}(Y|\widetilde{H},X)]$, and so $(\widetilde{H},\tilde{\beta})$ is reliable for $\text{ER}(\cdot|\widetilde{H},\cdot)$ under P^*. Hence the optimal $\widetilde{H}$ and a reliable estimate of its performance can both, with probability 1, be obtained.

If the error function is not simple, then things get more complicated. Nevertheless, *Theorem 5.18* shows that in the special case of the (non-simple) logarithmic error function, an analogue to the above (maximum likelihood estimators converging to an optimal and reliable model) still holds.

The squared error function *is* simple but satisfies an additional interesting property, as will be shown in *Theorem 5.19.*

Use of ML Estimators in the Theorems In the theorems we make use of the maximum likelihood estimator $\hat{\beta}(D|H)$ for fixed H which is defined as in Definition 5.5. It is straightforward to show that, under our conditions for the sample space of the generating distribution P^*, a unique value of $\hat{\beta}(D|H)$ always exists. Sometimes we will also make use of the full maximum likelihood estimator $(\widehat{H},\hat{\beta})(D)$. In all these cases, it is straightforward to show that there exists at least one maximum of the likelihood. We use the convention that, if several (H,β) maximizing the likelihood exist, then $(\widehat{H},\hat{\beta})$ denotes the first one according to some prespecified ordering over $\langle \mathcal{M} \rangle_{\text{ER}}$.

5.3.2 General Error Functions

In this subsection we present Theorem 5.16 which is applicable to all regular error functions.

Theorem 5.16 *Let $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$. Let $\mathcal{M}$ be a class of models and let $\text{ER} : \mathbf{E} \times \mathcal{M} \to \mathbf{R}$ be an error function for $\mathcal{M}$. Assume that data (x,y) are generated by independent sampling from a distribution P^* over $\mathbf{E}_{P^*}$ as in Definition 5.13. Then for all fixed $H \in \mathcal{M}$, $E_{P^*}[\text{ER}(Y|H,X)]$ exists, and*

1. *there exists a unique $\tilde{\beta}_H$ depending on H such that*

$$E_{(H,\tilde{\beta}_H)}[\text{ER}(Y|H,X)] = E_{P^*}[\text{ER}(Y|H,X)]. \tag{5.21}$$

and at the same time, for all $\beta \in \Gamma_{nat}(H)$ with $\beta \neq \tilde{\beta}_H$,

$$E_{P^*}[-\ln P(Y|\beta,H,X)] > E_{P^*}[-\ln P(Y|\tilde{\beta}_H,H,X)] \tag{5.22}$$

2. *with P^*-probability 1,*

$$\lim_{n\to\infty} \hat{\beta}(x^n, y^n|H) = \tilde{\beta}_H \tag{5.23}$$

and hence

$$\lim_{n\to\infty} E_{(H,\hat{\beta}(D|H))}[\text{ER}(Y|H,X)] = E_{P^*}[\text{ER}(Y|H,X)] \tag{5.24}$$

Proof: We only prove the theorem for continuous $\mathbf{E}_x$ and $\mathbf{E}_y$. The case where $\mathbf{E}_x$ or $\mathbf{E}_y$ or both are discrete is completely analogous.

We first prove existence of $E_{P^*}[\text{ER}(Y|H,X)]$. Definition 5.13 tells us that P^* is defined over a compact subspace of $\mathbf{E}$. Conditions **C1**-**C3** on ER (see the remark at the beginning of Section 5.2) make sure that $\text{ER}(y|H,x)$ is continuous at all $(x,y) \in \mathbf{E}$. For continuous $\mathbf{E}$, we only consider P^* with associated continuous density functions (see Chapter 1, Section 1.2). Existence of $E_{P^*}[\text{ER}(Y|H,X)]$ now follows.

Now to prove item 1, note that by Definition 5.13, $E_{P^*}[\text{ER}(Y|H,X)]$ must lie in the interior of $\mathbf{U}_H$. Therefore, by Proposition 5.9 there must be a unique value $\tilde{\beta}_H$ for which (5.21) holds.

We have, for each β, $E_{P^*}[-\ln P(Y|H,\beta,X)] = \beta E_{P^*}[\text{ER}(Y|H,X)] + \ln Z_H(\beta)$. By differentiating with respect to β one verifies that $E_{P^*}[-\ln P(Y|H,\beta,X)]$ as a function of β is convex and reaches its unique minimum at the value $\tilde{\beta}_H$ for which (5.21) holds. This proves (5.22).

Concerning item 2, we will first prove (5.24). Since the data are i.i.d. we can apply the strong law of large numbers [47] which gives that with P^*-probability 1, $n^{-1}\sum_{i=1}^{n} \text{ER}(y_i|H,x_i)$ converges to $E_{P^*}\text{ER}(Y|H,X)$. (5.24) then follows by the reliability of estimates of ER (Proposition 5.6). (5.23) is now immediate by (5.24) and the fact (which we just showed) that $E_{P^*}[-\ln P(Y|\beta,H,X)]$ as a function of β is convex and reaches its single maximum at $\beta = \tilde{\beta}_H$. □

We now proceed to the special case where ER is a simple error function.

5.3.3 Simple Error Functions

For simple error functions Proposition 5.12 applies: if $\beta > 0$, then minimization of logarithmic error $-\ln P(y^n|H,\beta,x^n)$ corresponds to minimization of the error function ER. This allows us to prove Theorem 5.17, which says that the maximum likelihood estimator $(\widehat{H},\hat{\beta})$ for data D converges to a model $(\widetilde{H},\tilde{\beta})$ where, if $\tilde{\beta} > 0$, then $\widetilde{H}$ minimizes the 'true' expected error $E_{P^*}[\text{ER}(Y|H,X)]$ over all $H \in \mathcal{M}$. Let us briefly consider the case $\tilde{\beta} < 0$. In the case of the squared error, $\Gamma_{nat}(H)$ only contains positive parameter values, so then always $\tilde{\beta} > 0$ and the problem does not occur. In the special case of the 0/1-error, something interesting happens which we illustrate with an example. Suppose our concept class $\mathcal{M}$ contains only two models H_1 and H_2. Suppose P^* to be such that $E_{P^*}[\text{ER}(Y|H_1, X)] = 0.3$ and $E_{P^*}[\text{ER}(Y|H_2, X)] = 0.9$. Then the hypothesis minimizing the expected 0/1-error is clearly H_1. However, H_2 can be trivially modified into another 'inverse' hypothesis $\overline{H_2}$ with $E_{P^*}[\text{ER}(Y|\overline{H_2}, X)] = 0.1$: $\overline{H_2}(x)$ predicts 1 if $H_2(x) = 0$ and 0 otherwise. This trivial modification can be achieved by entropification: the entropified model $(\widetilde{H}, \tilde{\beta})$ that leads to the shortest expected code length will in our example be given for $\widetilde{H} = H_2$ and $\tilde{\beta} < 0$; the fact that $\tilde{\beta} < 0$ makes H_2 behave like its inverse $\overline{H_2}$. H_2 will lead to much shorter (expected) code lengths than H_1 (all this can be easily checked using equations 5.11 and 5.12 of Example 5.3, page 89).

Theorem 5.17 *Let* $\mathbf{E}$, *data* (x, y), P^* *and* E_{P^*} *be as in the statement of Theorem 5.16. Let* ER *be a simple error function and assume* $\mathcal{M}$ *to be such that* $\langle\mathcal{M}\rangle_{\text{ER}}$ *is finitely parameterized by* $\Gamma_{\mathcal{M}} \times \Gamma_{nat}$ *where* $\Gamma_{\mathcal{M}}$ *is compact. Then*

1. *The following minima exist:*

$$\widetilde{\text{ER}}(P^*) := \min_{H \in \mathcal{M}} E_{P^*}[\text{ER}(Y|H, X)] \tag{5.25}$$

$$\widetilde{L}(P^*) := \min_{P(\cdot|\theta,\cdot) \in \langle\mathcal{M}\rangle_{\text{ER}}} E_{P^*}[-\ln P(Y|\theta, X)] \tag{5.26}$$

Let $\tilde{\theta}$ *be one of the models for which the minimum in (5.26) is obtained. Then*

2. $\tilde{\theta} = (\widetilde{H}, \tilde{\beta})$ *for some* $\tilde{\beta} \in \Gamma_{nat}$. *If* $\tilde{\beta} > 0$, *then* $\widetilde{H}$ *is (one of) the hypothesis (hypotheses) for which the minimum in (5.25) is obtained (* $\tilde{\beta}$ *is identical for all such* $\widetilde{H}$*).*

Let $(\widehat{H}, \hat{\beta}) := (\widehat{H}, \hat{\beta})(D)$ *denote the maximum likelihood estimator in* $\langle\mathcal{M}\rangle_{\text{ER}}$.

3. *We have:*

$$\begin{aligned} &\lim_{n\to\infty} E_{(\widehat{H},\hat{\beta})}[\text{ER}(Y|\widehat{H}, X)] = \\ &\quad E_{(\widetilde{H},\tilde{\beta})}[\text{ER}(Y|\widetilde{H}, X)] = E_{P^*}[\text{ER}(Y|\widetilde{H}, X)] \end{aligned} \tag{5.27}$$

Hence, for each 'true', 'generating' distribution P^* there exists an optimal model $(\widetilde{H}, \tilde{\beta})$ such that the 'true' expectation under P^* of the error $\text{ER}(Y|\widetilde{H}, X)$ is minimal and *equal* to the expectation of this error under $(\widetilde{H}, \tilde{\beta})$: when given enough data, every reasonable (Chapter 4, Definition 4.3) inference procedure will hit upon a model that is optimal in this sense.

Proof: We only prove the theorem for continuous $\mathbf{E}_x$ and $\mathbf{E}_y$. The case where $\mathbf{E}_x$ or $\mathbf{E}_y$ or both are discrete is completely analogous.

Concerning items 1 and 2, existence of $\widetilde{\text{ER}}(P^*)$ is straightforward by compactness of $\Gamma_{\mathcal{M}}$. Existence of $\widetilde{L}(P^*)$ and item 2 will now be proven at the same time. First write

$$E_{P^*}[-\ln P(Y|\theta, X)] = \beta E_{P^*}[\text{ER}(Y|H,X)] + \ln Z(\beta) \tag{5.28}$$

for $(H, \beta) = \theta$. Since we assume ER to be simple here, $Z(\beta)$ does not depend on H. This shows that for each fixed $\beta > 0$, $E_{P^*}[-\ln P(Y|(H,\beta),X)]$ reaches its minimum for the set $\mathcal{H}^+$ of H minimizing $E_{P^*}[\text{ER}(Y|H,X)]$. By differentiating with respect to β and the fact that $Z(\beta)$ does not depend on H, one finds that there exists a single $\tilde{\beta}^+$ which minimizes $E_{P^*}[-\ln P(Y|(H,\beta),X)]$ for all $H \in \mathcal{H}^+$. For fixed $\beta < 0$, $E_{P^*}[-\ln P(Y|(H,\beta),X)]$ reaches its minimum for the set $\mathcal{H}^-$ of H maximizing $E_{P^*}[\text{ER}(Y|H,X)]$ (which exists by compactness of $\Gamma_{\mathcal{M}}$). In this case there exists a single $\tilde{\beta}^-$ which minimizes $E_{P^*}[-\ln P(Y|(H,\beta),X)]$ for all $H \in \mathcal{H}^-$. If $\beta = 0$, then $E_{P^*}[-\ln P(Y|(H,\beta),X)]$ reaches its minimum for all $H \in \mathcal{M}$. From this it easily follows that a $(\tilde{\beta}, \widetilde{H})$ minimizing (5.28) exists, that all minima of (5.28) have the same component $\tilde{\beta}$ and that if $\tilde{\beta} > 0$, then $\widetilde{H} \in \mathcal{H}^+$. This proves both existence of $\widetilde{L}(P^*)$ and item 2 of the theorem.

The key to the proof of item 3 is the result (5.33) below. In order to obtain this result we need to apply Lemma 5.14 of page 96. The lemma cannot be simply applied to model classes $\langle \mathcal{M} \rangle_{\text{ER}}$, since these contain conditional rather then regular probabilistic models. To avoid this problem we change $\langle \mathcal{M} \rangle_{\text{ER}}$ into a class of essentially equivalent regular probabilistic models over $\mathbf{E}_x \times \mathbf{E}_y$ by extending it with the uniform distribution over $\mathbf{E}_x$ (this is possible since $\mathbf{E}_x$ is compact by condition **C4**; see the beginning of Section 5.2) (for motivation, see Lemma 5.23 in Appendix 5.6 where a similar trick is used).

Let P^u be the uniform distribution over $\mathbf{E}_x$ and let

$$P^u(x^n, y^n|H, \beta) = P(y^n|H, \beta, x^n)P^u(x^n) \tag{5.29}$$

be the distribution that extends each conditional distribution $P(\cdot|H,\beta,\cdot)$ to a full distribution over $\mathbf{E}_x \times \mathbf{E}_y$. We have for all $H \in \mathcal{M}$:

$$-\ln P^u(x^n, y^n|H, \tilde{\beta}) = -\ln P(y^n|H, \tilde{\beta}, x^n) + C \cdot n \tag{5.30}$$

for some constant C. Here $\tilde{\beta}$ is as in the statement of the theorem, item 2. Let

$$\widetilde{L}^u(P^*) := \min_{H \in \mathcal{M}} E_{P^*}[-\ln P^u(X, Y|H, \tilde{\beta})] = \widetilde{L}(P^*) + C \tag{5.31}$$

Here C is the same constant as in (5.30). Finally, let $\langle \mathcal{M} \rangle_{\tilde{\beta}} = \{P(\cdot,\cdot|H,\tilde{\beta}) \mid H \in \mathcal{M}\}$ be the class of probabilistic models of form (5.29) for which $\beta = \tilde{\beta}$.

It is straightforward to check that $\langle \mathcal{M} \rangle_{\tilde{\beta}}$ is such that Lemma 5.14 of page 96 applies. Substituting $\hat{L}(x^n) = -n^{-1} \ln P^u(x^n, y^n|\widehat{H}, \tilde{\beta})$, this gives that with P^*-probability 1,

$$\lim_{n\to\infty} -\frac{1}{n} \ln P^u(x^n, y^n|\widehat{H}, \tilde{\beta}) = \widetilde{L}^u(P^*) \tag{5.32}$$

(5.30), (5.31) and (5.32) give us (with P^*-probability 1):

$$\lim_{n\to\infty} -\frac{1}{n} \ln P(y^n|\widehat{H}, \tilde{\beta}, x^n) = \widetilde{L}(P^*) = E_{\tilde{\beta}}[-\ln P(Y|\widetilde{H}, X)] \tag{5.33}$$

for all the $\widetilde{H}$ minimizing (5.25), where the last equality follows from item 2 in the statement of the theorem which we proved already. By the identity $-\ln P(y^n|H,\beta,x^n) = \beta\text{ER}(y^n|H,x^n) + n\ln Z(\beta)$ this gives with P^*-probability 1:

$$\lim_{n\to\infty} -\frac{1}{n}\tilde{\beta}\text{ER}(y^n|\widehat{H}, x^n) + \ln Z(\tilde{\beta}) = \tilde{\beta} E_{\tilde{\beta}}[\text{ER}(Y|\widetilde{H}, X)] + \ln Z(\tilde{\beta}) \tag{5.34}$$

By reliability of estimates of ER (Proposition 5.6) we have $n^{-1}\text{ER}(y^n|\widehat{H}, x^n) = E_{(\widehat{H},\hat{\beta})}[\text{ER}(Y|\widehat{H}, X)]$. Plugging this into (5.34) proves (5.27). □

5.3.4 Non-Simple Error Functions; Logarithmic Error

It is, in general, difficult to analyze for non-simple error functions whether an analogue of Theorem 5.17 holds. The proof of Theorem 5.17 is based on the fact that, for simple error functions, minimization of logarithmic error corresponds to minimization (or maximization) of the error function ER. For non-simple error functions this need not be the case since $Z(\beta)$ varies with H. However, a special case occurs if $\mathcal{M}$ is probabilistic and we entropify with respect to the logarithmic error. In that case, the function $\text{ER} = \text{ER}_{lg}$ measures itself the log-likelihood of the data, while the optimal model in $\langle\mathcal{M}\rangle_{\text{ER}}$ is also optimal with respect to expected log-likelihood. This allows an analogue to Theorem 5.17 to be proven after all; it is embodied in Theorem 5.18 below. Since in this situation, it turns out to be somewhat harder to identify exact conditions under which the required minima exist, we will keep things simple and simply assume $\mathcal{M}$ to be such that they exist. Specifically, let $\mathcal{M}$ be a class of probabilistic models over sample space $\mathbf{E}$ that is finitely parameterized by $\Gamma_{\mathcal{M}}$ and let $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ be the entropification of $\mathcal{M}$ under the logarithmic error. $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ can be parameterized by $\Gamma_{\mathcal{M}} \times \Gamma_{nat}(H)$. Let data $x_1, x_2, \ldots$ be generated by independent sampling from a distribution P^* over $\mathbf{E}_{P^*}$ where P^* is as in Definition 5.13. We assume that (1) $\mathbf{E}_{P^*}$ is such that for all n, $x^n \in \mathbf{E}^n_{P^*}$, the maximum likelihood estimator of x^n in $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$, denoted by $\hat{\theta} := \hat{\theta}(x^n)$ exists and falls within a compact subset of $\Gamma_{\mathcal{M}} \times \Gamma_{nat}(H)$, and (2),

$$\widetilde{L}(P^*) := \min_{\theta\in\Gamma_{\mathcal{M}}\times\Gamma_{nat}(H)} E_{P^*}[-\ln P(X|\theta)] \tag{5.35}$$

exists and is obtained by a single model $\tilde{\theta}$.

Theorem 5.18 *Let $\mathcal{M}$, P^* and $\tilde{\theta}$ be as above. Then with P^*-probability 1:*

$$\lim\nolimits_{n\to\infty} E_{\hat{\theta}(x^n)}[-\ln P(X|\hat{\theta}(x^n))] = E_{\tilde{\theta}}[-\ln P(X|\tilde{\theta})] = E_{P^*}[-\ln P(X|\tilde{\theta})]. \tag{5.36}$$

Proof: In the proof we assume the notation of Example 5.4. Specifically, $P(\cdot|\eta)$ stands for the model in $\mathcal{M}$ indexed by η. $P(\cdot|(\eta,\beta)) = Z_\eta^{-1}(\beta)\exp(\beta \ln P(\cdot|\eta))$ stands for the model in $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ indexed by (η,β). Note also that $\hat\theta = (\hat\eta,\hat\beta)$. By reliability of the estimates of ER_{lg} when the class $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ is used (Proposition 5.6) we find

$$E_{(\hat\eta,\hat\beta)}[-\ln P(X|\hat\eta)] = -\frac{1}{n}\sum_{i=1}^{n}\ln P(x_i|\hat\eta)$$

By straightforward calculation this gives

$$E_{(\hat\eta,\hat\beta)}[-\ln P(X|(\hat\eta,\hat\beta))] = -\frac{1}{n}\sum_{i=1}^{n}\ln P(x_i|(\hat\eta,\hat\beta)) \tag{5.37}$$

Let $\langle\mathcal{M}\rangle'_{\text{ER}_{lg}}$ be the restriction of $\langle\mathcal{M}\rangle_{\text{ER}_{lg}}$ to models with parameter values in the compact set within which $\hat\theta(x^n)$ must fall (we assumed that such a set exists). Clearly, $\tilde\theta$ must be a member of this set. Now we can apply Lemma 5.14 (page 96) to $\langle\mathcal{M}\rangle'_{\text{ER}}$. This gives, with P^*-probability 1,

$$\lim_{n\to\infty} -\frac{1}{n}\ln P(x^n|\hat\theta(x^n)) = \widetilde{L}(P^*). \tag{5.38}$$

Together, (5.37) and (5.38) show that (5.36) holds. □

5.3.5 Special Status of the Squared Error

In classical statistics, the problem of curve-fitting is cast in the following terms: one assumes data to be independently generated by some unknown distribution P^* and one tries to identify the function H^* (called the 'regression function') that, for each x, gives the expected value (the mean) of Y given that $X = x$. (some would prefer to say: one assumes that data are generated by some function H^* with *errors* distributed according to P^*). Whatever the distribution of the errors, this can be achieved by using the squared error function in the learning phase, as will be shown in Theorem 5.19 below. Such results have been known for a long time [20]. For completeness and since it is not difficult, we have included Theorem 5.19 nevertheless.

Let $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$ where $\mathbf{E}_x \subset \mathbf{R}$ and $\mathbf{E}_y = \mathbf{R}$. We assume data to be generated by independent sampling from P^* as in Definition 5.13. Let $H^*(x) = E_{P^*}[Y|X = x]$. That is, $H^*(x)$ gives the mean of Y for each $x \in \mathbf{E}_x$. We will assume that $H^*(x)$ is continuous at all $x \in \mathbf{E}_x$. Let $(\sigma^*)^2 = E_{P^*}[(Y - H^*(X))^2]$. Hence $(\sigma^*)^2$ denotes the 'expected true variance' of Y. We already know that, for a given class $\mathcal{M}$ of functions $\mathbf{E}_x \to \mathbf{E}_y$, $\langle\mathcal{M}\rangle_{\text{ER}_{sq}}$ consists of conditional Gaussian distributions as given by Equation 5.2 on page 84 (Example 5.1). These distributions are obtained from the natural parameterization of $\langle\mathcal{M}\rangle_{\text{ER}_{sq}}$ by substituting $\beta = 1/2\sigma^2$. Under the natural parameterization, theorems 5.16 and 5.17 are applicable. The following theorem extends these theorems for the specific case of $\langle\mathcal{M}\rangle_{\text{ER}_{sq}}$. Briefly, the only non-trivial results that are added are the following: (1) for every P^*, the optimal model

$P(\cdot|\tilde{\sigma}^2, \widetilde{H}) = P(\cdot|\tilde{\beta}, \widetilde{H})$ will be such that $\widetilde{H}$ is the function in $\mathcal{M}$ that is closest (in the mean squared error sense) to the 'true' function H^* and (2) $\tilde{\sigma}^2$ can be interpreted as the mean squared error of $\widetilde{H}$. Since, for every P^*, $(\widehat{H}, \hat{\sigma}^2)$ will converge to $(\widetilde{H}, \tilde{\sigma}^2)$ with P^*-probability 1, this implies that in the special case with $H^* \in \mathcal{M}$, $\widehat{H}$ will converge to the true H^* and $\hat{\sigma}^2$ will converge to the true variance $(\sigma^*)^2$ with P^*-probability 1. This holds *independently* of whether $P^*(\cdot|x)$ is Gaussian or not.

In the following theorem, we assume models in $\langle\mathcal{M}\rangle_{\mathrm{ER}_{sq}}$ to be specified by (H, σ^2) rather than (H, β).

Theorem 5.19 *Let* $\mathbf{E}$, $\mathcal{M}$, $\langle\mathcal{M}\rangle_{\mathrm{ER}_{sq}}$ *and* P^* *be as above. Then:*

1. *For all* $H \in \mathcal{M}$, $E_{P^*}[\mathrm{ER}_{sq}(Y|H, X)]$ *exists, and there exists a* $\tilde{\sigma}^2_H$ *depending on* H *such that*

$$E_{(H,\tilde{\sigma}^2_H)}[(Y - H(X))^2] = \\ E_{P^*}[(Y - H(X))^2] = (\sigma^*)^2 + E_{P^*}[(H^*(X) - H(X))^2] \tag{5.39}$$

2. *Further assume* $\mathcal{M}$ *to be such that* $\langle\mathcal{M}\rangle_{\mathrm{ER}_{sq}}$ *is finitely parameterized by* $\Gamma_{\mathcal{M}} \times \Gamma_{nat}$ *where* $\Gamma_{\mathcal{M}}$ *is compact. Then the following minimum exists:*

$$\begin{aligned} \tilde{\sigma}^2 &:= \min_{H \in \mathcal{M}} E_{P^*}[(Y - H(X))^2] = \\ &= (\sigma^*)^2 + \min_{H \in \mathcal{M}} E_{P^*}[(H^*(X) - H(X))^2] \end{aligned} \tag{5.40}$$

Let $\widetilde{H}$ *be one of the models for which the minimum in (5.40) is obtained. We have:*

$$\lim_{n \to \infty} E_{(\widehat{H},\hat{\sigma}^2)}[(Y - \widehat{H}(X))^2] = E_{(\widetilde{H},\tilde{\sigma}^2)}[(Y - \widetilde{H}(X))^2] = \tilde{\sigma}^2 = \\ = (\sigma^*)^2 + \min_{H \in \mathcal{M}} E_{P^*}[(H^*(X) - H(X))^2] \tag{5.41}$$

In particular, if $H^* \in \mathcal{M}$, *then* $\hat{\sigma}^2$ *converges with probability 1 to the true variance* $(\sigma^*)^2$ *and* $\widehat{H}$ *converges to the true hypothesis* H^*.

Proof: Most of item (1) of the theorem is straightforward from Theorem 5.16; the only thing that still needs to be proven is the fact that

$$E_{P^*}[(Y - H(X))^2] = (\sigma^*)^2 + E_{P^*}[(H^*(X) - H(X))^2] \tag{5.42}$$

Item (2) follows Theorem 5.17. The only thing that is not immediate from this theorem is, once more, (5.42), and additionally

$$E_{(\widetilde{H},\tilde{\sigma}^2)}[(Y - \widetilde{H}(X))^2] = \tilde{\sigma}^2. \tag{5.43}$$

It is a standard fact of regression [6] (also straightforward to verify by calculation) that for all σ^2 and H, we have $E_{(H,\sigma^2)}[(Y - H(X))^2] = \sigma^2$. This shows (5.43). (5.42) is a variation of the well-known bias-variance decomposition [55], also straightforward to prove:

$$
\begin{aligned}
&E_{P^*}[(Y - H(X))^2] - E_{P^*}[(Y - H^*(X))^2] \stackrel{(1)}{=} \\
&E_{P^*(X)}[E_{P^*(Y|X)}[2YH^*(X) - 2YH(X) + H(X)^2 - H^*(X)^2 | X = x]] \stackrel{(2)}{=} \\
&E_{P^*}[(H(X) - H^*(X))^2]
\end{aligned}
\tag{5.44}
$$

(1) follows from using the linearity of expectation, working out the squares and conditioning on X. (2) is obtained by using $E_{P^*}[Y|X = x] = H^*(x)$. From (5.44), the equality (5.42) is immediate. □

5.4 Entropification and MDL

Is entropification merely a convenient tool to make predictions reliable or are there additional reasons as to why we should 'entropify' our model classes? In this section we show that if we use the MDL Principle as our statistical inference procedure, then it is often a good idea to use an entropified model class for at least two further reasons: first, entropification can serve to optimize the trade-off between hypothesis complexity and goodness-of-fit as needed in the two-part MDL code and the stochastic complexity. Second, it leads to codes for non-probabilistic model classes that can be justified in terms of minimizing expected code length. These points are discussed in Section 5.4.1 and Section 5.4.2 respectively.

5.4.1 Entropification and Model Selection

There have been different proposals in the literature on how to deal with two-part codes and stochastic complexity codes for non-probabilistic model classes in the MDL framework. For simplicity we will restrict our discussion to the two-part codes. Recall that in the basic, probabilistic case, we select the H minimizing

$$- \log P(x^n|H) + L_{C_1}(H) \tag{5.45}$$

where C_1 is some code used for encoding the parameters indexing the hypothesis. Rissanen ([128], page 18/19) proposes to turn a non-probabilistic model class into a probabilistic class $\mathcal{M}_{pr}$ (which essentially corresponds to entropification with $\beta = 1$). This leads to finding the H minimizing

$$\beta \text{ER}(y^n|H, x^n) + n \ln Z_H(\beta) + L_{C_1}(H) \tag{5.46}$$

with $\beta = 1$. The problem here is that choosing $\beta = 1$ is essentially arbitrary but can have large consequences: choosing a different value of β we may end up, at least for small n, with H of very different complexity (the closer β to 0, the larger the relative weight of the complexity term in (5.46)).

Barron [13] proposes to select the hypothesis H minimizing the sum of the empirical error $\text{ER}(y^n|H, x^n)$ and the square root of the complexity term times the sample size,

$\sqrt{n \cdot L_{C_1}(H)}$. While this criterion can be shown to have some strong asymptotic properties, it is in a sense not faithful to the MDL principle since the resulting sum does not have a natural interpretation as a code length.

Yamanishi [167] proposes to minimize $\beta \text{ER}(y^n|H, x^n) + L_{C_1}(H)$ for some β whose value is made dependent on the size of the training set. Again, this has strong asymptotic properties, but again, it is not clear how to interpret the resulting sum from a purely coding-theoretic point of view.

Instead, we propose to entropify $\mathcal{M}$ and then use (5.46) (now with an additional term $L_{C_1}(\beta)$ added to account for the number of bits needed to encode β). We think there are several advantages to using entropified model classes. We first note that, at least non-asymptotically, using the entropified class $\langle \mathcal{M} \rangle_{\text{ER}}$ can lead one to choose different models for the same data than when using $\mathcal{M}_{pr}$ for fixed β. We give a little example.

Example 5.20 Consider a class of continuous functions $\mathcal{M}$ entropified with the squared error (as in Example 5.1, page 82). Let data $D = (x^n, y^n)$ and model $H \in \mathcal{M}$ be given. Denote the average squared error H makes on D by $\overline{\text{ER}_{sq}}$. Using the code (5.46) for fixed β, we obtain as total description length of the y^n given the x^n:

$$L(y^n; H|x^n) = n(\overline{\text{ER}_{sq}} + \frac{1}{2} \ln \frac{\pi}{\beta}) + L_{C_1}(H) \tag{5.47}$$

while using (5.46) for the entropified model (H, β) where $\beta = \hat{\beta}(D|H)$ is the parameter that maximizes the likelihood of D given H, we obtain:

$$L'(y^n; H|x^n) = \frac{1}{2} n(1 + \ln 2\pi + \ln \overline{\text{ER}_{sq}}) + L_{C_1}(\beta) + L_{C_1}(H) \tag{5.48}$$

which depends logarithmically rather than linearly on the average error. (both equations can be easily verified by substituting $\beta = 1/2\sigma^2$ as in Example 5.1). When two-part code MDL is used, the MDL-optimal β_{mdl} for given D and fixed H will not be equal to $\hat{\beta}$, but nevertheless it will be reasonably close. $L_{C_1}(\beta)$ will be equal to $1/2 \log n + c$ for some constant c (for both facts see Chapter 1, Section 1.4). This implies that there can very well be hypotheses H_1 and H_2 with a different number of parameters (so $L_{C_1}(H_1) \neq L_{C_2}(H_2)$) such that H_1 minimizes (5.47) while H_2 minimizes (5.48). In such a case, two-part code MDL based on the entropified model class leads to a different optimal H.

Using (5.46) with entropified model classes $\mathcal{M}_{\text{ER}}$ allows the sum (5.46) (now with the additional term $L_{C_1}(\beta)$) to be interpreted as a code length. By *learning* the optimal value of β from the data (which is what entropification in the two-part code setting amounts to), we essentially choose the value that allows for the shortest code length of the data, which is in line with the general MDL philosophy. Moreover, since each model in $\langle \mathcal{M} \rangle_{\text{ER}}$ corresponds to a code, we can also define stochastic complexity with respect to such model classes in the usual way and use it as a basis of model class selection; it allows us to compare different model classes based on different error functions for the same data, since the performance of all the classes are measured using the same criterion, namely, the codelength. This is once again in line with the general

MDL philosophy of using code lengths as a 'universal yardstick' [128], to be employed whenever different models or model classes are to be compared for the same data.

Another thing to be said for entropification is that it unifies different instantiations of MDL. In the existing literature on MDL, the question of how to code the data given an hypothesis has been given different answers depending on the category of model class used. For probabilistic model classes, generally the Shannon-Fano code with $L(D) = -\log P(D)$ is used [128, 14]. For *concept* classes (classes consisting of functions $\mathbf{E}_x \rightarrow \{0, 1\}$; see Chapter 2, Example 2.4), the usual approach (see e.g. [122, 84]) has been to explicitly code the mistakes a hypothesis H makes on data D; see Example 5.21 below for details. This is different from all the versions of the two-part code we have seen before. For the case of non-probabilistic model classes with arbitrary error functions ER, there have been several proposals, as we saw above. Entropification (where data $D = (x^n, y^n)$ given hypothesis (H, β) is encoded using the code with lengths $-\log P(y^n|H, \beta, x^n))$ is an approach to handle non-probabilistic model classes that contains the existing treatments of probabilistic and concept classes as special cases. In the probabilistic case, as long as the model class is a full exponential family, then entropification will not change anything; we showed this in Example 5.4. In the case where $\mathcal{M}$ is a concept class, the code based on entropification with respect to the 0/1-error (Example 5.3), while superficially different, is essentially equivalent to the traditional approach of coding the mistakes H makes on D. We show this formally in Example 5.21 below. This suggests (but does not prove of course) that entropification can serve as the general 'preprocessing' tool to make a single version of two-part code MDL applicable to essentially arbitrary model classes.

Example 5.21 [concept learning and Bernoulli parameters II] Let $\mathcal{M}$ be a class of concepts (Chapter 2, Example 2.4) over $\mathbf{E} = \mathbf{E}_x \times \{0, 1\}$ and let the observational data $D = (x^n, y^n)$. Two-part codes for concept classes are traditionally (e.g. in [122] and [84]) based on the following coding scheme: the x_i are regarded as given. The y_i are encoded by first encoding an hypothesis $H \in \mathcal{M}$ and then encoding the *exceptions* to H, which are all the indices i for which $y_i \neq H(x_i)$. We assume that hypotheses are encoded using some fixed code $C_1 : \mathcal{M} \rightarrow \mathbf{B}^*$. Clearly, given the x_i, H and the list of exceptions $M = \{i_1, \ldots, i_k\}$ we can fully reconstruct $y_1, \ldots, y_n$ (for x_i with $i \notin M$ we set $y_i = H(x_i)$; for x_i with $i \in M$ we set $y_i = |1 - H(x_i)|$).

If H makes k mistakes on a sample D of length n, there are $\binom{n}{k}$ different exception sets $M = \{i_1, \ldots, i_k\}$. Hence we need $\ln \binom{n}{k} + L(k)$ nats to encode all these mistakes. Here $L(k) = O(\ln k)$ equals the number of nats needed to encode k using some prefix code for the numbers $0, \ldots, n$ (note that k has to be encoded too to allow unique decoding). The total description length of the y_i given the x_i becomes:

$$L(y^n, H|x^n) = \ln \binom{n}{k} + L(k) + L_{C_1}(H) \tag{5.49}$$

Another way to arrive at a two-part code for the data would be to first entropify the concept class $\mathcal{M}$ with respect to the 0/1-error function and then to encode data by first coding some H, then some parameter β (using a fixed code C'_1) and then encoding the y_i using the code based on $P(\cdot|H,\beta)$. This would take

$$L(y^n, H, \beta | x^n) = -\ln P(y^n|H,\beta,x^n) + L_{C_1'}(\beta) + L_{C_1}(H) \text{ nats.} \tag{5.50}$$

We proceed to show that (5.49) and (5.50) approximately coincide and hence that both ways of coding the data are essentially equivalent.

We saw in Example 5.3 (page 89) that, by substituting $\beta := \ln(1-\theta) - \ln\theta$, we find that the class $\langle\mathcal{M}\rangle_{\text{ER}_{01}}$ can equivalently be parameterized as follows: $\langle\mathcal{M}\rangle_{\text{ER}_{01}} = \{P(\cdot|H,\theta,\cdot) \mid H \in \mathcal{M}, 0 < \theta < 1\}$, such that, if $\text{ER}_{01}(y^n|H,x^n) = k$, then

$$P(y^n|H,\theta,x^n) = \theta^k(1-\theta)^{n-k} \tag{5.51}$$

Instead of coding β we can also code the θ corresponding to it. We can therefore rewrite (5.50) as follows:

$$L(y^n, H, \theta | x^n) = -\ln P(y^n|H,\theta,x^n) + L_{C_1'}(\theta) + L_{C_1}(H) \text{ nats.} \tag{5.52}$$

where $P(y^n|H,\theta,x^n)$ is given by (5.51). The ML estimator $\hat{\theta}$ maximizing (5.51) for fixed H and (x^n, y^n) is given by $\hat{\theta} = k/n$. Based on H and $\hat{\theta}$, the number of nats $-\ln P(y^n|H,\hat{\theta},x^n)$ needed to code the data becomes

$$-\ln P(y^n|H,\hat{\theta},x^n) = -\ln \hat{\theta}^k(1-\hat{\theta})^{n-k} \overset{(1)}{=} n\mathcal{H}(\hat{\theta}) \overset{(2)}{\approx} \ln \binom{n}{k}$$

where (1) follows by straightforward calculation and (2) by Stirling's approximation (3.15) (page 58) (or by applying (3.16) on page 58 with $\gamma = k/n$) (for precise bounds on $|n\mathcal{H}(\hat{\theta}) - \ln\binom{n}{k}|$ see [30], Chapter 12). Since $\hat{\theta} = k/n$ and hence, when n is known (as we assume in this example), can be reconstructed from k only, we need approximately $L(k)$ nats to describe $\hat{\theta}$, where $L(k)$ is defined as above. The total description length of the y_i then becomes:

$$-\ln P(y^n, H, \hat{\theta}|x^n) \approx \ln \binom{n}{k} + L(k) + L_{C_1}(H) \tag{5.53}$$

which is seen to coincide with (5.49). Hence if we code the data based on the entropified model class $\langle\mathcal{M}\rangle_{\text{ER}_{01}}$ and use the optimal β (corresponding to the optimal θ) for given D and H, then the number of bits we need coincides with the number of bits needed to efficiently encode the exceptions.

5.4.2 Worst-Case Optimality of the Shannon-Fano Code

The arguments given in the previous subsection suggest that entropification can serve as a general means to apply two-part code MDL to non-probabilistic model classes. Of course, they do not *prove* that entropification will be as well-behaved as either Barron's or Yamanishi's approach to this problem. Whether it really is or not is a challenging (and difficult) question at the time of writing this thesis. We did prove one small result though that provides some extra justification for entropification. This is discussed in the present subsection.

In Chapter 1 we saw that when a probabilistic model class is used in two-part code MDL, data is encoded by first encoding some model θ and then coding the data based on the *Shannon-Fano code* $L(x^n|\theta) = -\log P(x^n|\theta)$. At page 17 we already asked ourselves why to use this code and not any other one. There are many other possibilities; to give an example, we could map each model θ to the code with lengths $L'(x^n|\theta) = -\log(\sqrt{P(x^n|\theta)}/\sum_{z^n in \mathbf{E}^n} \sqrt{P(z^n|\theta)})$ which, by the Kraft Inequality (Theorem 1.6) also corresponds to a probability distribution over $\mathbf{E}^n$. Why does the Shannon-Fano code have a special status? In Chapter 1 we gave the justification that, using the Shannon-Fano code, the code length of the data precisely reflects the probability: if $P(D_1|\theta) = a \cdot P(D_2|\theta)$, then $L(D_1|\theta) = L(D_2|\theta) + \log a$.

Some authors prefer a different (or at any rate, additional) justification based on the Information Inequality (Equation 3.4 on page 52): if θ turns out to be the 'true' model, i.e. data is generated by repeated sampling from θ, then the *expected* code length $E_\theta[L(X^n)]$ is minimized if we set $L(x^n) := -\log P(x^n|\theta)$. By using the Shannon-Fano code, we map each model θ to the code that will be optimal if θ is actually true; hence it is the code that best 'suits' θ. This justification of the use of the Shannon-Fano code can be found in, for example [37, 162, 160].

We have always had some doubts about this argument, for two reasons: (a) it does not say anything about the (realistic) case where the model class contains models that allow us to compress the data (hence we can learn something about the data) yet none of these models are close to the 'true' one generating the data; (b) it is not clear how to extend this argument to non-probabilistic model classes.

Proposition 5.22 below shows how entropification allows us to extend the Shannon-Fano argument to a more general case which includes non-probabilistic model classes. Whereas we still assume the existence of *some* true probability distribution generating the data, we do not assume any more that it is contained in the model class $\mathcal{M}$ under consideration. For simplicity, we will consider only the unconditional, 'unsupervised' case, where we are interested in coding the complete outcomes (and not just y-values conditioned on x-values). Formally, we consider a class $\mathcal{M}$ of models and an error function $\text{ER} : \mathbf{E} \times \mathcal{M} \to \mathbf{R}$ (in case $\mathcal{M}$ is probabilistic we take ER to be the logarithmic error). We let $\theta = (H, \beta)$ index a model in $\langle \mathcal{M} \rangle$. Let $\mathcal{G}$ be the class of probability distributions P^* over $\mathbf{E}$ satisfying

$$E_{(H,\beta)}[\text{ER}(X|H)] = E_{P^*}[\text{ER}(X|H)]. \tag{5.54}$$

Let $\mathcal{L}$ be the class of all code length functions $L : \mathbf{E} \to \mathbf{R} \cup \{\infty\}$ satisfying the Kraft Inequality $\sum_{x \in \mathbf{E}} 2^{-L(x)} \leq 1$.

We are now ready to state our proposition. We discuss its implications after the proof.

Proposition 5.22 *Let $\mathcal{M}$, ER, $\mathcal{G}$, θ and $\mathcal{L}$ be as above. Let $L(\cdot|\theta) \in \mathcal{L}$ be the code length function of the Shannon-Fano code for $\theta = (H, \beta)$, restricted to one outcome $x \in \mathbf{E}$. That is,*

$$L(x|H, \beta) = -\log P(x|H, \beta) = \beta\text{ER}(x|H) + \ln Z_H(\beta)$$

We have:

1.

$$L(\cdot|\theta) = \arg\inf_{L \in \mathcal{L}} \sup_{P^* \in \mathcal{G}} E_{P^*}[L(X|\theta)] \tag{5.55}$$

That is, $L(\cdot|\theta)$ gives the shortest worst-case expected code lengths, the worst-case being taken over all distributions satisfying (5.54).

2. *Let, for given H, $\mathbf{U}_H$ be the smallest interval such that $\forall x : \text{ER}(x|H) \in \mathbf{U}_H$. For every $H \in \mathcal{M}$ and for every $P^* \in \mathcal{G}$ for which $E_{P^*}[\text{ER}(X|H)]$ lies in the interior of $\mathbf{U}_H$, there exists a β such that (5.54) holds.*

Proof: Define $t := E_{(H,\beta)}[\text{ER}(X|H)] = E_{P^*}[\text{ER}(X|H)]$. As is clear from our regularity conditions for error functions (page 87), the probability distribution $P(\cdot|H,\beta)$ is the maximum entropy distribution for constraint $E[\text{ER}(X|H)] = t$. By Figure 3.1 on page 62 we have that $E_{P^*}[L(X|H,\beta)] = E_{(H,\beta)}[L(X|H,\beta)] = \mathcal{H}(H,\beta)$ for every $P^* \in \mathcal{G}$. On the other hand, let $L' \in \mathcal{L}$ be a code length function different from $L(\cdot|H,\beta)$. By Figure 3.1 there exists a $P^* \in \mathcal{G}$ (namely, $P^* = P(\cdot|H,\beta)$) such that $E_{P^*}[L'(X)] > \mathcal{H}(H,\beta)$. This proves (1).

To prove (2), note that the class of maximum entropy models for function $\text{ER}(X|H)$ coincides with the class of models in $\langle\mathcal{M}\rangle_{\text{ER}}$ restricted to fixed H. Let us denote this subclass by $\mathcal{M}_{me}$. By Proposition 3.9 (Chapter 3, page 56), for each t in the interior of $\mathbf{U}$, $\mathcal{M}_{me}$ contains a model satisfying $E[\text{ER}(X|H)] = t$. This proves (2). □

This proposition shows that the Shannon-Fano code for models θ in entropified model classes (a) leads to codes that are worst-case optimal if the probability distribution θ is 'true' only in the sense that its expectation of the error coincides with the true expectation of error, and (b) that entropified model classes *always* (except possibly for P^* with expected errors at the boundaries of the error space) contain a model that is 'true' in this weak respect.

If one uses a non-probabilistic model class $\mathcal{M}$, one usually does not have a clear idea about the distribution generating the data. If one is at all willing to assume that such a distribution nevertheless exists[1], then it seems reasonable to make as few assumptions as possible about it. This directly leads to our worst-case scenario, which really says that *every* i.i.d. distribution is a possible candidate for generating the data. That is why we regard this proposition as justifying the use of the Shannon-Fano code for the entropification (probabilistic version) of $\mathcal{M}$.

We hasten to add though that there do exist codes (based on non-i.i.d. model classes) whose expected code lengths under every P^* are arbitrarily close to that of the Shannon-Fano code. An example is the code based on the universal computer language that was discussed in Chapter 1, Section 1.1.2.

The proposition also has something to say about the case where $\mathcal{M}$ itself is probabilistic and we entropify with respect to the logarithmic error ER_{lg}. If $\mathcal{M}$ is itself an exponential family, this will not change $\mathcal{M}$ (Example 5.4), and the proposition tells us

[1] Of course, one may argue that if one uses a non-probabilistic model class, then the data generating process may be inherently non-probabilistic. But then it becomes very hard to prove anything about it. In any case, the present setting (assuming an underlying distribution while working with non-probabilistic model classes) is taken for granted by many researchers [167, 13, 69].

that the Shannon-Fano code for $\mathcal{M}$ is optimal not only in the case that data is generated by one of the models in $\mathcal{M}$, but also in the case that it is generated by some i.i.d. model not in $\mathcal{M}$. If $\mathcal{M}$ is not an exponential family, then the usual optimality of the Shannon-Fano code holds for the models in $\mathcal{M}$, while the 'worst-case' optimality holds, by Proposition 5.22, for the models in $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$. Whether one should entropify or not then depends on whether one thinks that one of the models in the class will be very close to being 'truely a true model': if one entropifies, one adds an extra dimension to the parameter space. This can lead to logarithmically larger code lengths; if $\mathcal{M}$ contains the true model, then it will lead even with probability 1 to larger codelengths of the data, when data is encoded using either the stochastic complexity or the two-part MDL code. However, if the true model is not in $\mathcal{M}$, then using $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$ instead of $\mathcal{M}$ can sometimes lead to a linear *decrease* in codelengths. We briefly show why.

To see that if $\mathcal{M}$ contains the true model, one will need more bits to encode the data based on $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$ rather than $\mathcal{M}$, we use a result by Clarke and Barron [27]. They proved an analogue of the asymptotic expansion of stochastic complexity (Chapter 2, Section 2.6) for the case where data is distributed according to one of the models in a (probabilistic) model class $\mathcal{M}$. Let $\mathcal{M}$ be a probabilistic model class consisting of i.i.d. models. Clarke and Barron showed that, if the data are generated by one of the models θ^* in $\mathcal{M}$, then under some mild further conditions on $\mathcal{M}$,

$$L_{sc}(x^n|\mathcal{M}) = -\log P(x^n|\theta^*) + \frac{k}{2}\log n + O(1) \tag{5.56}$$

with θ^*-probability 1. Here $L_{sc}(x^n|\mathcal{M})$ is the stochastic complexity of x^n with respect to $\mathcal{M}$ and k is the number of parameters needed for parameterizing $\mathcal{M}$. By the results discussed in Chapter 2, Section 2.6, the two-part code length is within $O(1)$ of the stochastic complexity. Observe that if the true model θ^* is in $\mathcal{M}$, then it is also in $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$; therefore, by (5.56), using $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$, the number of parameters k is increased by 1 which results in a logarithmic increase in codelength (with probability 1).

If θ^* is not in $\mathcal{M}$ then, supposing $\mathcal{M}$ is finitely parameterized by $\Gamma \in \mathbf{R}^k$, the asymptotic expansion of both stochastic complexity and two-part code (Chapter 2, Section 2.6) gives

$$L_{sc}(x^n|\mathcal{M}) = -\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log n + O(1)$$

By applying Lemma 5.14, one sees that with θ^*-probability 1,

$$\lim_{n\to\infty} -\frac{1}{n}L_{sc}(x^n|\mathcal{M}) = \min_{\theta\in\Gamma} E_{\theta^*}[-\ln P(X|\theta)]$$

This will hold for both the original model class $\mathcal{M}$ and its entropified version $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$ (in the latter case, the values in β have to be restricted to a compact set). By Proposition 5.22, there exist θ^* such that

$$\min_{\theta\in\langle\Gamma\rangle_{\mathrm{ER}_{lg}}} E_{\theta^*}[-\ln P(X|\theta)] < \min_{\theta\in\Gamma} E_{\theta^*}[-\ln P(X|\theta)],$$

where we used $\langle\Gamma\rangle_{\mathrm{ER}_{lg}}$ to denote the parameterization of $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$. In such a case, both the two-part and the stochastic complexity code based on $\langle\mathcal{M}\rangle_{\mathrm{ER}_{lg}}$ will clearly achieve more compression (by a linear amount) than the codes based on $\mathcal{M}$. Since $\mathcal{M} \subset \langle\mathcal{M}\rangle_{\text{ER}_{lg}}$, the opposite event (the code based on $\mathcal{M}$ achieving a linear gain in compression compared to the code based on $\langle M\rangle_{\text{ER}_{lg}}$) has zero probability for any θ^*.

5.5 Conclusion

We have introduced the concept of 'entropification' and shown how it can be used in the context of estimating prediction error and in the context of MDL. We leave detailed conclusions to the Epilogue to Part I of this thesis (page 119), where we discuss how the results obtained in this chapter can be used to partially resolve the problematic issues concerning MDL that were raised in the Introduction and in Chapters 1-3.

5.6 Appendix: Entropification and Maximum Entropy Model Classes

In this appendix we prove the properties of entropification that we presented in Section 5.2.3 (Equations 5.18 and 5.19; Propositions 5.6-5.9).

Let $\mathcal{M}$ be a class of models, ER be an error function and $\langle\mathcal{M}\rangle_{\text{ER}}$ be the entropification of $\mathcal{M}$. $\langle H\rangle_{\text{ER}}$, the subclass of models from $\mathcal{M}$ restricted to fixed H (i.e. $\langle H\rangle_{\text{ER}} := \{(H,\beta) \mid (H,\beta) \in \langle\mathcal{M}\rangle_{\text{ER}}\}$) is essentially a maximum entropy model class. The properties we want to show would follow immediately if it were really such a class. However, since $\langle H\rangle_{\text{ER}}$ is a class of *conditional* models, we need to use a trick: we extend $\langle\mathcal{M}\rangle_{\text{ER}}$ to a class of distributions over $\mathbf{E}_x \times \mathbf{E}_y$ by supplying it with the uniform distribution over $\mathbf{E}_x$; this distribution exists since we assume $\mathbf{E}_x$ to be either finite or compact. As will be shown below, the resulting model class, which we denote by $\langle\mathcal{M}\rangle^u_{\text{ER}}$, *is* a maximum entropy model class. We then use standard results about such classes to prove certain properties for $\langle\mathcal{M}\rangle^u_{\text{ER}}$, and we then show that if these properties hold for $\langle\mathcal{M}\rangle^u_{\text{ER}}$, they must also hold for the class of conditional distributions $\langle\mathcal{M}\rangle_{\text{ER}}$. This will be done in Lemma 5.23 below. After having proved the lemma, we will show that the properties presented in the main text follow as immediate corollaries from this lemma.

Lemma 5.23 *Let $\mathbf{E} = \mathbf{E}_x \times \mathbf{E}_y$ and let* ER $: \mathbf{E} \times \mathcal{M} \to \mathbf{R} \cup \{\infty\}$ *be an error function. Let $P^u(\cdot)$ be the uniform distribution over $\mathbf{E}_x$. Let $\langle\mathcal{M}\rangle^u_{\text{ER}}$ be the class of probabilistic models $P^u(\cdot,\cdot|H,\beta)$ where, for each (H,β) in $\mathcal{M}$, for each (x^n,y^n), $P^u(x^n,y^n|H,\beta) = P(y^n|H,\beta,x^n)P^u(x^n)$. We have:*

1. *There exists a constant $c \in \mathbf{R}$ such that for all n, all $(x^n,y^n) \in \mathbf{E}^n$,*

$$P(y^n|\beta,H,x^n) \cdot c^n = P^u(x^n,y^n|\beta,H) \tag{5.57}$$

Let, for fixed $H \in \mathcal{M}$, $\langle H\rangle^u_{\text{ER}} = \{(H,\beta)|(H,\beta) \in \langle\mathcal{M}\rangle^u_{\text{ER}}\}$ be the restriction of $\langle\mathcal{M}\rangle^u_{\text{ER}}$ to models with fixed H.

2. *$\langle H\rangle^u_{\text{ER}}$ is the maximum entropy model class for the function $\phi(x,y) := \text{ER}(y|H,x)$ with range $\mathbf{U}$. Here $\mathbf{U}$ is the smallest (open or closed) interval in $\mathbf{R}$ such that $\forall(x,y) \in \mathbf{E} : \phi(x,y) \in \mathbf{U}$.*

Let x be an arbitrary element of $\mathbf{E}_x$. Let $P(\cdot|H,\beta,x)$ be the distribution over $\mathbf{E}_y$ given by $P(y|H,\beta,x) = Z_H(\beta)^{-1}\exp(-\beta\text{ER}(y|H,x))$ and let $\mathcal{H}(P(\cdot|H,\beta,x))$ stand for the entropy of the distribution $P(\cdot|H,\beta,x))$.

3. *We have for all (H,β):*

$$E_{P^u(\cdot,\cdot|H,\beta)}[\text{ER}(Y|H,X)] = E_{P(\cdot|H,\beta,x)}[\text{ER}(Y|H,X)|X=x] = \\ = -\frac{\partial}{\partial\beta}\ln Z_H(\beta) \quad (5.58)$$

4. *We also have for all (H,β):*

$$\mathcal{H}(P(\cdot|H,\beta,x)) = \mathcal{H}(P^u(\cdot,\cdot|H,\beta)) + \ln c \quad (5.59)$$

Proof: Item (1) is straightforward. Item (2) follows directly by our assumptions on ER (see the remark at the beginning of Section 5.2, page 87) and the definition of $\langle\mathcal{M}\rangle_{\text{ER}}$ (Definition 5.2). To prove item (3), note that for each $P(\cdot|H,\beta,\cdot)$ the corresponding unconditional model $P^u(\cdot,\cdot|H,\beta)$ is given by

$$P^u(x,y|H,\beta) = \frac{c}{Z_{H(\beta)}}\exp(-\beta\text{ER}(y|H,x))$$

Since $\langle\mathcal{M}\rangle^u_{\text{ER}}$ is a maximum entropy model class, we can apply Proposition 3.9 on page 56 to get:

$$E_{P^u(\cdot,\cdot|H,\beta)}[\text{ER}(Y|H,X)] = -\frac{\partial}{\partial\beta}(\ln Z_H(\beta) - \ln c) = -\frac{\partial}{\partial\beta}\ln Z_H(\beta) \quad (5.60)$$

Now choose an arbitrary $x \in \mathbf{E}_x$. Let $\langle H(x)\rangle_{\text{ER}}$ be the class of models containing, for each $\beta \in \Gamma_{nat}(H)$, the distribution $P(\cdot|H,\beta,x)$ defined as in the statement of item (3) of the lemma. $\langle H(x)\rangle$ is a class of maximum entropy models for function $\psi(y) := \text{ER}(y|H,x)$ (note that ψ is a function of y only; x is kept fixed). This is straightforward to verify by our assumptions on ER, see page 87. We can therefore apply Proposition 3.9 on page 56 to give us:

$$E_{P(\cdot|H,\beta,x)}[\text{ER}(Y|H,X)|X=x] = -\frac{\partial}{\partial\beta}\ln Z_H(\beta) \quad (5.61)$$

(5.60) and (5.61) coincide. Since we picked x arbitrarily, (5.58) follows.

Now for item (4). By straightforward calculation we see that the entropy of $P(\cdot|H,\beta,x)$ is equal to

$$\beta E_{P(\cdot|H,\beta,x)}[\text{ER}(Y|H,X)|X=x] + \ln Z(\beta).$$

while the entropy of $P(\cdot,\cdot|H,\beta)$ is given by

$$\beta E_{P(\cdot,\cdot|H,\beta)}[\mathrm{ER}(Y|H,X)] + \ln Z(\beta) - \ln c.$$

Together with (5.58) in item (3) of the proposition, equality (5.59) follows. □

Having proven Lemma 5.23, we can proceed to prove the properties stated in Section 5.2.3.

Proving the Properties

We first note that Equations 5.18 and 5.19 (page 92) follow trivially from items 3 and 4 of Lemma 5.23. This leaves us with propositions 5.6, 5.8 and 5.9, which we now restate and prove.

Proposition 5.6 *Let $D = (x^n, y^n)$. For each $H \in \mathcal{M}$, $E_{(H,\beta)}[\mathrm{ER}(Y|H,X)]$ as a function of β is continuous. Moreover,*

$$E_{(H,\hat{\beta}(D|H))}[\mathrm{ER}(Y|H,X)] = \frac{1}{n}\sum_{i=1}^{n} \mathrm{ER}(y_i|H,x_i) \tag{5.62}$$

Proof: Let $H \in \mathcal{M}$ be fixed. By Lemma 5.23, item (1), the model $P^u(\cdot,\cdot|H,\hat{\beta})$ that maximizes, for fixed H, the likelihood of D within the class of unconditional models $\langle H\rangle^u_{\mathrm{ER}}$ (as defined in Lemma 5.23) is indexed by the same value $\hat{\beta}$ as the model $P(\cdot|H,\hat{\beta},\cdot)$ that maximizes the likelihood of D within the class of conditional models $\langle H\rangle_{\mathrm{ER}}$. Also by Lemma 5.23, $\langle H\rangle^u_{\mathrm{ER}}$ is a maximum entropy model class. Therefore, using Proposition 4.7 about the reliability of estimates for maximum entropy model classes, we have that the expectation under the unconditional model indexed by H and $\hat{\beta}(D|H)$ is equal to the average over data D:

$$E_{P^u(\cdot,\cdot|H,\hat{\beta}(D|H))}[\mathrm{ER}(Y|H,X)] = \frac{1}{n}\sum_{i=1}^{n} \mathrm{ER}(y_i|H,x_i) \tag{5.63}$$

By Lemma 5.23, item 3 this shows (5.62). □

Proposition 5.8 *For all $\beta \in \Gamma_{nat}(H)$ with $\beta > 0$, the entropy $\mathcal{H}(H,\beta)$ is a strictly decreasing function of β. For all $\beta < 0$, $\mathcal{H}(H,\beta)$ is a strictly increasing function of β.*

Proof: We know from Lemma 5.23, item (4) that $\mathcal{H}(H,\beta) = \ln c + \mathcal{H}(P^u(\cdot,\cdot|H,\beta))$ for some constant c. The second term in this expression stands for the entropy of a maximum entropy distribution with parameter β (Lemma 5.23, item (2)). Hence we can apply Proposition 3.9 of page 56, item 2(b) (which says that for maximum entropy distributions, the entropy is a strictly increasing (decreasing) function for $\beta > 0$ ($\beta < 0$)) and the result follows. □

Let, for fixed H, $\mathbf{U}_H$ be the smallest (possibly unbounded) interval in $\mathbf{R}$ such that $\forall(x,y) \in \mathbf{E} : \mathrm{ER}(y|H,x) \in \mathbf{U}_H$.

Proposition 5.9 $E_{(H,\beta)}[\mathrm{ER}(Y|H,X)]$ *is a strictly decreasing function of* β. *For each* t *in the interior of* $\mathbf{U}_H$ *there exists a unique value of* β *such that* $E_{(H,\beta)}[\mathrm{ER}(Y|H,X)] = t$.

Proof: We know from Lemma 5.23, item (3) that for all $\beta \in \Gamma_{nat}(H)$, $E_{(H,\beta)}[\mathrm{ER}(Y|H,X)] = E_{P^u(\cdot,\cdot|H,\beta)}[\mathrm{ER}(Y|H,X)]$. Here (Lemma 5.23, item (2)) $P^u(\cdot,\cdot|H,\beta)$ is a maximum entropy distribution with natural parameter β. Hence we can apply Proposition 3.9 of page 56. The first part of the proposition above follows immediately from item 2(a) of Proposition 3.9 (which says exactly that $E_{P^u(\cdot,\cdot|H,\beta)}[\mathrm{ER}(Y|H,X)]$ is a strictly decreasing function of β). The second part of the proposition above follows immediately from item 5 of Proposition 3.9. □

5.7 Appendix: Proof of Lemma 5.14

In this appendix we prove Lemma 5.14, which is restated below. Essentially, it says the following: if $\mathcal{M}$ is compactly parameterized, then the average code length of x^n based on the maximum likelihood model for x^n converges (with probability 1) to the expected code length based on the model in the class that minimizes this expected code length. This is reminiscent of the 'uniform laws of large numbers' that are central to Vapnik's work [159]. The exact connection needs to be further investigated.

Lemma 5.14 *Let* $\mathcal{M} = \{P(\cdot|\theta) \mid \theta \in \Gamma\}$ *be a class of i.i.d. probabilistic models over sample space* $\mathbf{E}$ *that is finitely parameterized by* $\Gamma \subset \mathbf{R}^k$ *where* Γ *is compact. Let the data be i.i.d. according to some* P^* *satisfying Definition 5.13. Then the following minima exist for all* n*, all* $x^n \in \mathbf{E}^n$*:*

$$\hat{L}(x^n) := \min_{\theta\in\Gamma}\{-\ln P(x^n|\theta)\} \tag{5.64}$$

$$\widetilde{L}(P^*) := \min_{\theta\in\Gamma}\{E_{P^*}[-\ln P(X|\theta)]\} \tag{5.65}$$

We have with P^**-probability 1:*

$$\lim_{n\to\infty} \frac{1}{n}\hat{L}(x^n) = \widetilde{L}(P^*) \tag{5.66}$$

Proof: In the proof we assume that $\mathbf{E}$ is continuous. Adaption to the case of $\mathbf{E}_{P^*} = \mathbf{E}_{P^*,1} \times \ldots \times \mathbf{E}_{P^*,m}$ where some of the $\mathbf{E}_{P^*,i}$ are discrete is completely straightforward.

By compactness of $\mathbf{E}_{P^*}$ and Γ and the fact that we only consider P^* with associated density functions (see Chapter 1, Section 1.2) the minima (5.64) and (5.65) evidently exist. We can cover Γ with a grid of k-dimensional rectangles with side width s. The set Γ is thus partitioned into a finite number, say M, of rectangles R_i. Let, for $1 \leq i \leq M$, θ^i be the model in Γ corresponding to the center of R_i. In this way we obtain a reduced parameter set $\Gamma_s = \{\theta^1, \ldots, \theta^M\}$.

We first consider the simple case where $\mathcal{M}$ is such that the following four minima are all attained by a unique value for each n, $x^n \in \mathbf{E}^n$:

$$\begin{aligned}
\tilde{\theta} &= \arg\min_{\theta \in \Gamma}\{E_{P^*}[-\ln P(X|\theta)]\} \\
\tilde{\theta}_s &= \arg\min_{\theta_s \in \Gamma_s}\{E_{P^*}[-\ln P(X|\theta_s)]\} \\
\hat{\theta}(x^n) &= \arg\min_{\theta \in \Gamma}\{-\ln P(x^n|\theta)\} \\
\hat{\theta}_s(x^n) &= \arg\min_{\theta_s \in \Gamma_s}\{-\ln P(x^n|\theta_s)\} \qquad (5.67)
\end{aligned}$$

We now show in two stages that (5.66) holds in the case where these single minimizing values exist.

Stage 1 Let n and $\epsilon > 0$ be given. We claim that if we pick the rectangle side width s small enough both of the following equations will hold:

$$|E_{P*}[-\ln P(X|\tilde{\theta}_s)] - E_{P*}[-\ln P(X|\tilde{\theta})]| < \frac{1}{3}\epsilon \qquad (5.68)$$

$$| - \frac{1}{n}\ln P(x^n|\hat{\theta}_s(x^n)) + \frac{1}{n}\ln P(x^n|\hat{\theta}(x^n))| < \frac{1}{3}\epsilon \qquad \text{for all } x^n \in \mathbf{E}^*_{P^*} \qquad (5.69)$$

We first show (5.69). Let

$$f_n(\theta, x^n, \theta_0) := - \frac{1}{n}\ln P(x^n|\theta) + \frac{1}{n}\ln P(x^n|\theta_0) = \frac{1}{n}\sum_{i=1}^{n}[-\ln P(x_i|\theta) + \ln P(x_i|\theta_0)] \qquad (5.70)$$

$\mathcal{M}$ is finitely parameterized by Γ. Checking Definition 2.7 on page 35, we see that for all $x \in \mathbf{E}_{P^*}$, $f_1(\theta, x, \theta_0)$ regarded as a function of θ must be continuous at all $\theta \in \Gamma$. By compactness of $\mathbf{E}_{P^*}$ it is easy to show that

$$\begin{aligned}
f_{\max}(\theta_0, \theta) &:= \max_{x \in \mathbf{E}_{P^*}} f_1(\theta, x, \theta_0) \quad \text{and} \\
f_{\min}(\theta_0, \theta) &:= \min_{x \in \mathbf{E}_{P^*}} f_1(\theta, x, \theta_0)
\end{aligned}$$

are well-defined continuous functions of θ with $f_{\max}(\theta_0, \theta_0) = f_{\min}(\theta_0, \theta_0) = 0$. By compactness of Γ, it is clear that the following function is well-defined for all $\delta > 0$:

$$g_{\max}(\delta) := \max_{\mathcal{U}(\delta)}\{\ |f_{\max}(\theta_0, \theta)|\ \}, \qquad (5.71)$$

where the maximum is taken over the set $\mathcal{U}(\delta) = \{\theta, \theta_0 \in \Gamma \mid |\theta - \theta_0| \leq \delta\}$. Moreover (compactness of Γ) one can easily show that $\lim_{\delta \downarrow 0} g_{\max}(\delta) = g_{\max}(0) = 0$. The same holds for $g_{\min}(\delta)$ which we define analogously to $g_{\max}$. These properties of $g_{\max}$ and

$g_{\min}$ show the following: for every $\epsilon > 0$, we can pick the rectangle width s small enough such that the following implication holds for all $\theta, \theta_0 \in \Gamma$ and all $x^n \in \mathbf{E}^n_{P^*}$:

if θ and θ_0 both fall in the same rectangle R_i *then*

$$f_{\max}(\theta_0, \theta) < \epsilon/3 \text{ and } f_{\min}(\theta_0, \theta) > -\epsilon/3 \quad (5.72)$$

It can be seen from (5.70) that for $n \geq 1$, for all $x^n \in \mathbf{E}_{P^*}$,

$$f_{\min}(\theta_0, \theta) \leq f_n(\theta, x^n, \theta_0) \leq f_{\max}(\theta_0, \theta) \quad (5.73)$$

(5.69) now follows by combining (5.72) and (5.73) and substituting $\hat{\theta}(x^n)$ for θ_0.

A similar but simpler argument shows that (5.68) holds. We omit the details.

Stage 2 By the strong law of large numbers [47], we have with P^*-probability 1 that for all $\theta_s \in \Gamma_s$, for all $\delta > 0$, there exists an n_0 such that $n \geq n_0 \Rightarrow | - n^{-1} \sum \ln P(x^n|\theta_s) - E_{P^*}[-\ln P(X|\theta_s)]| < \delta$. Since Γ_s contains only a finite number of elements, this implies that, for all $\epsilon > 0$, with P^*-probability 1 there exists an n_0 such that *for all* $\theta_s \in \Gamma_s$:

$$n \geq n_0 \quad \Rightarrow \quad | - \frac{1}{n} \ln P(x^n|\theta_s) - E_{P*}[-\ln P(X|\theta_s)]| < \frac{1}{3}\epsilon \quad (5.74)$$

In addition, we also have for all $x^n \in \mathbf{E}^*$:

$$\begin{aligned} E_{P^*}[-\ln P(X|\hat{\theta}_s(x^n))] &\geq E_{P^*}[-\ln P(X|\tilde{\theta}_s)] && (5.75) \\ -\frac{1}{n} \ln P(x^n|\hat{\theta}_s(x^n)) &\leq -\frac{1}{n} \ln P(x^n|\tilde{\theta}_s) && (5.76) \end{aligned}$$

By first applying (5.74) with $\theta_s := \hat{\theta}_s(x^n)$ and then (5.75) we find

$$-\frac{1}{n} \ln P(x^n|\hat{\theta}_s(x^n)) \geq E_{P^*}[-\ln P(X|\tilde{\theta}_s)] - \frac{1}{3}\epsilon. \quad (5.77)$$

Using (5.76) and then applying (5.74) with $\theta_s := \tilde{\theta}_s$ we find

$$-\frac{1}{n} \ln P(x^n|\hat{\theta}_s(x^n)) \leq E_{P^*}[-\ln P(X|\tilde{\theta}_s)] + \frac{1}{3}\epsilon. \quad (5.78)$$

Combining (5.68), (5.69), (5.77) and (5.78) we find that for all $\epsilon > 0$, there exists an n_0 such that with P^*-probability 1, for all $n \geq n_0$,

$$| - \frac{1}{n} \ln P(x^n|\hat{\theta}(x^n)) - E_{P^*}[-\ln P(X|\tilde{\theta})]| < \epsilon \quad (5.79)$$

which is equivalent to (5.66). This proves the lemma for the case that the four minima in (5.67) are all attained by single values in the parameter space.

If this is not the case, we proceed as follows: by compactness all minima exist; the only problem is that they may be attained for several values. This can be handled by defining $\widetilde{\Theta}$ as the set of all θ minimizing $E_{P^*}[-\ln P(X|\theta)]$ (analogously to the first line of (5.67)) and defining $\widetilde{\Theta}_s$, $\widehat{\Theta}$ and $\widehat{\Theta}_s$ similarly. By the same reasoning as in Stage 1 of the proof, we can now prove the following existentially quantified version of (5.68) and (5.69): 'Let n and $\epsilon > 0$ be given. We claim that if we pick the rectangle side width s small enough then there exist $\tilde{\theta}_s \in \widetilde{\Theta}_s$, $\tilde{\theta} \in \widetilde{\Theta}$, $\hat{\theta}_s(x^n) \in \widehat{\Theta}_s$ and $\hat{\theta}(x^n) \in \widehat{\Theta}$ such that (5.68) and (5.69) hold'. By the same reasoning as in Stage 2 of the proof, we can also prove a universally quantified version of (5.68) and (5.69): 'if n is large enough, then for all $\hat{\theta}_s(x^n) \in \widehat{\Theta}_s$ and all $\tilde{\theta}_s \in \widetilde{\Theta}_s$ equations (5.77) and (5.78) hold with P^*-probability 1.' Combining these new versions of (5.68), (5.69), (5.77) and (5.78) we can proceed as above to show that (5.79) holds. The lemma then follows. $\square$

Epilogue: Using Models in a Careful Way

In this Epilogue to Part I of the thesis we study whether we have resolved the problematic issues concerning MDL raised in the Introduction and in Chapters 1-3. Briefly, these were:

1. Can we use models that are partially wrong to give reasonable predictions of future data? If so, what can we do with them and what not? ('What does a model say about the world?' as we wrote on page 28).

2. What to do if we are asked to use our model to predict against various loss functions? (Chapter 2, Section 2.2)

3. When using a probabilistic model class, why should we use the Shannon-Fano code $L(D|\theta) = -\log P(D|\theta)$ to encode the data with the help of model? When using a non-probabilistic model class, why should we use the code for which $L(D|H) = \text{ER}(D|H) + K$ for all D (Chapter 2, Section 2.2)?

Can we use models that are partially wrong? From the point of view of the MDL philosophy, we choose a model class $\mathcal{M}$ because we think it will help us to capture some of the regularity inherent in the data - but we have no hope that it will capture all (except if $\mathcal{M}$ corresponds to the class of all computer programs written in some language- but then the inference process becomes noncomputable). In Chapter 1 and in the Introduction to Part I of this thesis, we argued that this is the situation we will usually be in: all our models will always, to some extent, be wrong; therefore, though the question *came up* in connection with the MDL philosophy, it should be relevant not only to MDL, but to all statistical inference procedures.

Once we accept the fact that our model H for data D will always be partially wrong, we are faced with the question of what can be reliably inferred from such a model and what not. In Chapter 5, we showed that we can always change our model classes in such a way that we can reliably estimate the prediction error over future data. We proved (in Section 5.3) that this will lead, with high probability, to accurate estimates of error over future data even if data are independently drawn according to a distribution that is completely different from our model. Hence, if we are willing to make the i.i.d. assumption, then as long as we measure the error our model makes when predicting future data using the same error function as the one that was used in inferring the model from the data, we can use the model H reliably: even though it is partially wrong, it will give a correct impression of how accurate it is in predicting future data. Note however, that this result was proved *only* under the i.i.d. assumption[2] – so we still have to assume something about 'the truth out there'. Hence, we have not resolved in general whether using an overly simple (that is, partially wrong) model can lead to 'disastrous' results, as was feared by the statistician mentioned in the Introduction (page 3). We have only shown that prediction errors may be accurately estimated under much wider assumptions than the classical assumption that one's model class contains the true model. The question remains of whether this is not too weak; whether we will not always be interested in estimating more aspects about the data than just their prediction error. The example below shows at the same time that sometimes 'reliable' predictions are enough, while 'unreliable' predictions can lead to very misleading results.

Example 5.24 [Classification] Recall that in concept learning the model class consists of functions $H : \mathbf{E}_x \rightarrow \{0, 1\}$. Frequently the goal will be to use the concept $\ddot{H}$ learned on the basis of data D to *classify* new data: one is given a value $x \in \mathbf{E}_x$ and one has to predict the corresponding $y \in \{0, 1\}$. Suppose one uses a class of concepts $\mathcal{M}$ entropified with the 0/1-error for this. Suppose further that, for given data D, the estimate inferred is $(\ddot{H}, \ddot{\theta})$. In Example 5.3 on page 89 it was shown that this estimate says that 'if the model $\ddot{H}$ is used, then the probability of making a wrong prediction is $\ddot{\theta}$. Let us assume that data are i.i.d. according to a model P^* (Definition 5.13) so that Theorem 5.17 applies; this theorem tells us that if D is large enough and one uses a reasonable estimation procedure (Definition 4.3), then $\ddot{\theta} \approx P^*\{\ddot{H}(X) \neq Y\}$ (see Example 5.3). This means that if we are only interested in *classifying* future data, the component $\ddot{\theta}$ of our model $(\ddot{H}, \ddot{\theta})$ will give us a good idea on just how well we can do that. Hence if we use our model only for classification, we will have neither an overly optimistic nor an overly pessimistic idea of how good our model is at this task.

Now let us consider a specific example where the model $(\ddot{H}, \ddot{\theta})$ as issued by our estimation procedure on the basis of a large data set $D = (x^n, y^n)$ has $\ddot{\theta} = .95$. By Theorem 5.17 this means that, if D is really large, we can predict future data with 95% accuracy. However, it may be the case that, for the x_i where $\ddot{H}(x_i) = 1$, y_i is always equal to 1 while for the cases where $\ddot{H}(x_i) = 0$, $y_i \neq \ddot{H}(x_i)$ half the time. If $\ddot{H}(x_i) = 1$ in 90% of the cases, then we will have $\ddot{\theta} \approx .95$ while, if a new value x is given such that $\ddot{H}(x) = 0$ and we use $\ddot{H}(x)$ for prediction, we will only be right in about 50% of the cases. Hence our model is very bad for new data with $\ddot{H}(x) = 0$, and if a loss function is used such that predicting a 'false zero' leads to a much higher loss than when predicting a 'false one', then our model will really be quite worthless. Prediction against such a loss function is not reliable - and indeed, if we stick to 'safe statistics', we are not allowed to make such a prediction. We used an extreme example, but similar examples, where there exists a very simple rule that gives accurate predictions for a large subset of the x_i while being quite bad on the remaining x_i do occur in practice.

[2]One might conjecture though that the results can in some cases be extended to non-i.i.d. sources. For example, in the case of stationary ergodic non-i.i.d. sources the law of large numbers, which was the key to Theorems 5.16-5.18 remains applicable.

The 'naive Bayes' model class that we will use in the next chapter sometimes yields exactly such models based on real-world data sets.

This automatically brings us to the next question:

Predicting against arbitrary loss functions If we use an entropified model class $\langle\mathcal{M}\rangle_{\text{ER}}$, and we want to use our estimate $(\ddot{H},\ddot{\beta})$ to make predictions or decisions against loss functions LOSS that can*not* be written as a linear combination of ER we see that, at least if the sample space $\mathbf{E}$ is discrete, this will often still work – but it will also often be *unreliable*. That is, because the model class $\langle\mathcal{M}\rangle_{\text{ER}}$ restricted to models with fixed H is essentially a maximum entropy model class, one can apply the concentration phenomenon (Chapter 3, Section 3.5). In this case, it tells us that for an exponentially large majority of those data sets to which $(\ddot{H},\ddot{\beta})$ gives a good fit, the frequencies $(\gamma_1,\dots,\gamma_k)$ will be approximately equal to the probabilities $P(1|\ddot{H},\ddot{\beta}),\dots,P(k|\ddot{H},\ddot{\beta})$. If future data indeed belongs to this majority, then the average of every function (hence also every loss function) over future data will be approximately equal to its expectation over $(\ddot{H},\ddot{\beta})$, and the predictions will be accurate. Only in the few cases where the frequencies and the probabilities do not coincide will the predictions not be accurate - nevertheless, as we saw in the example above, these cases may certainly occur.

The Universal Yardstick This leaves us with the question of justifying the use of the Shannon-Fano code and how to associate codes with non-probabilistic models. In Section 5.4.2 we showed that, when using entropified model classes, the Shannon-Fano code can be justified in terms of minimizing worst-case expected code length *for probabilistic and non-probabilistic model classes alike.* This makes 'entropification' a quite general means of turning model classes into codes. As such, it is in line with the general MDL philosophy, in which all models are viewed as 'probabilistic' (Chapter 2.3, Section 2.2), or more properly, as *codes.* Let us consider a quote by Rissanen ([128], page 20):

> "... we then see that the unification obtained by interpreting all models as probabilistic has given us an immutable yardstick, the code length, which we never can reduce to zero by scaling or other devices. The same cannot be said about the usually suggested prediction error measures, which we easily can scale to any size, and which therefore will never be able to serve as a universal yardstick for model selection."

We agree that it is desirable and probably even possible to compare all models and model classes for given data D in terms of the code length they assign to D. But we also think that Rissanen's view leaves open two questions: first, how to base predictions and decisions on a 'probabilistic' model - since the main interpretation of the model is a code rather than a probability distribution according to which data are distributed, it is not a priori clear how this should be done. The second question is how to change model classes that are normally viewed as being 'non-probabilistic' into associated probabilistic model classes in a principled way. As we see it, the concept of 'reliable estimation' is a step towards answering the first question, while 'entropification' is a step towards answering the second question.

Part II

Experiments with MDL

Introduction to Part II

There are several theorems which say that, under reasonable conditions, using MDL leads to good or even optimal results [128, 14]. But such theorems are not sufficient to guarantee that it will work in the real world! First of all, while the mathematical conditions under which our optimality theorems hold may be reasonable, in practice they will never be *completely* met (is data ever generated by independent sampling from a fixed probability distribution?) Furthermore, when applying a theoretically optimal procedure in a practical setting, one always encounters additional problems. For example, parts of the data are missing; or data has to be discretized and it is not clear how to do this in an optimal manner; or the data turns out to have outliers of a completely unexpected kind; or the model class used is structured in a way that makes it computationally too hard to find the optimal model etc. In practice one always has to compromise, and it is not at all clear whether the theoretically optimal procedures will work well in practice.

Experiments with an Overly Simple Model

If one bases practical inductive inference on model classes that are *sufficiently simple*, then at least one of the problems mentioned above disappears: it becomes possible to efficiently compute, for arbitrary D, the single optimal model for D. For most sophisticated model classes this is not possible (or at least, no algorithm is known for doing this), and one has to search for a good model using heuristic search strategies that do not necessarily converge to the global optimum. Examples of such more sophisticated model classes are feed-forward neural networks with hidden layers, finite mixture distributions in statistics, hidden Markov models etc.

In the following two chapters, we base predictions on the so-called *Naive Bayes* model class which is simple enough so that we can efficiently find the optimal model. This class has been repeatedly shown to lead to results competitive with (sometimes even better than) those obtained by using much more sophisticated model classes [50, 90]. Yet it is evidently too simple ('naive') to provide realistic models for the domains of real-world data we will consider. We will use domains from areas as diverse as finance, forensic science, biology and medicine. We give a concrete example of the latter. The 'heart disease database' (see page 146) contains 270 observations. Each observation represents data concerning a single *case* (patient) and contains 14 *attribute entries*. Some of these attributes are age, sex, type of chest pain the patient suffers from (there are 4 possibilities), blood pressure, blood sugar level, 'maximum heart rate achieved' etc. The 14th attribute has a special status and is called the *class attribute*. It is either 1 (presence of heart disease) or 2 (absence of heart disease). All datasets we use have a similar decomposition in several attributes (either discrete or continuous-valued) one of which is a class attribute (always discrete-valued).

Throughout part II of this thesis, we use several methods to learn from a subset of the observations, the *training set* in order to *classify* new cases that are randomly picked from the *test set*. This is the subset of observations that are not contained in the training set. 'Classification' means that we are given all attribute values of a new case except for the class value. We then have to come up with a good prediction of the class value. In our example, we would be given age, sex etc. of a patient and we would be asked to predict whether or not that patient has heart disease.

When given enough training data from the heart disease database, the optimal Naive Bayes model class can predict whether or not a patient has a heart disease with about 84 % accuracy. (whereas, by simply predicting in accordance with the majority of cases would give us an accuracy of only 55 %). Similar accuracies are reached for the other data sets considered.

The key idea of the Naive Bayes model class is that all the attributes are considered *independent* of each other once the class value has been given. To give a concrete example, according to a Naive Bayes model, *given* the fact that a patient does not have a heart disease, the fact whether or not he has high blood pressure is independent of his age. According to a general practitioner we asked, this independence assumption is contrary to general medical wisdom. But note that we are only interested in predicting *whether or not a patient has heart disease*: since we are not asking about other attributes (like whether or not the patient has high blood pressure), we do not really care so much that the interrelations between other attributes are wrongly modeled.

The Power of Naive Bayes Of course, there may exist more sophisticated model classes, making less independence assumptions, that lead to better classifications of the heart disease data. But for these model classes, it is often not possible to efficiently compute the globally optimal model and/or prediction for the data at hand. We then have to settle for a suboptimal solution. It is perfectly possible that such a suboptimal solution is worse than the optimal Naive Bayes solution, which *can* be efficiently computed. The good overall behaviour of Naive Bayes suggests that this phenomenon frequently occurs in practice.

Comparing MDL to Bayesian Statistics and Maximum Likelihood

Specifically, we will use the Naive Bayes model class to compare the classification accuracies obtained by MDL to those obtained by several other methods of inductive inference. In Chapter 6, we compare MDL to some Bayesian methods and to the Maximum Likelihood method of classical statistics. This also involves some theoretical work: we have to instantiate the stochastic complexity to the Naive Bayes model class, which involves computing *Jeffrey's prior* for this model class. To make our results more widely applicable, we instantiate stochastic complexity to the class of Bayesian networks with arbitrary but fixed structure. This widely used model class [116, 137, 92] contains the Naive Bayes class as a special case.

Comparing MDL to MML

The Minimum *Message* Length Principle [161, 162] is a method for inductive inference that is very close in spirit, yet different from, the Minimum *Description* Length Principle. When investigating the relations between MDL and MML, we found a significant oversight in the derivation of MML estimators as given in [162]. We give a theoretical analysis of MML estimators and present a revised version of them. The original MML estimators differ from MDL estimators in an essential manner; but the revised MML estimators turn out to be closely related to Rissanen's 1996 refinement of the MDL Principle. Since all our theoretical results are asymptotic, it is not clear whether they lead to any significant differences in a practical setting. We therefore tested the original MML estimators, our revised MML estimators and an MDL-based prediction method on real world data, using once more the Naive Bayes model class. Our practical results turn out to be in agreement with our theoretical ones: MDL and our revised MML estimators perform slightly better than the original MML estimators.

Acknowledgements

The work reported in Part II of this thesis has been carried out jointly with the CoSCo group of the University of Helsinki, in particular P. Kontkanen, P. Myllymäki, T. Silander and H. Tirri. Two of the domains used in our experiments (the breast cancer and the lymphography domains; page 146) were obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Slovenia. Thanks go to M. Zwitter and M. Soklič for providing the data.

Chapter 6

Predictive Distributions for Bayesian Networks

6.1 Introduction

This chapter is about discrete prediction problems where the task is to select one outcome from a finite set of possible alternatives. This is done by determining a *predictive distribution* over all the possible outcomes. A predictive distribution is simply a probability distribution over the sample space, conditioned on the *training data* D. The purpose of this chapter is to compare different alternatives for computing such a predictive distribution. We will consider two Bayesian and two MDL-based methods to arrive at a predictive distribution. All four methods satisfy some optimality criterion, and all four of them are, under reasonable conditions, guaranteed to work well for very large training samples. The question is then how well they work for small sample sizes – the theoretical results about the four distributions considered do not say too much about this situation.

More specifically, our predictive distributions will be based on a class of *candidate models* $\mathcal{M}$, which throughout this chapter we assume to be a finitely parameterizable class of probabilistic models over a finite sample space. Given some sample data (the *training data*), and an incomplete *query vector*, where the values of some of the problem domain variables are not given, the task is to compute the predictive distribution for the missing part of the query vector.

In the Bayesian *maximum a posteriori (MAP)* approach, the predictive distribution is determined by using the model in $\mathcal{M}$ with the highest posterior probability, given the training data and a prior distribution for the parameters. In the Bayesian *evidence* approach, the predictive distribution is obtained by averaging over all the distributions representable by the chosen model form. In the third and fourth approaches considered here, we define the predictive distribution by using Rissanen's recent [129] definition of stochastic complexity based on minimizing worst-case regret (Chapter 2, Definition 2.8). In the third approach, a form of the stochastic complexity is used that cannot be interpreted as a probabilistic model (Chapter 1, Definition 1.5). In the fourth approach, a probabilistic-model version of stochastic complexity is used. This form happens to coincide with the Bayesian evidence approach with the prior instantiated to Jeffreys' prior.

6.1.1 Overview of this chapter

In the theoretical part of this chapter (sections 6.2-6.3), we derive explicit formulas for the four distributions considered so as to make them applicable in a practical setting. In the practical part of the chapter (sections 6.5-6.6, we compare the performance of the distributions on several publicly available data sets coming from 'real-world' problem domains.

The details of the discrete prediction problem considered here, together with the MAP, the evidence and the stochastic complexity predictive distributions for solving this problem are described formally in Section 6.2. In Section 6.3, we introduce the *Bayesian network model class* [116, 92]. In Section 6.4, we instantiate the general results of Section 6.2 to this class. Specifically, we show how to define each of the above mentioned predictive distributions for a given Bayesian network structure and we show how to compute Jeffreys' prior for such structures.

We then (Section 6.5) use these formulas in experiments on real-world data sets. To avoid computational difficulties, in the experiments we restrict ourselves to simple *classification* problems, using the 'Naive Bayes' model class [42], which is a special case of a Bayesian network model class.

The experiments indicate that the evidence approaches produce the most accurate predictions in the log-score sense, outperforming both the MAP method and the version of stochastic complexity that cannot be seen as a probabilistic model. The evidence approaches are also quite robust in the sense that they predict surprisingly well even when only a small fraction of the full training set is used. We analyze the reasons for this behaviour in Section 6.5.3.

Our Contributions To enhance readability of this chapter, we have not kept a strict separation between results that were already known and our own, original contributions. We therefore indicate briefly what our contributions are, apart from the very idea of applying stochastic complexity to Bayesian networks: on the theoretical side, there are two contributions: first, we show how the stochastic complexity can be used to arrive at a predictive distribution in a direct manner (sections 6.2.3 and 6.4.2). Second, we show how to compute Jeffreys' prior for Bayesian networks (Section 6.4.4). All the empirical work reported in this chapter (sections 6.5 and 6.6) is completely ours.

6.2 Predictive Distributions for Discrete Domains

6.2.1 The Prediction and Classification Problems

Let $\mathbf{E} = \mathbf{E}_1 \times \ldots \times \mathbf{E}_m$ be a sample space that can be decomposed into m subspaces. Each $\mathbf{E}_i$ contains a finite number k_i of elements: $\mathbf{E}_i = \{e_{i1}, \ldots, e_{ik_i}\}$. To emphasize the distinction between vectors and their components we denote throughout this chapter an outcome of $\mathbf{E}$ by $\vec{x}$. An outcome of $\mathbf{E}_i$ is denoted by x_i. Hence $\vec{x} = (x_1, \ldots, x_m)^T$. Our training data will consist of a sample D of n outcomes: $D = (\vec{x}_1, \ldots, \vec{x}_n)$. Here $\vec{x}_i = (x_{i1}, \ldots, x_{im})$.

In the simplest setting, we want to use the training data D to arrive at a good prediction of a single new observation $\vec{x} \in \mathbf{E}$. We will call this observation the *test vector*. We do all our predictions on the basis of a *predictive distribution* $\mathcal{P}(\cdot|D)$ where $\mathcal{P}(\vec{X} = \vec{x}|D)$ is to be read as 'the probability that the test vector is $\vec{x}$ given that the training data was D'.

In most realistic settings some of the constituent variables of the test vector are given to us and are to be used for prediction of the remaining variables. We call the variables of the test vector that are given *clamped* and the variables that are to be predicted *free*. Specifically, let $\vec{x} = (x_1, \ldots, x_m)$ be the test vector. Without loss of generality we may assume that the vector $\vec{u}$ of clamped variables coincides with $(x_1, \ldots, x_u)$ for some $u < m$ while the vector $\vec{v}$ of free variables coincides with $(x_{u+1}, \ldots, x_m)$. The *classification problem* is a special case of our prediction problem. In the classification problem we are given a vector of 'feature variables' $(x_1, \ldots, x_{m'})$ which has to be classified into one of K categories or *classes*. We can represent this by setting $m = m' + 1$ and letting $\mathbf{E}_m = \{e_{m1}, \ldots, e_{mK}\}$ where each e_{mi} stands for a class. The 'clamped' variables are then $(x_1, \ldots, x_{m'})$ and the free variable, also called the *class variable*, is x_m.

We can now state our aim more precisely as follows: we wish to compute, for all possible $\vec{v}$, the probabilities

$$\mathcal{P}(\vec{v}|\vec{u}, D) = \frac{\mathcal{P}(\vec{v}, \vec{u}|D)}{\mathcal{P}(\vec{u}|D)} = \frac{\mathcal{P}((\vec{v}, \vec{u})|D)}{\sum_{\vec{v}} \mathcal{P}((\vec{v}, \vec{u})|D)}, \tag{6.1}$$

where the summation is over all $\vec{v} \in \mathbf{E}_{u+1} \times \ldots \times \mathbf{E}_m$.

Consequently, we see that the conditional distribution for the free variables can be computed using the complete data vector predictive probabilities $\mathcal{P}(\vec{x}|D)$ for each of the possible complete vectors $\vec{x} = (\vec{v}, \vec{u})$. It should be noted that the number of terms in the summation over possible $\vec{v}$ grows exponentially with the number of free variables. Therefore, predicting the values of many variables given the values of only a few may be difficult. However, in many cases of practical interest we only want to predict the values of a very small number (in classification problems just one) of the variables and then such difficulties do not arise.

We model our data using a model class $\mathcal{M}$ that is finitely parameterized by some $\Gamma \subset \mathbf{R}^k$ (Definition 2.7 of page 35). Hence $\mathcal{M}$ can be written as $\mathcal{M} = \{P(\cdot|\theta) \mid \theta \in \Gamma\}$. We further assume that all models in $\mathcal{M}$ render the data i.i.d. Hence for every $\theta \in \Gamma$, $P(\vec{x}^n|\theta) = \Pi_{i=1}^n P(\vec{x}_i|\theta)$.

In the next section we consider four different ways to compute predictive distributions $\mathcal{P}(\cdot|D)$ based on the model class $\mathcal{M}$.

6.2.2 The Bayesian Predictive Distributions $\mathcal{P}_{map}$ and $\mathcal{P}_{av}$

Given a prior density $P(\theta)$ defined for all $\theta \in \Gamma$, we can arrive at a posterior distribution $P(\theta|D)$ by using Bayes' rule:

$$P(\theta|D) \propto P(D|\theta)P(\theta). \tag{6.2}$$

In the *maximum a posteriori (MAP) probability* approach, the predictive distribution $\mathcal{P}$ is identified with the distribution corresponding to the single model $\breve{\theta}(D)$ maximizing the posterior distribution (6.2),

$$\breve{\theta}(D) = \arg\max_{\theta \in \Gamma} P(\theta|D).$$

The corresponding predictive distribution is

$$\mathcal{P}_{map}(\vec{x} \mid D) := P(\vec{x} \mid \breve{\theta}(D)). \tag{6.3}$$

If we assume the prior distribution $P(\theta)$ to be uniform, then (6.2) becomes $P(\theta|D) \propto P(D|\theta)$ and the MAP model becomes equal to the *Maximum Likelihood (ML) model*; see Chapter 1, page 18.

We can get more sophisticated predictions by averaging (integrating) over all the models in $\mathcal{M}$ instead of using a single model $\breve{\theta}$. The resulting distribution is called the *marginal* or *evidence* distribution in the Bayesian literature; we denote it by P_{av}. For $\vec{x}^n \in \mathbf{E}^n$, $P_{av}(\vec{x}^n)$ is given by:

$$P_{av}(\vec{x}^n) = \int P(\vec{x}^n|\theta)P(\theta)d\theta. \tag{6.4}$$

where the integration goes over all the models θ in $\mathcal{M}$. The resulting predictive distribution then coincides with the evidence distribution conditioned on training data D:

$$\mathcal{P}_{av}(\vec{x} \mid D) := P_{av}(\vec{x} \mid D) = \frac{P_{av}(\vec{x}, D)}{P_{av}(D)} \tag{6.5}$$

For more details about the Bayesian MAP and evidence approaches we refer to Chapter 2, Section 2.8.

6.2.3 The 'direct' Stochastic Complexity Predictive Distribution $\mathcal{P}_{sc}$

In Chapter 2 we introduced the stochastic complexity distribution $P_{sc}(\cdot) = P_{sc}(\cdot|\mathcal{M})$ as the distribution that minimizes the worst case excess code length, or equivalently, the *regret* (see equations 2.5-2.7 of Section 2.3). In Section 2.7 we further showed that sequential proportional betting on the basis of P_{sc} minimizes the worst-case excess loss in capital. Here the 'excess' is over the loss we would have made if we had bet on the basis of the model in the class that, with hindsight, would have given us the least loss of capital; see page 43. This property of P_{sc} provides a strong reason for using it in actual prediction tasks.

For the model classes $\mathcal{M}$ we will consider in our experiments, a unique ML estimator $\hat{\theta}(D)$ exists for all $D \in \mathbf{E}^*$. In addition $\mathbf{E}$ is finite, and this means (Chapter 2, Equation (2.10), page 37) that the stochastic complexity distribution can be explicitly written as follows:

$$P_{sc}(D) = \frac{P(D|\hat{\theta}(D))}{\sum_{D \in \mathbf{E}^n} P(D|\hat{\theta}(D))} = (F(n)^{-1})P(D|\hat{\theta}(D)) \tag{6.6}$$

Here $F(n)$ is defined as $F(n) = \sum_{D \in \mathbf{E}^n} P(D|\hat{\theta}(D))$. At first sight, using (6.6) as the predictive distribution may seem infeasible in practice since computing the normalizing sum $F(n)$ involves summing over a number of terms exponential in n. The problem disappears if one computes a predictive distribution for a data set D of length n in the following straightforward manner:

$$\begin{aligned} P_{sc}(\vec{x} \mid D) &= \frac{P_{sc}(\vec{x}, D)}{\sum_{\vec{x}'} P_{sc}(\vec{x}', D)} \\ &= \frac{P(\vec{x}, D \mid \hat{\theta}(\vec{x}, D)) \cdot (F(n+1))^{-1}}{\sum_{\vec{x}'} P(\vec{x}', D \mid \hat{\theta}(\vec{x}', D)) \cdot (F(n+1))^{-1}} \\ &= \frac{P(\vec{x}, D \mid \hat{\theta}(\vec{x}, D))}{\sum_{\vec{x}'} P(\vec{x}', D \mid \hat{\theta}(\vec{x}', D))} \\ &\overset{\text{i.i.d.}}{=} \frac{P(\vec{x} \mid \hat{\theta}(\vec{x}, D)) P(D \mid \hat{\theta}(\vec{x}, D))}{\sum_{\vec{x}'} P(\vec{x}' \mid \hat{\theta}(\vec{x}', D)) P(D \mid \hat{\theta}(\vec{x}', D))}. \end{aligned} \tag{6.7}$$

The *stochastic complexity predictive distribution* $\mathcal{P}_{sc}$ can now be identified with the conditional stochastic complexity distribution $P_{sc}(\vec{x} \mid D)$ as given in Equation (6.7):

$$\mathcal{P}_{sc}(\vec{x}|D) := P_{sc}(\vec{x}|D). \tag{6.8}$$

6.2.4 Connecting P_{av} and P_{sc}: the $\mathcal{P}_{jef}$ Predictive Distribution

P_{sc} is not a probabilistic model

Though $P_{sc}(\cdot)$ is defined with respect to a class of probabilistic models $\mathcal{M}$ (Chapter 1, Definition 1.5), it is not *itself* a probabilistic model: in general,

$$\sum_{z \in \mathbf{E}} P_{sc}(D, z) \neq P_{sc}(D).$$

Let us contrast this with a P that *does* conform to the definition of probabilistic model: let P be a probabilistic model and let $n_1 < n_2$. Let P_{n_i} be the restriction of P to sequences of length n_i. Both P_{n_1} and P_{n_2} are probability distributions (over $\mathbf{E}^{n_1}$ and $\mathbf{E}^{n_2}$ respectively). The fundamental property of a probabilistic model is that, for all $D \in \mathbf{E}^{n_1}$, $P_{n_1}(D) = \sum_{D' \in \mathbf{E}^{n_2 - n_1}} P_{n_2}(D, D') = P_{n_2}(D)$: P_{n_2} can be seen as an extension of P_{n_1}.

SC has been defined so as to minimize the worst-case regret. The distribution which achieves this for us for sample size n_2 can*not* be interpreted as an extension of the distribution which achieves it for sample size n_1. Therefore, the stochastic complexity distributions defined for data sequences of length 1, of length 2 etc. are really all different: perhaps a better notation would be to write P_{sc}^n for the stochastic complexity distribution over sequences of length n.

Let us now once more consider the game we introduced in Chapter 2, Section 2.7, page 43: we are given a model class $\mathcal{M}$ and we sequentially predict x_{i+1} on the basis of x^i. We saw on page 43 that sequentially predicting on the basis of P_{sc} minimizes the worst-case regret. However, the P_{sc} we used there was really the stochastic complexity distribution as defined *for data of length n*. So we really predicted x_1 on the basis of $P_{sc}^n(x_1)$, x_2 on the basis of $P_{sc}^n(x_2|x_1)$ etc. In practical settings, the total number n of predictions we want to make is often unknown. In that case, we do not know whether to use P_{sc}^1, P_{sc}^2 or any other P_{sc}^i for prediction. Using a different n for different predictions will certainly not make our predictions optimal! But this is what will happen if we use $\mathcal{P}_{sc}$ as defined in Equation 6.8 for prediction. To see this, note that in Section 6.2.3 we implicitly used P_{sc}^{n+1} to define the predictive distribution (6.8): $\mathcal{P}_{sc}(\vec{x}|D)$ was defined in terms of $P_{sc}(D, \vec{x})$ and the sample $(D, \vec{x})$ is of length $n+1$. So if we would sequentially predict on the basis of (6.8), this would amount to predicting x_1 on the basis of P_{sc}^1, x_2 on the basis of $P_{sc}^2(\cdot|x_1)$, x_3 on the basis of $P_{sc}^3(\cdot|x_1, x_2)$ etc., which is not optimal in general.

Things are not necessarily as bad as they seem, however. Rissanen [129] proved that under suitable regularity conditions on the class of models $\mathcal{M}$, as n grows larger, P_{sc}^n starts to behave more and more like a probabilistic model - the restriction of distribution P_{sc}^{n+1} to outcomes of length n will be almost identical to the distribution P_{sc}^n. At least for the class of Bernoulli models, this 'convergence' to a probabilistic model seems to take place very fast [129]. For this reason we decided to use the predictive distribution P_{sc} in our experiments after all.

Nevertheless, we could ask ourselves whether there exists a probabilistic model that achieves a worst-case regret that is almost as small as that of the stochastic complexity. We present such a model below.

A Probabilistic-Model Version of P_{sc}

The evidence distribution P_{av} can be seen as a probabilistic model irrespective of the prior w used, since for all $D \in \mathbf{E}^*$ we have

$$\sum_{z \in \mathbf{E}} \int P(D, z|\theta) w(\theta) d\theta = \int P(D|\theta) w(\theta) d\theta.$$

Under certain regularity conditions on the class of models $\mathcal{M}$, one can prove that the prior over the models in $\mathcal{M}$ which yields the lowest regret $\mathcal{R}_{L_{av,w}}(D) = -\log P_{av}(D) - [-\log P(D|\hat{\theta}(D))]$ is the so-called *Jeffreys' prior* $\pi(\theta)$ [129] which we define below. We will denote the marginal distribution P_{av} with the prior instantiated to Jeffreys' prior by $P_{jef} = P_{jef}(\cdot|\mathcal{M})$.

The specific form of Jeffreys' prior depends on the model class $\mathcal{M}$ that is used. For the model classes we will consider in our experiments, Jeffreys' prior is proper[1]. In that case it is given by (see for example [129] or [17]):

$$\pi(\theta) = \frac{|I(\theta)|^{1/2}}{\int |I(\eta)|^{1/2} d\eta}. \tag{6.9}$$

Here $|I(\theta)|$ is the determinant of the *Fisher information matrix* $I(\theta)$. Denoting $\theta = (\theta_1, \ldots, \theta_k)$, entry (i, j) of matrix $I(\theta)$ is defined as

$$[I(\theta)]_{i,j} = -E_\theta \left[\frac{\partial^2 \log P(\vec{X}|\theta)}{\partial\theta_i \partial\theta_j} \right].$$

Originally, Jeffreys' prior was derived by invariance arguments [78, 17]: the value of $\pi(\theta)$ is invariant under 1-1 transformations of the parameter space. We will show below that if the model class $\mathcal{M}$ is an *exponential family* (Chapter 3, Section 3.4), then P_{jef} essentially minimizes the worst-case regret. In the next section we introduce the model class of Bayesian networks which we will use in our experiments. We show there that this class is an exponential family. This means that we can use the predictive distribution based on P_{jef} as an alternative version of the stochastic complexity predictive distribution P_{sc}. In Section 6.4, we derive an analytic expression for Jeffreys' prior $\pi(\theta)$ for the case where θ indexes a Bayesian network, and show how to calculate P_{jef} for the class of Bayesian networks with a fixed but arbitrary structure.

To show that P_{jef} minimizes regret we need the following fundamental result (a special case of a theorem to be found in [26]).

Theorem 6.1 (Clarke & Barron) *Let $\mathcal{M}$ be an exponential family of irreducible dimension k (as defined in Chapter 3, Section 3.4) over sample space $\mathbf{E}$. Let $\mathcal{M}$ be parameterized by some $\Gamma \subset \mathbf{R}^l$ such that (1) there exists a bijection $g : \Gamma \to \mathcal{M}$ and (2) $\ln P(x|\theta)$ as a function of θ is differentiable infinitely often for all $x \in \mathbf{E}$ at all $\theta \in \Gamma$. Let $w(\theta)$ be a proper prior density over Γ with $w(\theta) > 0$ for all $\theta \in \Gamma$. Let $x^n \in \mathbf{E}^n$ be generated by repeated sampling from $P(\cdot|\theta^*)$ where θ^* is an arbitrary member of Γ. Then, with θ^*- probability 1,*

$$-\ln P_{av}(x^n) = -\ln P(x^n|\hat{\theta}(x^n)) + \frac{k}{2} \ln \frac{n}{2\pi} + \ln \frac{\sqrt{|I(\theta^*)|}}{w(\theta^*)} + o(1). \tag{6.10}$$

where P_{av} is the evidence distribution (6.4) *with prior w.*

If we substitute Jeffreys' prior $\pi(\theta)$ (Equation 6.9) for the prior $w(\theta)$, then the term $\ln(\sqrt{|I(\theta^*)|}/w(\theta^*))$ in Equation 6.10 becomes independent of the data generating model θ^*. This means that whatever this 'true' θ^* is, we obtain, with probability one, the same asymptotic expansion

$$-\ln P_{jef}(x^n) = -\ln P(x^n|\hat{\theta}(x^n)) + \frac{k}{2} \ln \frac{n}{2\pi} + \ln \int \sqrt{|I(\theta)|} d\theta + o(1). \tag{6.11}$$

[1] A prior w is *proper* if $\int w(\theta) d\theta = 1$.

Let w be a proper prior such that within the parameter space there exists a region R of non-zero volume with $\int_R w(\theta)d\theta \neq \int_R \pi(\theta)d\theta$. Let P_{av} be the marginal distribution with such a prior w. Since both w and π are proper the parameter space must also contain a region R' of non-zero volume such that $w(\theta) < \pi(\theta)$ for all $\theta \in R'$. By Theorem 6.1 above it follows that for every $\theta^* \in \text{int}(R')$, it holds with θ^*-probability 1 that $-\ln P_{av}(x^n) = -\ln P_{jef}(x^n) + K + o(1)$ for some $K > 0$. Hence for all P_{av} based on proper priors w as defined above, there exists θ^* for which they perform worse than π, while π itself reaches (with probability 1) the same asymptotic expansion (6.11) for all $\theta^* \in \Gamma$. It follows that, among all proper priors, Jeffreys' prior minimizes the worst-case regret that is reached with probability 1. Here 'worst-case' is with respect to generating distributions instead of data sequences.

This provides a justification for basing predictions on P_{jef} instead of the stochastic complexity whenever the model class is an exponential family: First, P_{jef} is a probabilistic model (so we can use it without knowing the number of predictions to be made). Second, within the class of models that can be written as a Bayesian marginal distribution $P_{av}(\cdot|\mathcal{M})$, it is the one that minimizes worst-case regret (where worst-case regret is defined with respect to a worst-case generating distribution instead of a worst-case data sequence). Third, all such marginal distributions with priors $w(\theta) > 0$ for all $\theta \in \Gamma$ are asymptotically good approximations of the stochastic complexity; see [128].

Formally, we define the *predictive distribution based on P_{jef}* by equating it to P_{jef} conditioned on D:

$$\mathcal{P}_{jef}(\vec{x}|D) := P_{jef}(\vec{x}|D) \tag{6.12}$$

Discussion The line of reasoning just presented suffers from two weaknesses: first, Equation 6.10 has only been proven to hold (with probability 1) under the assumption that there is some true model in $\mathcal{M}$ generating the data. Second, it is not yet clear how close $-\ln P_{jef}(x^n)$ really is to $-\ln P_{sc}(x^n)$, or whether there does not exist a different probabilistic model that can*not* be written as a Bayesian marginal distribution, but that nevertheless achieves a smaller regret.

Both weaknesses can most likely be removed: *very* recently (a week before this thesis had to go to the printer's), Takeuchi and Barron [149] announced that they have proven theorems stating that for all exponential families the following holds: First, (6.11) is the case not just with probability 1 but, independent of any underlying distribution, *uniformly* for all $x^n \in \mathbf{E}^n$ which are such that $\hat{\theta}(x^n)$ lies in the interior of the parameterspace. Second, $-\ln P_{sc}(x^n) = -\ln P_{jef}(x^n) + o(1)$ also uniformly for all $x^n \in \mathbf{E}^n$ which are such that $\hat{\theta}(x^n)$ lies in the interior of the parameter space. Therefore, using P_{jef} should lead to approximately the same (and hence by definition minimal in the worst-case) regret as P_{sc}. This would resolve both weaknesses. No proofs of these theorems have yet been published however.

6.3 Bayesian Networks

A Bayesian (belief) network [116, 79, 92] is a graphical high-level representation of a probability distribution over a finite set of discrete random variables $X_1, \ldots, X_m$. Bayesian networks have their roots in the notion of 'conditional independence' in statistics [35] and the method of 'path analysis' in the social sciences [165]. Nowadays, they are used extensively in the field of 'reasoning under uncertainty' [116, 114]. Before defining Bayesian networks themselves we need to say something about the sample space over which they are defined.

The Sample Space The variables $X_1, \ldots, X_m$ take on values in $\mathbf{E}_1, \ldots, \mathbf{E}_m$ respectively. In our setting, we assume the $\mathbf{E}_i$ to be finite. Previously we denoted elements in $\mathbf{E}_i$ as follows: $\mathbf{E}_i = \{e_{i1}, \ldots, e_{ik_i}\}$. Whenever in the sequel we refer to an outcome x of one of the random variables X_1, X_2, ... or X_m, it will be clear from the context to which of these random variables x belongs. We can therefore denote outcome e_{ij} by the number j without introducing any ambiguity. When defining Bayesian networks this will allow for a considerable simplification of notation. Hence, from now on we write $\mathbf{E}_i = \{1, \ldots, k_i\}$.

6.3.1 Definition of Bayesian Networks

A Bayesian network consists of a *structure* $\mathcal{G}$ and a *parameter set* θ. We discuss both in turn.

The Bayesian Network Structure The Bayesian network structure is a directed acyclic graph (DAG), where the nodes correspond to the domain variables $X_1, \ldots, X_m$. An example Bayesian network structure is shown in Figure 6.1. Below we first introduce the concepts involved an a Bayesian network structure; we then give an example based on Figure 6.1.

The graph $\mathcal{G}$ is represented by a set of $m - 1$ *Parent variable sets* $Pa_i \subseteq \{X_{i+1}, \ldots, X_m\}$ where $1 \leq i < m$. For each node (corresponding to) X_i, the set Pa_i represents the nodes in the graph that are the parents of X_i. We define $\mathbf{E}_{\mathrm{pa}_i}$ to be the set of all possible configurations of the values of the parents Pa_i of variable X_i. Each $q_i \in Pa_i$ stands for a vector of values of random variables. We write $P(Pa_i = q_i)$ to denote the probability that the parent variables of X_i take on the values corresponding to configuration q_i. Hence $P(Pa_i = q_i)$ is an abbreviation of $P(X_{j_1} = x_{j_1}, \ldots, X_{j_{m'}} = x_{j_{m'}})$ where $X_{j_1}, \ldots, X_{j_{m'}}$ are the elements of Pa_i and $q_i = (x_{j_1}, \ldots, x_{j_{m'}})$.

From the definition of conditional probability one sees that each probability distri-

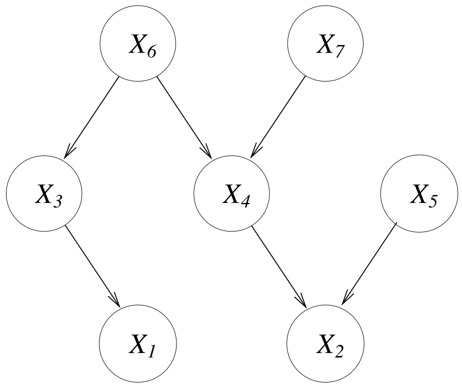


Figure 6.1: An example Bayesian network.

bution over $\mathbf{E} = \mathbf{E}_1 \times \ldots \times \mathbf{E}_m$ can be written such that, for all $\vec{x} \in \mathbf{E}$,

$$\begin{aligned}
P(\vec{x}) &= P(X_1 = x_1, \ldots, X_m = x_m) = \\
&P(X_1 = x_1 \mid X_2 = x_2, \ldots, X_m = x_m) \times P(X_2 = x_2 \mid X_3 = x_3, \ldots, X_m = x_m) \times \\
&\cdots \times P(X_m = x_m) = \\
&\prod_{i=1}^{m} P(X_i = x_i | X_{i+1} = x_{i+1}, \ldots, X_m = x_m) \quad (6.13)
\end{aligned}$$

The key idea behind Bayesian Networks is to use a given network structure $\mathcal{G}$ to define a set of *independence assumptions* such that, in the decomposition (6.13), the distribution of each X_i depends not on *all* of the variables $X_{i+1}, \ldots, X_m$, but only on the subset of variables given by Pa_i. Hence a structure $\mathcal{G}$ *restricts* the set of possible distributions over $\mathbf{E}$ to the set of distributions that can be written as follows:

$$\begin{aligned}
P(\vec{x}) &= P(X_1 = x_1, \ldots, X_m = x_m) = \\
&P(X_1 = x_1 \mid Pa_1 = q_1) \times P(X_2 = x_2 \mid Pa_2 = q_2) \times \\
&\cdots \times P(X_m = x_m) = \\
&\prod_{i=1}^{m} P(X_i = x_i | Pa_i = q_i), \quad (6.14)
\end{aligned}$$

where for notational convenience we write $P(X_m = x_m | Pa_m = q_m)$ instead of $P(X_m = x_m)$ (q_m can be thought of as always taking on a single value, independent of the outcome of any of the X_i).

Example 6.2 Consider Figure 6.1 again. Suppose that in this figure, $\mathbf{E}_4 = \{1, 2\}$ and $\mathbf{E}_5 = \{1, 2, 3\}$. Then $Pa_2 = \{X_4, X_5\}, \mathbf{E}_{\mathrm{pa}_2} = \{(1,1), (1,2), (1,3), (2,1), (2,2), (2,3)\}$. Let now $q_2 \in \mathbf{E}_{\mathrm{pa}_2}$; as an example, take $q_2 = (1, 2)$. Then $P(Pa_2 = q_2) = P(X_4 = 1, X_5 = 2)$.

Intuitively, the independence assumption encoded in a Bayesian network structure $\mathcal{G}$ is the following: given the values of its parents, a variable X_i becomes independent of all variables in the graph except its descendants. To get some (quite rough) idea of what this means, consider variables X_3 and X_4 in the figure. Let $j \in \{2,4,5,7\}$. The fact that the probability distribution over $X_1, \ldots, X_7$ can be written in the form (6.14) implies that , for all $x_3 \in \mathbf{E}_3, x_6 \in \mathbf{E}_6$ and all $x_j \in \mathbf{E}_j$, we have $P(X_3 = x_3 | X_6 = x_6, X_j = x_j) = P(X_3 = x_3 | X_6 = x_6)$: once the value of X_6 is known (given), the distribution of X_3 becomes independent of any of the other outcomes except its descendants (which, in this case, consist only of X_1). However, if the value of X_6 is not known, then the distribution of X_3 may very well depend on other outcomes. For example, it is easy to come up with a distribution of the form (6.14) for which $P(X_3 = x_3 | X_4 = 1) \neq P(X_3 = x_3 | X_4 = 2)$.

The Bayesian Network Parameter Set A Bayesian network structure $\mathcal{G} = \{Pa_1, \ldots, Pa_{m-1}\}$ represents the class of all probability distributions P on variables $X_1, \ldots, X_m$ such that $P(\vec{X} = \vec{x})$ can, for all $\vec{x}$, be written as in (6.14). Consequently, in the Bayesian network model class induced by a graph $\mathcal{G}$, a single distribution P is uniquely determined by fixing the values of the probabilities (6.14). We will therefore parameterize such a class by using one parameter for each of these probabilities. The parameter corresponding to probability $P(X_i = x_i | Pa_i = q_i)$ will be indicated by $\theta^i_{x_i|q_i}$. Hence for all $1 \leq i \leq m$, all $q_i \in \mathbf{E}_{\mathrm{pa}_i}$ and all $x_i \in \mathbf{E}_i$:

$$P(X_i = x_i | Pa_i = q_i) = \theta^i_{x_i|q_i} \tag{6.15}$$

The parameter vector consisting of the values $\theta^i_{x_i|q_i}$ for all $x_i \in \mathbf{E}_i$ and $q_i \in Pa_i$ will be denoted by θ^i (we assume the parameters $\theta^i_{x_i|q_i}$ to be ordered in some fixed manner). Summarizing:

Definition 6.3 (Bayesian Network) *A Bayesian network $(\theta, \mathcal{G})$ consists of a structure $\mathcal{G} = \{Pa_1, \ldots, Pa_{m-1}\}$ and a parameter vector $\theta = (\theta^1, \ldots, \theta^m)$. Here the θ^i are themselves vectors with components $\theta^i_{x_i|q_i}$ for all $x_i \in \mathbf{E}_i$ and $q_i \in Pa_i$. The Bayesian network probability distribution P_θ corresponding to Bayesian network $(\theta, \mathcal{G})$ is defined such that for all $\vec{x} \in \mathbf{E}$, all $1 \leq i \leq m$, all $q_i \in \mathbf{E}_{pa_i}$ and all $x_i \in \mathbf{E}_i$, Equations 6.14 and 6.15 hold.*

Definition 6.4 (Bayesian Network Model Class) *The class $\Gamma_\mathcal{G}$ of parameter values for Bayesian networks with structure $\mathcal{G}$ contains exactly those θ with, for all $1 \leq i \leq m$ and all $q_i \in \mathbf{E}_{pa_i}$:*

1. *for all $x_i \in \mathbf{E}_i$: $\theta^i_{x_i|q_i} > 0$*

2. $\sum_{x_i \in \mathbf{E}_i} \theta^i_{x_i|q_i} = 1$.

The class of Bayesian networks with structure $\mathcal{G}$, denoted by $\mathcal{M}_\mathcal{G}$, is given by $\mathcal{M}_\mathcal{G} = \{P_\theta | \theta \in \Gamma_\mathcal{G}\}$.

In the following we assume an arbitrary but fixed structure $\mathcal{G}$ and we consider the class of probability distributions $\mathcal{M}_{\mathcal{G}}$. Note that our definition of $\mathcal{M}_{\mathcal{G}}$ excludes points at the boundaries of the parameter space. Note further that, by (6.14) conditional distributions of variables given values for their parent values are multinomial: $X_i|q_i \sim \text{Multi}(1; \theta^i_{1|q_i}, \ldots, \theta^i_{k_i|q_i})$.

In the sequel, we will sometimes use the notation $P(\cdot|\theta)$ to denote $P_\theta(\cdot)$.

6.3.2 Parameter Priors for Bayesian Networks

The family of Dirichlet distributions is *conjugate* (see e.g. [39] or [17]) to the family of multinomials: when a Dirichlet density is used as a prior for the family of multinomials, then the functional form of the parameter distribution remains invariant in the prior-to-posterior transformation. It is therefore convenient to use prior distributions from this family. Moreover, as we will see, all theoretically motivated prior distributions for Bayesian network model classes that we consider in our experiments in this and the next chapter turn out to be Dirichlet. We now give the definition of the Dirichlet distributions in terms of their densities (adapted from [17]):

Definition 6.5 *Let $\theta_1, \ldots, \theta_k$ be such that $\sum_{i=1}^k \theta_i = 1$ and $0 \leq \theta_i \leq 1$ for all i. The* Dirichlet density *corresponding to* hyperparameters $(\mu_1, \ldots, \mu_k)$ *is given by*

$$P(\theta_1, \ldots, \theta_k|\mu_1, \ldots, \mu_k) = c \cdot \prod_{i=1}^{k} (\theta_i)^{(\mu_i - 1)} \tag{6.16}$$

where c is a constant needed for normalization. If $\theta_1, \ldots, \theta_k$ are distributed according to (6.16), we write

$$(\theta_1, \ldots, \theta_k) \sim \text{Di}(\mu_1, \ldots, \mu_k)$$

For our Bayesian network distributions, we will assume that for each $1 \leq i \leq m$, for each $q_i \in \mathbf{E}_{\text{pa}_i}$,

$$(\theta^i_{1|q_i}, \ldots, \theta^i_{k_i|q_i}) \sim \text{Di}(\mu^i_{1|q_i}, \ldots, \mu^i_{k_i|q_i})$$

for some $(\mu^i_{1|q_i}, \ldots, \mu^i_{k_i|q_i})$. We further assume that the parameter vectors $(\theta^i_{1|q_i}, \ldots, \theta^i_{k_i|q_i})$ are independent, so that the joint prior distribution of all the parameters θ becomes

$$P(\theta) \propto \prod_{i=1}^{m} \prod_{q_i \in \mathbf{E}_{\text{pa}_i}} \prod_{x_i \in \mathbf{E}_i} (\theta^i_{x_i|q_i})^{(\mu^i_{x_i|q_i} - 1)}, \tag{6.17}$$

which is itself a Dirichlet distribution.

Observe that the *uniform prior* is a Dirichlet density. It is obtained when the values of all hyperparameters $\mu^i_{x_i|q_i}$ are set to 1, since in that case all the exponents in 6.17 become 0.

6.3.3 Bayesian network families $\mathcal{M}_{\mathcal{G}}$ are exponential families

As explained in Section 6.2.4, for model classes that satisfy the regularity conditions of Theorem 6.1, the Bayesian evidence (6.4) with the prior instantiated to Jeffreys' prior can serve as an approximation of stochastic complexity. The following theorem demonstrates that Bayesian network model classes $\mathcal{M}_{\mathcal{G}}$ satisfy these conditions:

Proposition 6.6 *Let $\mathcal{M}_{\mathcal{G}}$ be the class of probability distributions corresponding to an arbitrary Bayesian network structure $\mathcal{G}$ and let $\Gamma_{\mathcal{G}}$ be the class of associated parameters. Then (1) the class $\mathcal{M}_{\mathcal{G}}$ is an exponential family of irreducible dimension k for some k, and (2) there exists a bijection $g : \Gamma_{\mathcal{G}} \to \mathcal{M}_{\mathcal{G}}$ such that $P(\vec{x}|\theta) = g(\theta)$ as a function of θ is differentiable infinitely often for all $\vec{x} \in \mathbf{E}$ at all $\theta \in \Gamma_{\mathcal{G}}$.*

Proof: To see that $\mathcal{M}_{\mathcal{G}}$ is an exponential family let $a = |\mathbf{E}|$, order the elements of $\mathbf{E}$ from 1 to a and, for $1 \leq i \leq a$, let $\vec{x}_i$ be the element in $\mathbf{E}$ corresponding to index i in this ordering. Now write:

$$P(\vec{X} = \vec{x}|\beta) = \frac{1}{Z(\beta)} \exp(-\sum_{i=1}^{a} \beta_i \mathcal{I}(\vec{x} = \vec{x}_i)) \tag{6.18}$$

where $\mathcal{I}$ is the indicator function (Chapter 3, page 51). Clearly, this distribution is of the form (3.11) required for the exponential family (page 56). One easily checks that each model $P \in \mathcal{M}_{\mathcal{G}}$ can be written in the form (6.18). One can show [82] that for each exponential family there exists some minimal k such that it can be written in the form (3.11) with $\beta = (\beta_1, \ldots, \beta_k)$. Hence $\mathcal{M}_{\mathcal{G}}$ must be irreducible for some k, which proves (1).

We now prove (2). The only part of (2) that is not immediate is the condition that g must map each parameter vector to a different model. We prove this by showing that for all parameter values $\theta, \theta' \in \Gamma$ with $\theta \neq \theta'$ we have $P(\cdot|\theta) \neq P(\cdot|\theta')$. By renaming random variable X_i to X_{m-i+1} and renaming all the elements in Pa_i correspondingly, we see that the following definition of a Bayesian network structure is equivalent to the one we gave in Section 6.3.1: a Bayesian network structure over a set of m discrete random variables $X_1, \ldots X_m$ is defined by a set of parent variable sets $\{Pa_1, \ldots, Pa_m\}$ where $Pa_i \subseteq \{X_1, \ldots, X_{i-1}\}$. This alternative way of ordering variables will be adopted in the proof.

Clearly, for $1 \leq l \leq m$, the set $\mathcal{G}_l = \{Pa_1, \ldots, Pa_l\}$ forms a Bayesian network structure for the set of variables $X_1, \ldots, X_l$. For any set of parameter values $\theta = (\theta^1, \ldots, \theta^m) \in \Gamma$ we let $\theta\rfloor_l$ stand for the restriction of θ to $(\theta^1, \ldots, \theta^l)$

Let Γ_l be the set of parameters corresponding to $\mathcal{M}_{\mathcal{G}_l}$. We now prove by induction on l that for $1 \leq l \leq m$, for all $\theta_1, \theta_2 \in \Gamma_l$, $P(\cdot|\theta_1) \neq P(\cdot|\theta_2)$, from which our desired result follows as a special case. The base case, $l = 1$, is immediate: if $\theta\rfloor_1 \neq \theta'\rfloor_1$, then for some $x_1 \in \mathbf{E}_1$, $P(x_1|\theta\rfloor_1) \neq P(x_1|\theta'\rfloor_1)$.

Now for the induction step. Suppose for every $\theta\rfloor_l$ and $\theta'\rfloor_l$ such that $\theta\rfloor_l \neq \theta'\rfloor_l$ we have $P(x_1, \ldots, x_l|\theta\rfloor_l) \neq P(x_1, \ldots, x_l|\theta'\rfloor_l)$ for at least one $(x_1, \ldots, x_l) \in \mathbf{E}_1 \times \ldots \times \mathbf{E}_l$. Now consider any θ and θ' such that $\theta\rfloor_{l+1} \neq \theta'\rfloor_{l+1}$. For these θ and θ' we either (first case) have $\theta\rfloor_l \neq \theta'\rfloor_l$ or (second case) $\theta\rfloor_l = \theta'\rfloor_l$ and $\theta^{l+1} \neq (\theta')^{l+1}$. In the first case, suppose by means of contradiction that

$$P(x_1, \dots, x_{l+1}|\theta\rfloor_{l+1}) = P(x_1, \dots, x_{l+1}|\theta'\rfloor_{l+1}) \text{ for all } x_1, \dots, x_{l+1} \tag{6.19}$$

It follows that for all $x_1, \dots, x_l$

$$P(x_1, \dots, x_l|\theta\rfloor_l) = \sum_{x_{l+1}} P(x_1, \dots, x_{l+1}|\theta\rfloor_{l+1}) = P(x_1, \dots, x_l|\theta'\rfloor_l)$$

where the second equality follows from Equation (6.19). But this contradicts our induction hypothesis.

In the second case, there must be a q_{l+1} and an x_{l+1} such that $\theta^{l+1}_{x_{l+1}|q_{l+1}} \neq (\theta')^{l+1}_{x_{l+1}|q_{l+1}}$. There must also be an assignment $X_1 = x_1, \dots, X_l = x_l$ such that $Pa_{l+1} = q_{l+1}$. Consider the data vector $\vec{x} = (x_1, \dots, x_l, x_{l+1})$ for the $x_1, \dots, x_l$ and x_{l+1} just defined. It is easy to see that $P(\vec{x}|\theta\rfloor_{l+1}) = c \cdot \theta^{l+1}_{x_{l+1}|q_{l+1}}$ while $P(\vec{x}|\theta'\rfloor_{l+1}) = c \cdot (\theta')^{l+1}_{x_{l+1}|q_{l+1}}$ for the same constant c, so $P(\vec{x}|\theta\rfloor_{l+1}) \neq P(\vec{x}|\theta'\rfloor_{l+1})$ which is what we had to prove. $\Box$

6.4 Predictive Distributions for Bayesian Networks

In the previous section we defined the model class of Bayesian networks together with the functional form of the prior distributions we will use for this model class. This allows us to write the predictive distributions $\mathcal{P}_{map}$ (6.3) and $\mathcal{P}_{av}$ (6.5) more explicitly, as will be shown in the next two sections. The direct stochastic complexity predictive distribution $\mathcal{P}_{sc}$ (6.8) is instantiated for the Bayesian network case in Section 6.4.3. In order to determine the $\mathcal{P}_{jef}$ predictive distribution (6.12), we show in Section 6.4.4 how to compute Jeffreys' prior for Bayesian network model classes.

6.4.1 The $\mathcal{P}_{map}$ Predictive Distribution for Bayesian Networks

Let $\mathcal{M}_{\mathcal{G}}$ be a Bayesian network family for arbitrary network structure $\mathcal{G}$. We now interpret the elements in $\mathcal{M}_{\mathcal{G}}$ as i.i.d. probabilistic models: $P(\cdot|\theta) \in \mathcal{M}_{\mathcal{G}}$ stands for the hypothesis that the data are i.i.d., each single observation being distributed according to $P(\cdot|\theta)$. We denote by $X_1, \dots, X_m$ the random variables in $\mathcal{G}$, such that $(X_1, \dots, X_m)$ takes on values in $\mathbf{E} = \mathbf{E}_1 \times \dots \times \mathbf{E}_m$. Let $D = (\vec{x}_1, \dots, \vec{x}_n) \in \mathbf{E}^n$. We denote by X_{ji} the instance of random variable X_i for the j-th outcome. We denote by Pa_{ji} the instance of random variable Pa_i for the j-th outcome. Let $x_i \in \mathbf{E}_i$ and let $q_i \in \mathbf{E}_{\mathrm{pa}_i}$. We will use indicator functions (see Chapter 3, page 51) $\mathcal{I}(X_{ji} = x_i)$ (abbreviated to $\mathcal{I}_j(x_i)$) and $\mathcal{I}(Pa_{ji} = q_i)$ (abbreviated to $\mathcal{I}_j(q_i)$):

$$\mathcal{I}_j(x_i) = \begin{cases} 1, & \text{if } X_{ji} = x_i, \\ 0, & \text{otherwise.} \end{cases}, \text{ and } \mathcal{I}_j(q_i) = \begin{cases} 1, & \text{if } Pa_{ji} = q_i, \\ 0, & \text{otherwise.} \end{cases} \tag{6.20}$$

For a test vector $\vec{x}$ we simply omit the subscript j and write $\mathcal{I}(x_i)$ and $\mathcal{I}(q_i)$.

As already noted in Section 6.2.2, the MAP predictive distribution can be determined by computing the likelihood of a test vector $\vec{x} = (\vec{u}, \vec{v})$:

$$\mathcal{P}_{map}(\vec{x} \mid D) = P(\vec{x} \mid \breve{\theta}(D)) = \prod_{i=1}^{m} \prod_{q_i \in \mathbf{E}_{\mathrm{pa}_i}} \prod_{x_i \in \mathbf{E}_i} (\breve{\theta}^i_{x_i|q_i})^{\mathcal{I}(x_i)\mathcal{I}(q_i)}, \tag{6.21}$$

where $\mathcal{I}(x_i)$ and $\mathcal{I}(q_i)$ are indicator variables for the test vector $\vec{x}$ and $\breve{\theta}^i_{x_i|q_i}$ is the MAP model given by (see, for example, [71])

$$\breve{\theta}^i_{x_i|q_i} = \frac{f^i_{x_i|q_i} + \mu^i_{x_i|q_i} - 1}{\sum_{l=1}^{k_i} \left(f^i_{l|q_i} + \mu^i_{l|q_i} \right) - k_i},$$

Here $\mu^i_{x_i|q_i}$ are the hyperparameters as defined in Section 6.3.2 and $f^i_{x_i|q_i}$ are the sufficient statistics of the training data D: $f^i_{x_i|q_i}$ is the number of data instantiations where random variable X_i has taken on value x_i and the parents of X_i have configuration q_i:

$$f^i_{x_i|q_i} = \sum_{j=1}^{n} \mathcal{I}_j(x_i)\mathcal{I}_j(q_i)$$

With the uniform prior all the hyperparameters $\mu^i_{x_i|q_i}$ are set to 1 (Section 6.3.2), in which case we get the standard maximum likelihood estimator,

$$\hat{\theta}^i_{x_i|q_i} = \frac{f^i_{x_i|q_i}}{\sum_{l=1}^{k_i} f^i_{l|q_i}}.$$

6.4.2 The $\mathcal{P}_{av}$ Predictive Distribution for Bayesian Networks

The evidence predictive distribution (6.5) is defined as an integral over the parameter space. As shown in [29, 71], with Bayesian networks this integral can be solved analytically, yielding

$$\mathcal{P}_{av}(\vec{x} \mid D) = \prod_{i=1}^{m} \prod_{q_i \in \mathbf{E}_{\mathrm{pa}_i}} \prod_{x_i \in \mathbf{E}_i} (\bar{\theta}^i_{x_i|q_i})^{\mathcal{I}(x_i)\mathcal{I}(q_i)} \tag{6.22}$$

where

$$\bar{\theta}^i_{x_i|q_i} = \frac{f^i_{x_i|q_i} + \mu^i_{x_i|q_i}}{\sum_{l=1}^{k_i} \left(f^i_{l|q_i} + \mu^i_{l|q_i} \right)}.$$

From (6.22) we see that (perhaps somewhat surprisingly) similarly to the $\mathcal{P}_{map}$ predictive distribution case, the resulting predictive distribution can be regarded as a likelihood of the test vector $\vec{x}_t$, but now taken at the mean of the posterior rather than at the mode (maximum).

6.4.3 The $\mathcal{P}_{sc}$ Predictive Distribution for Bayesian Networks

The stochastic complexity predictive distribution is proportional to the likelihood of the combined data set $D^+ = D \cup \vec{x}$ at the maximum likelihood point:

$$\mathcal{P}_{sc}(\vec{x} \mid D) \propto P(D^+ \mid \hat{\theta}(D^+)) = \prod_{i=1}^{m} \prod_{q_i \in \mathbf{E}_{pa_i}} \prod_{x_i \in \mathbf{E}_i} (\hat{\theta}^i_{x_i|q_i})^{(f^i_{x_i|q_i})^+},$$

where

$$\hat{\theta}^i_{x_i|q_i} = \frac{(f^i_{x_i|q_i})^+}{\sum_{l=1}^{k_i} (f^i_{l|q_i})^+},$$

and $(f^i_{l|q_i})^+$ are the sufficient statistics of D^+. Consequently, the predictive distribution can also in this case be regarded as a likelihood of the test vector $\vec{x}$, but the maximum likelihood estimator is now computed from the extended data set, consisting of the original data set together with the test vector itself.

6.4.4 The $\mathcal{P}_{jef}$ Predictive Distribution for Bayesian Networks

As can be seen from (6.9), Jeffreys' prior is proportional to the square root of the determinant of the Fisher information matrix $I(\theta)$. We proceed to compute this matrix. Observe that, for a Bayesian network distribution $P(\cdot|\theta) \in \mathcal{M}_{\mathcal{G}}$, the log-likelihood of a single data vector $\vec{x}$ can be written as

$$\log P(\vec{x} \mid \theta) = \sum_{i=1}^{m} \sum_{q_i \in \mathbf{E}_{pa_i}} \mathcal{I}(q_i) \left(\sum_{x_i=1}^{k_i - 1} \left(\mathcal{I}(x_i) \log \theta^i_{x_i|q_i} \right) + \mathcal{I}(k_i) \log \theta^i_{k_i|q_i} \right). \quad (6.23)$$

where $\mathcal{I}(q_i)$ and $\mathcal{I}(x_i)$ are defined as in (6.20).

Let us consider the element $(\theta^{i_1}_{l_1|q_{i_1}}, \theta^{i_2}_{l_2|q_{i_2}})$ of the second derivative (Hessian) matrix of (6.23). If either the variable indices i_1, i_2 or the parent configurations q_{i_1}, q_{i_2} are different, then clearly the second derivative is zero, and thus also the corresponding element of the information matrix is zero. It follows that the only non-zero elements of the information matrix are in sub-matrices where both parameters in question have the same variable and configuration index. Consider one of these sub-matrices $I^i_{q_i}(\theta)$, where i is the variable index and q_i the parent configuration. After some simple calculus we get

$$-\frac{\partial^2 \log P(\vec{x} \mid \theta)}{\partial \theta^i_{l_1|q_i} \partial \theta^i_{l_2|q_i}} = \begin{cases} \frac{\mathcal{I}(q_i)\mathcal{I}(k_i)}{(\theta^i_{k_i|q_i})^2}, \text{ if } l_1 \neq l_2, \\ \\ \frac{\mathcal{I}(q_i)\mathcal{I}(l_i)}{(\theta^i_{l_i|q_i})^2} + \frac{\mathcal{I}(q_i)\mathcal{I}(k_i)}{(\theta^i_{k_i|q_i})^2}, \text{ if } l_1 = l_2. \end{cases} \quad (6.24)$$

The elements of the Fisher information matrix are now the expectations of (6.24) over the set of all possible data vectors $\vec{x} \in \mathbf{E}$. For the case where $l_1 \neq l_2$ we get

$$\mathrm{E}_\theta \left[\frac{-\partial^2 \log P(\vec{x} \mid \theta)}{\partial \theta^i_{l_1|q_i} \partial \theta^i_{l_2|q_i}} \right] = \sum_{\vec{x} \in \mathbf{E}} P(\vec{x} \mid \theta) \frac{\mathcal{I}(q_i)\mathcal{I}(k_i)}{(\theta^i_{k_i|q_i})^2} = \tag{6.25}$$

$$= \frac{P(Pa_i = q_i, X_i = k_i \mid \theta)}{(\theta^i_{k_i|q_i})^2} = \frac{P(Pa_i = q_i \mid \theta)}{\theta^i_{k_i|q_i}}. \tag{6.26}$$

Similarly, with $l_1 = l_2$ we get

$$\mathrm{E}_\theta \left[\frac{-\partial^2 \log P(\vec{x} \mid \theta)}{\partial (\theta^i_{l|q_i})^2} \right] = \frac{P(Pa_i = q_i \mid \theta)}{\theta^i_{l|q_i}} + \frac{P(Pa_i = q_i \mid \theta)}{\theta^i_{k_i|q_i}}. \tag{6.27}$$

which gives the $(k_i - 1) \times (k_i - 1)$ sub-matrix $I^i_{q_i}(\theta)$ of the Fisher information matrix:

$$I^i_{q_i}(\theta) = \begin{pmatrix} (\frac{P^i_{q_i}}{\theta^i_{1|q_i}} + \frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) & (\frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) & \cdots & (\frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) \\ (\frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) & (\frac{P^i_{q_i}}{\theta^i_{2|q_i}} + \frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) & \cdots & (\frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) \\ \vdots & \vdots & \ddots & \vdots \\ (\frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) & (\frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) & \cdots & (\frac{P^i_{q_i}}{\theta^i_{k_i-1|q_i}} + \frac{P^i_{q_i}}{\theta^i_{k_i|q_i}}) \end{pmatrix}, \tag{6.28}$$

where $P^i_{q_i} = P(Pa_i = q_i|\theta)$. One can show by performing a sequence of Gaussian elimination steps on this matrix (see [18]) that its determinant is given by

$$|I^i_{q_i}(\theta)| = \frac{(P^i_{q_i})^{k_i-1}}{\prod_{l \in \mathbf{E}_i} \theta^i_{l|q_i}}. \tag{6.29}$$

The whole Fisher information matrix $I(\theta)$ is a block diagonal matrix, where the blocks are the sub-matrices $I^i_{q_i}(\theta)$. The determinant of a block diagonal matrix is the product of the determinants of the blocks, and thus

$$|I(\theta)| = \prod_{i=1}^{m} \prod_{q_i \in \mathbf{E}_{\mathrm{pa}_i}} \frac{(P^i_{q_i})^{k_i-1}}{\prod_{l \in \mathbf{E}_i} \theta^i_{l|q_i}}. \tag{6.30}$$

Finally, as noted in Section 6.2.4, $\pi(\theta) \propto \sqrt{|I(\theta)|}$, so we get

$$\pi(\theta) \propto \prod_{i=1}^{m} \prod_{q_i \in \mathbf{E}_{\mathrm{pa}_i}} (P^i_{q_i})^{\frac{k_i-1}{2}} \prod_{l \in \mathbf{E}_i} (\theta^i_{l|q_i})^{-\frac{1}{2}} \propto \prod_{i=1}^{m} \prod_{q_i \in \mathbf{E}_{\mathrm{pa}_i}} (P^i_{q_i})^{\frac{k_i-1}{2}} \prod_{l \in \mathbf{E}_i} (\theta^i_{l|q_i})^{-\frac{1}{2}}. \tag{6.31}$$

Computing Jeffreys' prior as formulated above requires computing for each variable the marginal distribution of its parents. Unfortunately, for multi-connected Bayesian networks, this problem is NP-hard [28]. In the experiments reported in Section 6.5, we used a simple tree-structured Bayesian network, in which case Jeffreys' prior is of a proper conjugate (Dirichlet) form, and it can be computed efficiently.

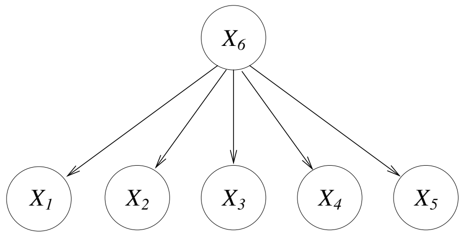


Figure 6.2: An example Naive Bayes structure. In this case, $m = 6$ and the feature variables are $X_1, \ldots, X_5$. *Given* the value of X_6 (the class value), the feature variables become independent.

6.5 Empirical Results

6.5.1 Experimental Setup

In our experiments, we concentrated on the standard classification problem, where the task is to predict the value of the single *classification variable* X_m, given the values of all the other variables. As explained in Section 6.2.1, this is achieved by computing probabilities of the form $\mathcal{P}(x_m \mid x_1, \ldots, x_{m-1}, D)$.

We used as our model class the class of Naive Bayes classifiers [42]. In a Naive Bayes model the clamped variables $X_1, \ldots, X_{m-1}$ are assumed to be independent, given the value of variable X_m. Consequently, we can regard the Naive Bayes model as a simple tree-structured Bayesian network, where variable X_m forms the root of the tree, and variables $X_1, \ldots, X_{m-1}$ are represented by the leaves. Figure 6.2 gives an example of a Naive Bayes structure. An informal introduction to the Naive Bayes model class was given on page 125.

Definition 6.7 (Naive Bayes) *Let* $\mathbf{E} = \mathbf{E}_1 \times \cdots \times \mathbf{E}_m$. *Let for* $1 \leq i \leq m - 1$, $Pa_i = \{X_m\}$ *and let* $\mathcal{G} = \{Pa_1, \ldots, Pa_{m-1}\}$. *The Naive Bayes model class for feature variables* $X_1, \ldots, X_{m-1}$ *and class variable* X_m *is the class of Bayesian network distributions with structure* $\mathcal{G}$.

Let $\mathcal{M}_{nb}$ be the class of Naive Bayes models over $m - 1$ feature variables and $K = k_m$ different class values. In this case the Jeffreys' prior formula (6.31) reduces to (writing θ_k^m as short for $\theta_{k|1}^m = P(X_m = k|\theta)$):

$$\begin{aligned} \pi(\theta) &\propto \prod_{k=1}^{K} (\theta_k^m)^{-\frac{1}{2}} \prod_{i=1}^{m-1} \prod_{k'=1}^{K} (\theta_{k'}^m)^{\frac{k_i - 1}{2}} \prod_{l \in \mathbf{E}_i} (\theta_{l|k'}^i)^{-\frac{1}{2}} \\ &= \prod_{k=1}^{K} (\theta_k^m)^{\frac{1}{2}(\sum_{i=1}^{m-1}(k_i - 1) - 1)} \prod_{i=1}^{m-1} \prod_{k'=1}^{K} \prod_{l \in \mathbf{E}_i} (\theta_{l|k'}^i)^{-\frac{1}{2}}, \end{aligned} \tag{6.32}$$

which by Definition 6.5 can be seen to be a Dirichlet density.

For our experiments with the Naive Bayes classifier, eight public domain classification data sets of varying size were used (the data sets can be obtained from the UCI data repository[2]). Table 6.1 describes the size (n), the number of attributes (m), and the number of classes (K) for each of these data sets. All continuous attributes were discretized. To get an idea of how such a data set looks like we refer back to page 125 where the 'heart disease' set was discussed in some detail.

Dataset	Data vectors	Attributes	Classes	CV folds
Heart Disease (HD)	270	14	2	9
Iris (IR)	150	5	3	5
Lymphography (LY)	148	19	4	5
Australian (AU)	690	15	2	10
Breast Cancer (BC)	286	10	2	11
Diabetes (DB)	768	9	2	12
Glass (GL)	214	10	6	7
Hepatitis (HE)	150	20	2	5

Table 6.1: The datasets used in the experiments

For comparing the predictive accuracy of different predictive distributions, we used two different score functions: the *log-score* and the *0/1-score*. The log-score of a predictive distribution $\mathcal{P}(\cdot|\vec{u}, D)$ when the actual outcome is $X_m = k$, is defined as $-\log \mathcal{P}(X_m = k|\vec{u}, D)$. It thus coincides with the logarithmic loss; see Chapter 2, page 44 for several interpretations of this loss function.

For the 0/1-score, we simply first determine the k for which the probability $\mathcal{P}(X_m = k|\vec{u}, D)$ is maximized, and the 0/1-score is then defined to be 1, if the actual outcome indeed was k, otherwise it is defined to be 0.

Two separate sets of experiments were performed on each data set by using the following predictive inference methods with the Naive Bayes modelclass:

- ML: The $\mathcal{P}_{map}$ predictive distribution (6.3) with uniform prior (equivalent to the predictive distribution with the maximum likelihood model).
- EV: The $\mathcal{P}_{av}$ predictive distribution (6.5) with uniform prior.
- SC: The $\mathcal{P}_{sc}$ predictive distribution (6.8).
- EJ: The $\mathcal{P}_{jef}$ predictive distribution (6.12), which coincides with the $\mathcal{P}_{av}$ distribution (6.5) with Jeffreys' prior (6.32).

Note that all these predictive distributions are 'consistent' [164] in the following sense: in the hypothetical situation in which the data are truly generated by repeated sampling from a Naive Bayes model θ^*, then as D gets larger all four predictive distributions $\mathcal{P}(\cdot|D)$ will converge, with probability 1, to $P(\cdot|\theta^*)$. For the ML approach, this follows from the consistency of maximum likelihood estimators for exponential families (recall that Naive Bayes is an exponential family); see [164]. For the SC approach, we see from (6.8) and (6.7) that $\mathcal{P}_{sc}(\cdot|D)$ converges to $P(\cdot|\hat{\theta}(D))$ as the size of D increases.

[2] "http://www.ics.uci.edu/~mlearn/".

Consistency of the EJ and EV approach can be seen to follow from the consistency of the ML approach: compare the explicit formula for the ML predictive distribution $\mathcal{P}_{map}(\cdot \mid D)$ (Equation 6.21) with that of the EJ and EV approaches $\mathcal{P}_{av}(\cdot \mid D)$ (Equation 6.22) and observe that the estimates of the single parameters $\breve{\theta}^i_{x_i|q_i}$ and $\bar{\theta}^i_{x_i|q_i}$ occurring in these equations converge to the same value for increasing sample size.

We see that for very large samples all predictive distributions considered here are optimal in the sense that they are 'consistent'. Interesting differences will therefore only occur for samples of limited size. In that case, all four distributions have properties which make them an interesting choice to base predictions on: ML since it embodies the standard method of classical, 'Fisherian' statistics [48]; EV since the uniform prior is more or less the standard way of using the Bayesian evidence in discrete domains when there is no prior knowledge; SC since it is based on minimizing worst-case logarithmic loss; and EJ on the one hand since it is also based on minimizing worst-case logarithmic loss but is not dependent on knowledge of the sample size; and on the other hand because, according to some Bayesians, Jeffreys' prior, rather than the uniform one, is the sole prior expressing 'no prior knowledge' [17].

In the first set of experiments (Section 6.5.2) we measured the crossvalidated prediction performance by using the two score functions described above for training sets containing a substantial amount of data. In our second set of experiments (Section 6.5.3), we studied how the prediction quality of our various approaches depends on the size of the training set D.

6.5.2 Crossvalidation Results

In the machine learning literature, cross-validation has become a standard means of testing an algorithm's prediction and classification performance; here we briefly explain how k-fold cross-validation works; for more details about the procedure we refer to [22].

The k-fold cross-validation procedure for a sample D of size n starts by (randomly) partitioning D into k sets $D_1, \ldots, D_k$ of size (as close as possible to) n/k. These sets are called the ***folds***. The idea is now to use the data in the union of $k-1$ of these sets as training data and to use the data in the remaining set as test data. We achieve this by setting $i := 1$ and determining the predictive distribution $\mathcal{P}(\vec{x}|D \setminus D_i)$ where $D \setminus D_i$ stands for the training sample D with the subsample D_i removed. We use this distribution to predict all items in D_i and we denote by $\text{SCORE}(D_i)$ the accumulated score obtained in these predictions. For example, in case of the log-score, let $D_i = (\vec{x}_1, \ldots, \vec{x}_{n_i})$. Then $\text{SCORE}(D_i) := \text{LOSS}_{lg}(D_i) = -\sum_{1 \leq j \leq n_i} \log \mathcal{P}(\vec{x}_j|D \setminus D_i)$. We repeat this procedure to determine $\text{SCORE}(D_i)$ for all $1 \leq i \leq k$. We then divide each score $\text{SCORE}(D_i)$ by the number of items in D_i to obtain the average score for test-set D_i. Finally, we add the k average test-set scores and divide by k to obtain the *cross-validated prediction score*.

In our experiments, we initially used with each of the datasets the same number of folds as in the major experimental comparison performed in the Statlog project [111] (the number of folds used in each case can be found in Table 6.1). It should be noted though that although the result of one crossvalidation run is an average of n numbers, where n is the number of folds used, the result depends of course on how the n folds are selected from the sample data. To see how much the results vary with different fold partitionings, we performed 100 independent crossvalidation runs where the data was randomly partitioned into n folds, and computed the minimum, the average, and the maximum of the crossvalidated prediction accuracies obtained. As can be seen in Figures 6.3 (in the log-score case) and 6.4 (in the 0/1-score case), the crossvalidation results can vary quite a lot depending on the specific fold partitioning used.

Though the differences in crossvalidation results between different prediction methods are small, we see that for the log-score, evidence with uniform prior performs consistently better than the other methods, followed very closely by the evidence with Jeffreys' prior. The ML approach produces the worst results. For the 0/1-score, the picture is not as clear-cut. It should be noted that in this case the ML approach produces the best results with two of the data sets (GL and HE). A partial explanation for this fact may be that for the much coarser 0/1-score, it is in many cases not important exactly what probability we attach to a class value being k; all probability distributions over the class values for which k gets the maximum probability will lead to the same prediction. Thus it can very well happen that, while the ML prediction captures less well the regularities underlying the data (and hence performs worse with respect to log-score), it still captures them well enough to give maximum probability to the class value that should actually receive maximum probability.

In addition to comparison purposes between the different predictive distributions, the results in Figures 6.3 and 6.4 are interesting as they show surprisingly good performance of the Naive Bayes model when compared to the results reported in the machine learning literature. This highly competitive behaviour of 'Naive' Bayes has been noted before by several authors [50, 90].

The high variance of the results obtained indicate that one single n-fold crossvalidation run cannot be used as a reliable measure for comparing various predictive inference methods, unless the same specific fold partitioning is used in all cases. If, however, a number of independent runs is performed, then some statistical measure, such as the average, can be used for this purpose. Alternatively, the leave-one-out results[3] seem to follow the behavior of the averaged crossvalidation results quite accurately. For this reason, we decided to restrict ourselves to leave-one-out crossvalidation in the next section.

6.5.3 Results with Varying Amount of Training Data

To see how the prediction quality of our various approaches depends on the size of the training set D, we performed a set of experiments using only small fractions of the available training data. In these experiments, leave-one-out crossvalidation was used, but at each step, only the k first vectors from the training set were used in order to predict the single test vector that was "left out", and this procedure was repeated for $k = 1, \ldots, n-1$. As this setup is dependent on the ordering of the data vectors, the whole leave-one-out crossvalidation cycle was then repeated 100 times with 100 randomly generated permutations of the dataset. The averaged (over the 100 leave-one-out crossvalidation runs) results are plotted as a function of k in Figures 6.5–6.8

[3]'leave-one-out crossvalidation' over a data set of n observations is defined as n-fold crossvalidation over this data set.

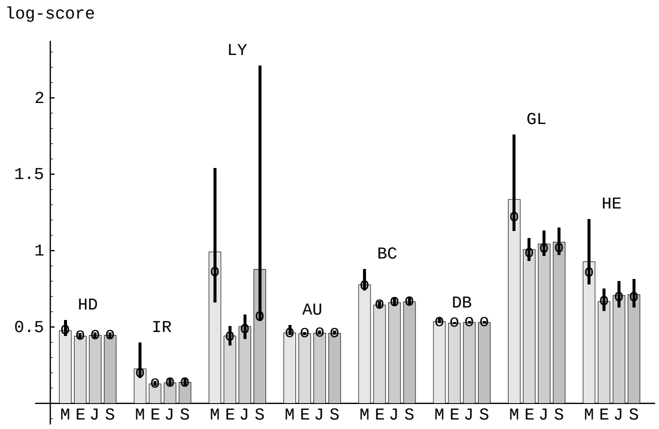


Figure 6.3: The minimum (lower end of the black line), the average (grey bar), and the maximum (upper end of the black line) of the crossvalidated log-score (= prediction error with logarithmic loss function) obtained by 100 independent crossvalidation runs. The corresponding leave-one-out crossvalidation results are marked with small circles. The y-axis represents the log-score, so the smaller the score, the better. The prediction methods used are ML (denoted here by M), EV (E), EJ (J), and SC (S).

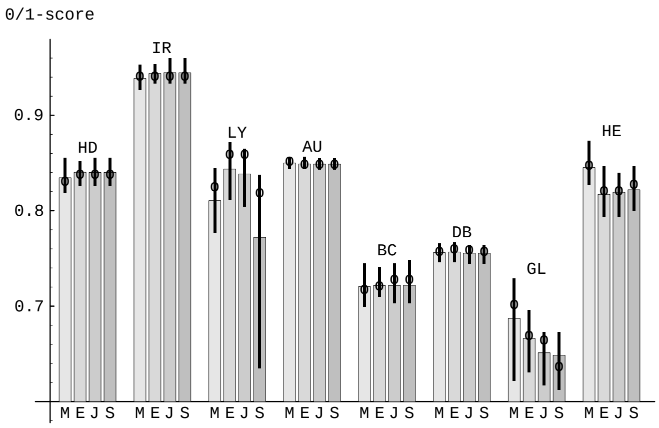


Figure 6.4: The minimum (lower end of the black line), the average (grey bar), and the maximum (upper end of the black line) of the crossvalidated 0/1-scores obtained by 100 independent crossvalidation runs. The corresponding leave-one-out crossvalidation results are marked with small circles. In this picture higher score is better. The prediction methods used are ML (denoted here by M), EV (E), EJ (J), and SC (S).

for the eight datasets used. These statistics of the behavior of different predictive distributions as a function of increasing amount of training data should now give us some idea as to the typical behavior of our prediction methods. For small sample sizes, the ML method will sometimes yield infinitely bad log-score. In order to prevent scaling problems when presenting the results graphically, the *pr-score* (the probability of the correct class, instead of its logarithm) was used in these tests as the alternative score for the 0/1-score, and not the log-score.

All in all, the results with all the eight data sets used show very similar behavior: the evidence-based EV and EJ approaches perform surprisingly well even in cases where the training data consists of only a few data vectors, which shows that the data sets used here are quite redundant, and when properly used, only a very small sample of these data sets is needed for constructing good models.

We now analyze three further interesting aspects of the results.

ML vs. the Rest It is a well-known fact that, for small sample sizes, the ML predictor is too dependent on the observed data and does not take into account that future data *may* turn out to be different. Our results support this observation and show that compared to the other methods, the ML predictive distribution appears to be much more sensitive to the amount of data available. This phenomenon can be explained by the fact that the EV and EJ approaches are more conservative methods as they base their predictions on averaging over all the parameter values, while ML makes more "eager" predictions based on the single maximum likelihood estimator. Let us consider a very simple example to illustrate this point. Suppose our data consists of a string of ones and zeros generated by some i.i.d. Bernoulli-process $p = P(X = 1)$. If we have seen an initial string consisting of one '1', and no zeros, then the ML predictor will determine that the probability of the second symbol being a '1' is unity. However, using the EV prediction, this probability is $\frac{2}{3}$. If the next data item turns out to be a '0', then the log-score of the ML (ML) predictor will be $-\infty$ while that of the EV will be $\log 2 - \log 3$. The behavior of the SC and EJ methods lies somewhere in between that of ML and EV. In our Bernoulli example, the probability of the second symbol being a '1' would be $\frac{3}{4}$ for EJ and $\frac{4}{5}$ for SC.

EJ vs. SC We see in the figures that for small sample sizes, EJ usually performs somewhat better than SC. Though this may have something to do with the fact that SC is not based on a probabilistic model (Section 6.2.4), we conjecture there is another reason: $\mathcal{P}_{sc}$ as we use it here is defined so that the method would give a maximally high probability for full unseen vectors $(x_1, \ldots, x_m)$, but the methods were tested in the restricted case, where only one variable (the class variable x_m) was actually predicted. The optimization of the SC method for the supervised classification case would require the use of a *conditional* maximum likelihood estimator of data D instead of the unconditional ML estimator used in equations 6.7 and 6.8. Using the notation of Section 6.2, this is the model θ in model class $\mathcal{M}$ that maximizes the *conditional* probability $P(\vec{v}_1, \ldots, \vec{v}_n | \vec{u}_1, \vec{u}_n, \theta)$. In other words, it is the model maximizing the probability of the sequence that is to be predicted conditioned on the sequence that is given. Apparently the evidence-based predictions are not so sensitive to exactly what

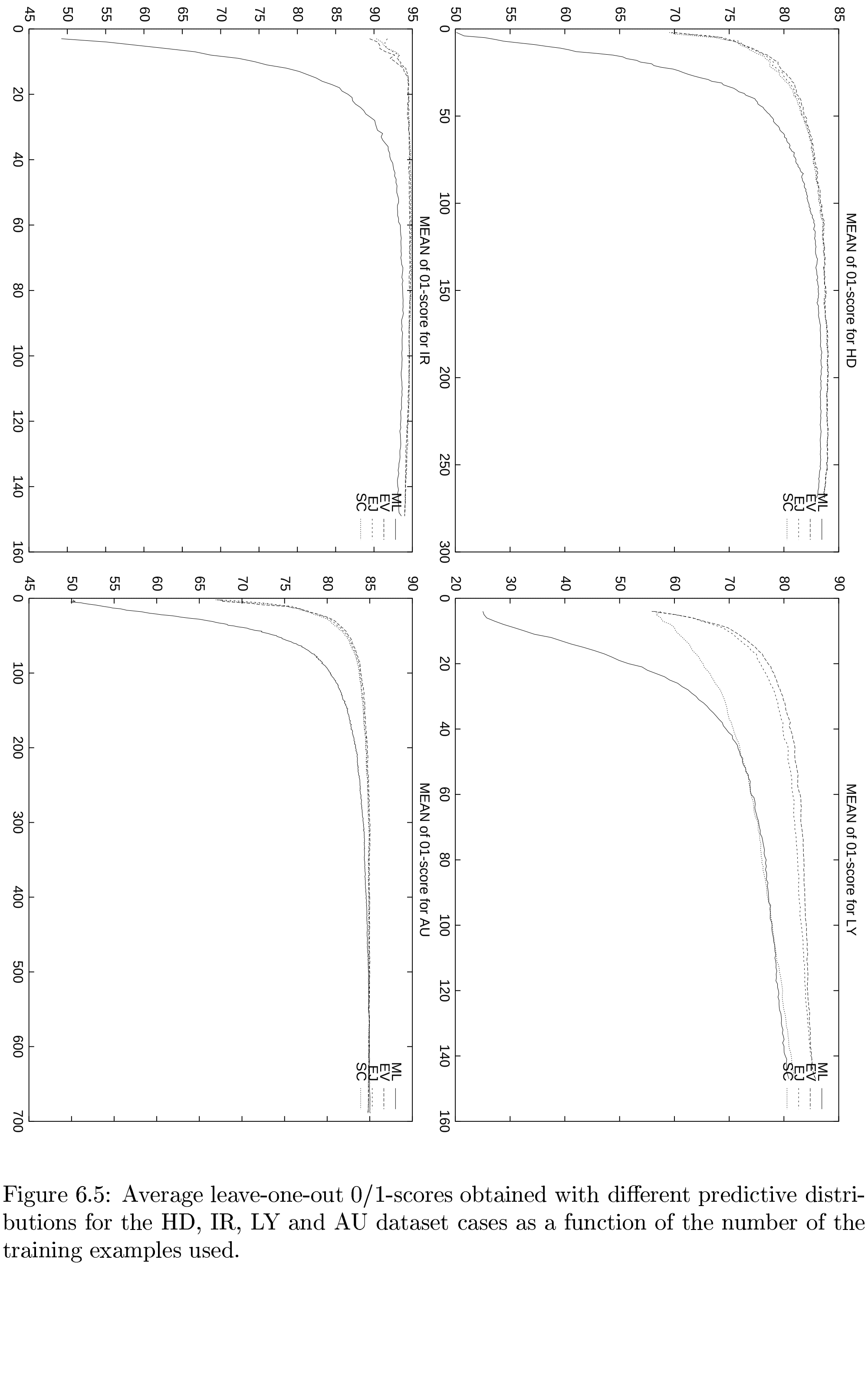


Figure 6.5: Average leave-one-out 0/1-scores obtained with different predictive distributions for the HD, IR, LY and AU dataset cases as a function of the number of the training examples used.

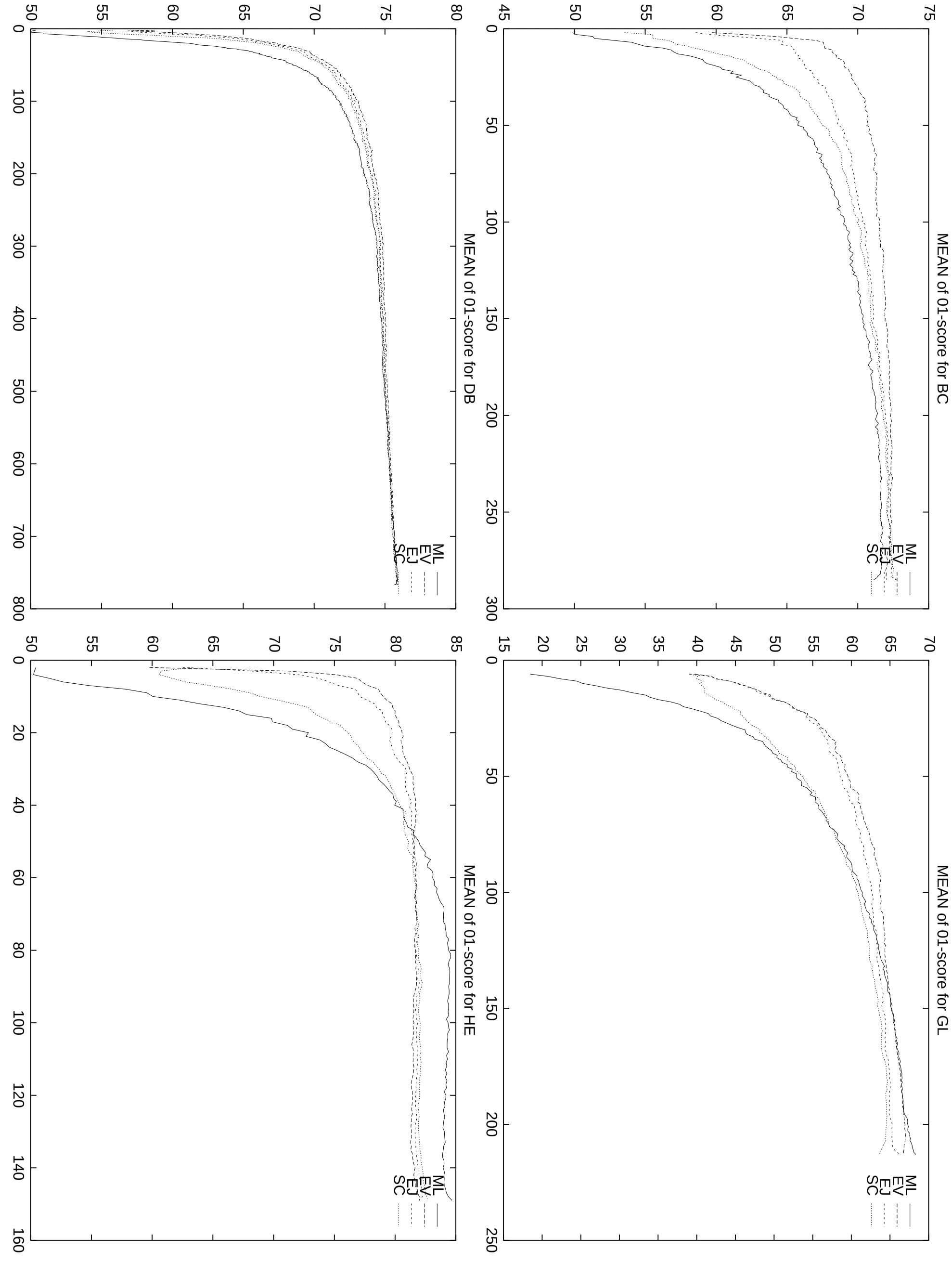


Figure 6.6: Average leave-one-out 0/1-scores obtained with different predictive distributions for the BC, DB, GL and HE dataset cases as a function of the number of the training examples used.

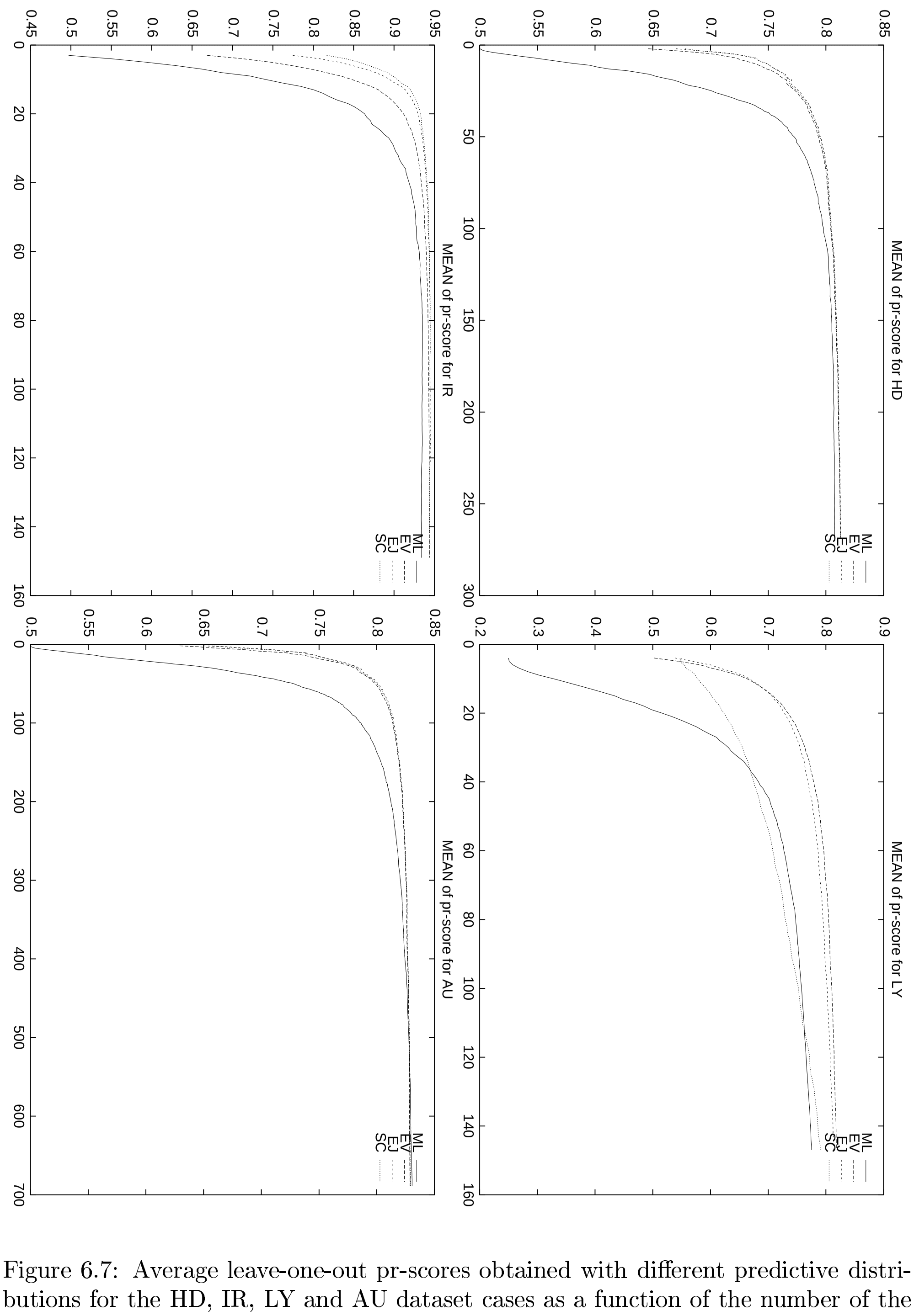


Figure 6.7: Average leave-one-out pr-scores obtained with different predictive distributions for the HD, IR, LY and AU dataset cases as a function of the number of the training examples used.

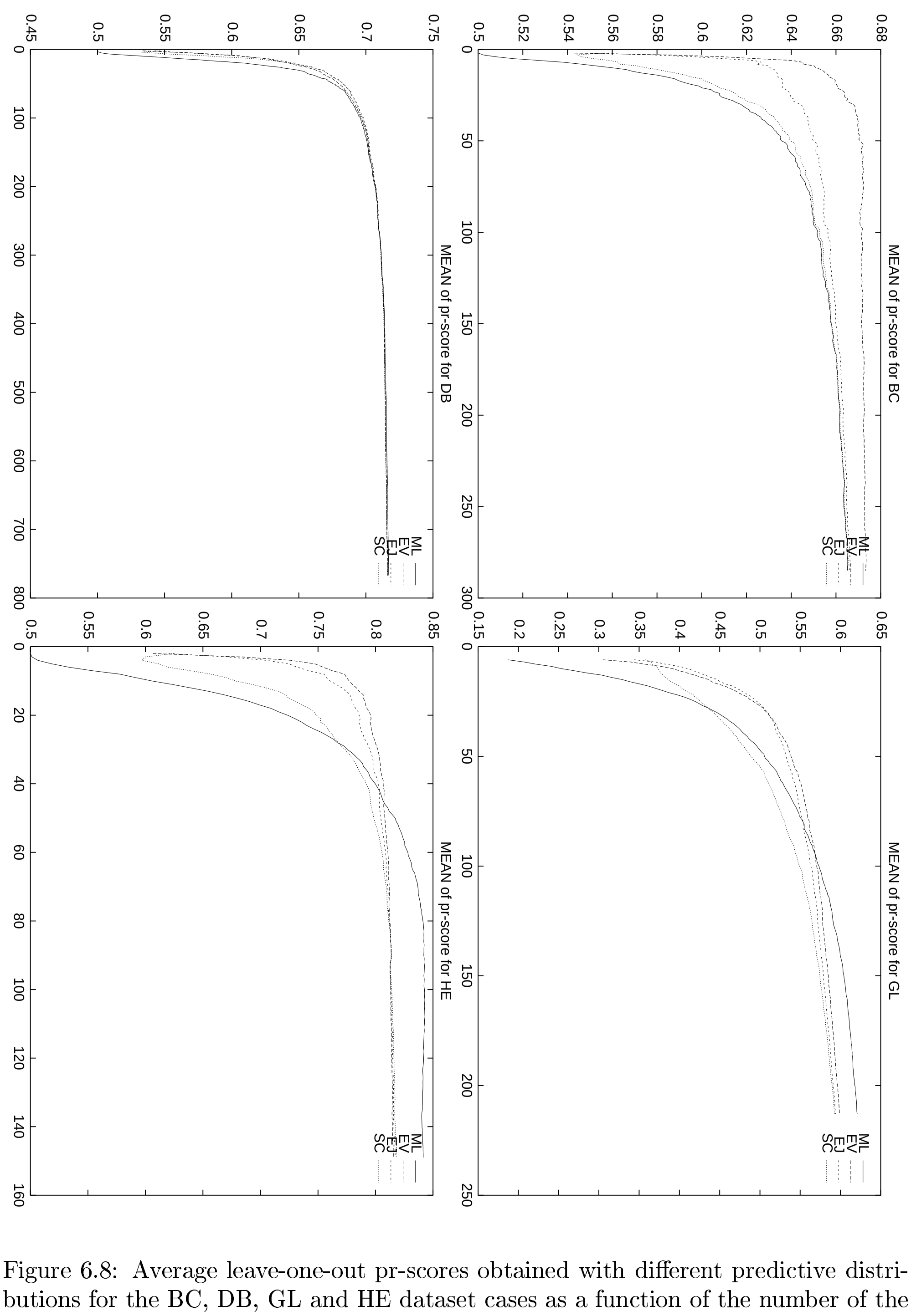


Figure 6.8: Average leave-one-out pr-scores obtained with different predictive distributions for the BC, DB, GL and HE dataset cases as a function of the number of the training examples used.

kind of predictive performance is being optimized. We have no proof of this; it is strongly suggested though by some additional experiments we performed but do not present here. In general, the non-asymptotic difference between EJ and SC seems an interesting topic for further research.

EV vs. EJ and SC Predictions based on EV usually give slightly better results than predictions based on EJ. This may be due to the fact that $-\log P_{jef}(D)$ approximates the stochastic complexity (Equation 2.7, page 2.7) and as such optimizes *worst-case* regret: *whatever* the data D is, $-\log P_{jef}(D)$ will be about equally close to $-\log P(D|\hat{\theta}(D))$, the best-fitting model in the class for D. For the model class of Bernoulli processes, Rissanen [129] showed that this leads to a considerable gain (in the sense of smaller number of bits needed to code the data and hence better predictions in the log-score sense) over $\mathcal{P}_{av}$ with the uniform prior for highly 'skewed' data sets. These are data sets for which the ML estimator lies near the boundary of the parameter space $\mathcal{M}$. The price to pay is that for all other, more 'average' data sets, using P_{jef} leads to a slightly larger code length than using P_{av} with the uniform prior. We conjecture that something similar is happening in our experiments in that the datasets are just not 'skewed' enough for EVP to outperform EV.

6.6 Conclusion

We have described how to obtain four different predictive distributions, one based on the Bayesian MAP estimator, one based on the Bayesian evidence distribution, one based on a direct but possibly non-optimal application of stochastic complexity and one based on a version of stochastic complexity that coincides with the evidence distribution using Jeffreys' prior. We have shown how to compute these predictive distributions for the model class of Bayesian networks.

In the experimental part of the chapter, the predictive accuracy of the ML predictive distribution (the MAP predictive distribution with uniform prior distribution), the evidence predictive distribution (with both uniform and Jeffreys' prior), and the stochastic complexity distribution was evaluated empirically by using publicly available classification data sets. For computational reasons, the specific model used in the tests was the structurally simple Naive Bayes model. In the experiments performed, in the 0/1-score case with substantial training data there was no clear winner. In the other three cases (0/1-score case with small amount of training data and log-score case with either substantial or small amount of data), the evidence-based predictive distributions outperformed the other methods. The effect of using either the uniform or Jeffreys' prior was extremely small, but fairly consistent in that it occurred in nearly every experiment we performed: the best results were obtained by using the uniform prior. One reason for the evidence predictive distribution performing just a little bit worse with Jeffreys' prior may be caused by the fact that Jeffreys' prior is in a sense a "worst-case" prior as we indicated in the last section.

The results with decreasing amount of training data show that the evidence-based approaches predict surprisingly well even with small training sets. The behavior of the SC method usually converges close to that of the EJ method as the training set size increases. As indicated in the last section, the somewhat worse results with small training sets may be explained by the fact that the SC predictive distribution was defined so that the method would give a maximally high probability for full unseen vectors, but the methods were tested in the restricted case, where only one variable (the class variable) was actually predicted.

As a final remark we repeat that the actual classification accuracies obtained with the Naive Bayes model are surprisingly high, when compared to the results obtained by alternative models. This has actually been noted before by several authors (see for example [90, 50]); the results reported here strengthen these previous conclusions in that they indicate that, when the evidence predictive distribution based on Naive Bayes is used, very good performance may already be achieved for quite small sample sizes. This suggests that in many natural domains, using the Naive Bayes model may not be so naive after all.

Chapter 7

MML vs. MDL

7.1 Introduction

In this chapter we compare MDL to the closely related *Minimum Message Length (MML)* principle. MML was introduced by Wallace and Boulton in 1968 [161], nine years before Rissanen independently introduced the MDL Principle [126]. Like MDL, MML is based on the idea that the more we are able to compress a given set of data, the more we have learned from that set of data. Nevertheless, as discussed in [15], there are subtle differences between these two approaches in both the underlying philosophy and the proposed formal criteria.

Recently [129], Rissanen has refined his MDL approach to incorporate effects on the description length of the data that are due to local geometrical properties of the hypothesis space. Wallace and Freeman (WF) already took these properties into account earlier, in their 1987 paper [162] on MML estimation. It has been informally claimed by several people at several conferences that a large part of Rissanen's 1996 work is already implicit in WF's 1987 paper. In this chapter we investigate this claim, and show that it does not hold: though superficially similar, the refinement of MDL proposed in [129] is quite different from the MML approach proposed in [162]. The difference is even quite dramatic in the sense that in the MDL approach the likelihood of the data, given a model θ, is multiplied by a factor correcting for the local structure of the model space near θ, while in the MML approach it is *divided* by the very same factor. Trying to account for this difference, we discovered an oversight in Wallace and Freeman's 1987 derivation of MML estimators. Based on our analysis, we present two revised versions of MML: a pointwise estimator which gives the MML-optimal single parameter model, and a volumewise estimator which gives the MML-optimal region in the parameter space. Of these estimators, the pointwise MML estimator turns out to coincide with the standard Bayesian MAP estimator, at least if a uniform 'subjective' prior is used. The volumewise MML estimator is found to be related to Rissanen's MDL estimator.

Our theoretical analysis indicates that both the revised MML estimators and the MDL approaches should lead to better predictive performance than the original MML estimator. However, these results (1) are asymptotic in nature (2) are strictly speaking only valid if the uniform prior is used as the subjective prior and (3) they are based on a worst-case analysis which may not necessarily be optimal for practical purposes. For these reasons it is not a priori clear what will happen in practical settings with small sample sizes. Therefore we studied this question empirically. The results suggest that with small data sets, the MDL approach yields more accurate predictions than both the MML estimators. The empirical results also demonstrate that the revised versions introduced here indeed perform better than the original Wallace and Freeman MML estimator.

We would like to emphasize at the outset that we do *not* claim that there is anything wrong with the MML principle per se: the problem we discovered concerns only what WF call 'MML estimators', which are an approximation of the theoretically optimal (but usually not efficiently computable) 'strict' MML estimators. Our results do not say anything about either the strict MML estimators or the MML principle in general.

7.1.1 Structure of this Chapter

Section 7.2 reviews the MML principle and discusses some differences and similarities between MML and MDL. Sections 7.3-7.5 contain our theoretical work. In Section 7.3, we review in detail how the MML estimators were derived in [162], and point out the oversight in the derivation. Based on this analysis, we present our two 'revised' versions of MML. In Section 7.4, we compare both the original and the revised MML estimators to Rissanen's recent [129] refinement of MDL.

In Section 7.5 we discuss how to construct different predictive distributions based on the MML and MDL estimators considered. In Section 7.6 we compare their performance on several real-world datasets. The empirical results obtained demonstrate differences in performance that are consistent with the theoretical analysis.

7.2 The MML Principle

7.2.1 Definitions

We assume throughout this chapter that all data is recorded to a finite accuracy, which implies that the set $\mathbf{E}$ of all possible data values is countable. Let the model class $\mathcal{M} = \{P(\cdot|\theta) \mid \theta \in \Gamma\}$ be finitely parameterized (Chapter 2, Definition 2.7) by Γ where Γ is a bounded region of $\mathbf{R}^k$ and k is a positive integer. For simplicity we assume that $\mathcal{M}$ consists only of i.i.d. probabilistic models.

Two-part Codes

Recall that a *two-part description method* (Chapter 1, page 17) consists of a code C_1 for parameter values ('hypotheses') and a set of codes $C_2(\cdot|\theta)$ for encoding data sequences *with the help of* those parameter values. Since the set of parameter values Γ is uncountable, the code C_1 cannot have codewords for all of them; rather, C_1 will be a function $C_1 : \ddot{\Gamma} \to \mathbf{B}^*$ where $\ddot{\Gamma}$ is some countable subset of Γ.

A data sequence x^n can be encoded in two steps by first encoding a parameter value $\ddot{\theta} \in \ddot{\Gamma}$, and then encoding x^n by the code $C_2(\cdot|\ddot{\theta})$. The number of bits needed to encode data x^n on the basis of model $\ddot{\theta}$ then becomes:

$$L_{1,2}(x^n; \ddot{\theta}) = L_{C_2}(x^n|\ddot{\theta}) + L_{C_1}(\ddot{\theta}) = -\log P(x^n|\ddot{\theta}) + L_{C_1}(\ddot{\theta}) \tag{7.1}$$

7.2.2 The Minimum Message Length (MML) Principle

The basic idea behind MML modeling is to find a two-part coding system and an associated estimator minimizing the *expected* message length (number of bits needed to encode the data), where the expectation is taken over the Bayesian marginal distribution $P_{av}(\cdot|\mathcal{M}) = \int_{\theta \in \Gamma} P(x^n|\theta)w(\theta)d\theta$ (see Chapter 2, Section 2.8). Hence every MML analysis depends on a prior distribution w over the set of parameter values Γ. This prior is interpreted in a subjective Bayesian manner: it is taken to represent the prior knowledge one has about the parameter values [162].

MML thus seeks to find, for each n, the combination of (i) the subset $\ddot{\Gamma}$ of Γ, (ii) the code C_1 and (iii) the estimator $\ddot{\theta} : \mathbf{E}^n \to \ddot{\Gamma}$ minimizing the sum

$$\sum_{x^n \in \mathbf{E}^n} P_{av}(x^n)[-\log P(x^n|\ddot{\theta}(x^n)) + L_{C_1}(\ddot{\theta}(x^n))]. \tag{7.2}$$

The estimator $\ddot{\theta}$ that is optimal in the above sense is called the *strict MML (SMML) estimator* [162]. In practice, it is very hard to find this SMML estimator. For this reason, Wallace and Freeman (WF) propose an approximation to the SMML estimator which they simply call the *MML estimator*. The derivation of the WF MML estimator can be found in Section 7.3.

7.2.3 Differences between MML and MDL

A crucial difference between the MDL and MML principles is that the latter is based on finding a code minimizing *expected* code lengths, while the former is based on finding a code that yields short code lengths for *all* datasets that are well-modeled by $\mathcal{M}$. Another important difference is that the goal of the MML approach is to find an efficient code together with the associated estimator, while MDL is not, in general, concerned with estimators. What is more, the MML approach uses for this purpose always two-part codes; for MDL there are several options, of which the two-part code MDL is only one special case.

It should also be noted that while MML is Bayesian in the sense that the approach is dependent on a subjective prior provided by an external observer, the MDL principle does not depend on any specific prior distribution. To be sure, priors do arise in MDL modeling, but they are merely used as technical tools and *not* as representing prior knowledge about the problem at hand.

7.3 MML Estimators

The strict MML estimator is usually very difficult to find, but there are several ways for constructing an MML estimator approximating SMML. In Section 7.3.1 we present the approximating estimator suggested by Wallace and Freeman in [162] which they simply call the 'MML estimator'. We will refer to it as the WF MML estimator of the WF estimator for short. Our critique towards the WF estimator concerns the code for encoding the parameter values, which is suboptimal. In Section 7.3.2 we present an alternative code that is optimal under worst-case assumptions. We show how using this code leads to a revised pointwise MML estimator. In Section 7.3.3 we go on to show that, using the revised MML estimator and looking at large sample sizes, the optimal MML *region* in the parameter space does not necessarily contain the single optimal revised MML estimator. Basing MML estimators on the optimal region in parameter space turns out to be closely related to Rissanen's 1996 formulation of MDL. Section 7.4 reviews this work and summarizes the differences between the WF MML estimator, the revised MML estimator, the region-based MML estimator and the recent form of MDL. The reader who only wants to get a quick look at the essential differences may wish to skip the remainder of this section and turn to Section 7.4 immediately.

7.3.1 The Wallace and Freeman MML Estimator

In this subsection we summarize the original derivation of the MML estimator as presented in [162]. We concentrate first on the case of a model class $\mathcal{M}$ containing models depending on a single parameter (hence $\Gamma \subset \mathbf{R}^1$). Rather than trying to find the strict-MML code optimizing (7.2), Wallace and Freeman [162] consider the following problem: given an observed data sequence x^n, we are asked to choose an estimate $\theta' \in \Gamma$ together with a *precision quantum* q, so that if x^n is encoded by first stating θ' with precision determined by q and then stating x^n using the code corresponding to the stated estimate, then the length of the encoding is minimized. This means that the estimate θ' is coded using only a limited number of binary places; in other words, a truncated value $\ddot{\theta}$ is obtained from θ' by selecting a value from a quantized scale in which adjacent values differ by q: $|\ddot{\theta} - \theta'| \leq q/2$. Note that the precision *quantum* q gives the *width* between adjacent parameter values, whereas the word 'precision' per se is used in this thesis to denote the number of *bits* needed to encode a parameter value.

While in the SMML setup the goal was to minimize (7.2), now we only ask for a 'target' estimate θ' together with a precision quantum q. Consequently, in contrast to SMML, here we do not require the detailed coding of the actually used estimate $\ddot{\theta}$ to be specified. This makes the approach feasible; the price we pay is that the exact effect of encoding the data using the quantized value $\ddot{\theta}$ instead of θ' cannot be predicted, and the code length can be minimized only in expectation.

We assume that the quantization has the following effect:

$$E[\theta' - \ddot{\theta}] = 0 \qquad \text{(the quantization is unbiased)} \tag{7.3}$$
$$E[(\theta' - \ddot{\theta})^2] = q^2/12 \qquad \text{(as for a uniform distribution)} \tag{7.4}$$

Let w be the prior density of the parameters $\theta \in \Gamma$. The prior probability that θ lies within $\pm q/2$ of a quantized value $\ddot{\theta}$ is approximately $q \cdot w(\ddot{\theta})$. Let $C_{\text{wf}} : \ddot{\Gamma} \to \mathbf{B}^*$ be the code corresponding to this probability. Encoding the estimates $\ddot{\theta}$ using C_{wf}, the expected length of the first part of the message stating θ' to precision quantum

q is $-\log qw(\theta')$. In the second part of the message, we code x^n using the code corresponding to $\ddot{\theta}$, which requires $-\log P(x^n|\ddot{\theta})$ bits.

The length of the second part can be approximated by the following Taylor series expansion:

$$-\log P(x^n|\ddot{\theta}) = -\log P(x^n|\theta') \\ -(\theta'-\ddot{\theta})\frac{\partial}{\partial\theta}\log P(x^n|\theta') - \frac{1}{2}(\theta'-\ddot{\theta})^2\frac{\partial^2}{\partial\theta^2}\log P(x^n|\theta') + \ldots \quad (7.5)$$

Using the expectation of the effects of quantization as given by Equation (7.3) and (7.4), the total length is expected to be (to second order)

$$-\log qw(\theta') - \log P(x^n|\theta') + \frac{q^2}{24}I(x^n;\theta'), \quad (7.6)$$

where $I(x^n;\theta')$ is short for $-\partial^2 \log P(x^n|\theta')/\partial\theta^2$. The expected code length (7.6) is minimized by choosing

$$q^2 = \frac{12}{I(x^n;\theta')}. \quad (7.7)$$

Substituting this optimal precision quantum, we get the expected code length to be

$$-\log w(\theta') + \frac{1}{2}\log\frac{I(x^n,\theta')}{12} - \log P(x^n|\theta') + \frac{1}{2}. \quad (7.8)$$

The value θ' which minimizes this is called the *MML estimate*.

It is clear that in order to decode a two-part message as used here, one must first decode the parameter value $\ddot{\theta}$. For this, one must know the precision quantum q that was used to encode $\ddot{\theta}$. Since the optimal quantum depends on x^n, it is not constant and hence it seems that it must be made part of the code too (see Chapter 1, page 19). However, WF showed that the minimum of the expected message length reached for the optimal precision quantum $q^2 = 12/I(x^n,\theta')$ is very broad with respect to q. This implies that using a quantum q based on the *expectation* of $I(x^n,\theta')$, rather than $I(x^n,\theta')$ itself, will be reasonably efficient for most data values. Hence we can use $q^2 = 12/I_n(\theta')$, where $I_n(\theta')$ is the Fisher (expected) information for n observations:

$$I_n(\theta) = -E_\theta\left[\frac{d^2\log P(x^n|\theta)}{d\theta^2}\right]. \quad (7.9)$$

The advantage of using $I_n(\theta')$ is that now the optimal q is independent of the observed data and becomes a function of θ' only.

This means that there is only one set of possible truncated estimates which can be constructed without reference to the data. We can thus construct a code for the estimate which does not need a precision quantum preamble (for more details about the whole derivation, we refer to [162]). From (7.8) we see that the final definition of the WF MML estimator, from now on denoted by θ'_{wf}, becomes

$$\theta'_{\mathrm{wf}} = \arg\max_{\theta\in\Gamma}\frac{P(x^n|\theta)w(\theta)}{\sqrt{I_n(\theta)}}. \quad (7.10)$$

7.3.2 The MAP/ML Estimator as a Revised Pointwise MML Estimator

In this subsection we point out an oversight in Wallace and Freeman's derivation of MML estimators as given in the previous subsection. For the special case where a uniform prior is used, we present an alternative derivation that leads to a revised MML estimator which coincides with the Bayesian MAP (maximum a posteriori; see Chapter 6, Section 6.2.2) estimator. Since the uniform prior is used the MAP estimator coincides with the ML estimator.

Let us define $q : \Gamma \to \mathbf{R}$ as a function which gives for each value θ' the corresponding optimal precision quantum $q(\theta')$. Using this notation, and substituting $I_n(\theta')$ for $I(x^n; \theta')$ (as prescribed at the end of the previous section), the expected total length (7.6) can be rewritten as

$$L_{\text{mml-wf}}(x^n; \theta') \approx -\log q(\theta') - \log w(\theta') - \log P(x^n|\theta') + \frac{q(\theta')^2}{24} I_n(\theta'). \qquad (7.11)$$

We now make two assumptions. First, we assume that the value θ' which minimizes the expected code length may in principle lie anywhere in the interior of the parameter space. Second, we assume that the number of different possible truncated parameter values is finite, say N. Consequently, we can write $\ddot{\Gamma} = \ddot{\Gamma}_N = \{\ddot{\theta}_1, \ldots, \ddot{\theta}_N\}$. Both assumptions are quite reasonable. For example, the first assumption follows from the requirement that θ' should be consistent, together with the (much stronger) assumption that there exists a true value θ according to which data are actually drawn, and which may lie at any point in the interior of Γ. The WF MML estimator as given in Equation (7.10) is indeed consistent for all combinations of priors and i.i.d. model classes $\mathcal{M}$ for which the Bayesian maximum posterior (MAP) estimator is consistent. In the case of the WF MML estimator, the reason is that the influence of the denominator $\sqrt{I_n(\theta)}$ vanishes as n gets larger. It will be seen that the two revised MML estimators we introduce below are also consistent whenever the MAP estimator is consistent.

The second assumption (finite number of truncated parameter values) is reasonable as long as we allow N to grow with the number of observations n, as we indeed do. We only assume that for each fixed n there is a finite number of candidates N.

Now the point that (in our view) has been overlooked by WF is that nobody *forces* us to code the parameters using the code C_{wf} with lengths $-\log q(\theta') - \log w(\theta')$. Another code may sometimes, or even under fairly general circumstances lead to shorter codelengths. Specifically, the expected extra number of bits you need to code the *data* given a truncated model $\ddot{\theta}$ instead of θ' necessarily depends on the precision quantum $q(\theta')$. But why should we let the encoding of the parameters themselves depend on the $q(\theta')$? Below we present an alternative code where $q(\theta')$ does not influence the codelength of the parameters. The code will be optimal under worst-case assumptions. In our derivation, we will only consider the special case that w is the uniform prior ($w(\theta) = c$ for some constant c). In this case, our alternative code coincides with the *uniform code* C_{uni} rather than the code C_{wf}. The uniform code is simply the code that codes each element of $\ddot{\Gamma}_N$ using the same number of bits $\log N$. It follows directly from the Kraft inequality (Chapter 1, Theorem 1.6) that for every other code $C' : \ddot{\Gamma}_N \rightarrow \mathbf{B}^*$, we have $\max_{\ddot{\theta}_i \in \ddot{\Gamma}_N} L_{C'}(\ddot{\theta}_i) > \max_{\ddot{\theta}_i \in \ddot{\Gamma}_N} L_{C_{\text{uni}}}(\ddot{\theta}_i)$. We say that C_{uni} has the *optimal worst-case code length* (here 'optimal' is used in the sense of 'shortest').

Using C_{uni} instead of C_{wf}, the code length to encode $\ddot{\theta}$ becomes $\log N$ instead of $-\log q(\theta) \cdot w(\theta) = -\log q(\theta) \cdot c$, and the right hand side of (7.11) becomes

$$\log N - \log P(x^n|\theta') + \frac{q(\theta')^2}{24} I_n(\theta'). \tag{7.12}$$

For some θ' this will yield shorter code lengths than (7.11) while for others it will yield larger ones. However, using our assumption that θ' may in principle lie everywhere in the interior of Γ, it makes sense to take a worst-case point of view. We will take this view for granted for the remainder of this section; whether it can always be justified will be discussed in Section 7.3.4.

For the worst-case θ', the expected code length (7.12) is clearly smaller than the expected length (7.11) with $w(\theta)$ instantiated to c. According to the worst-case viewpoint, C_{uni} should then be preferred over C_{wf}, and indeed over every other possible code over the set $\ddot{\Gamma}_N$.

Using C_{uni} and denoting the total length needed to code x^n by $L_{\text{mml-p}}(x^n, \theta')$, Equation 7.12 becomes

$$L_{\text{mml-p}}(x^n, \theta') \approx \log N - \log P(x^n|\theta') + \frac{q(\theta')^2}{24} I_n(\theta'). \tag{7.13}$$

We see from this equation that using C_{uni} for encoding the parameter values is worst-case optimal independently of the way $q(\theta')$ is instantiated. Hence we should base our two-part codes on (7.13) rather than (7.11), and furthermore instantiate (7.13) by using the function $q(\theta')$ that gives shortest expected code lengths. Since we assume that θ' may lie everywhere in Γ, the optimal $q(\theta')$ becomes the function that minimizes the maximum value of the last term in (7.13):

$$q = \arg\min_q \max_{\theta' \in \Gamma} q(\theta')^2 I_n(\theta'), \tag{7.14}$$

which is clearly attained for $q(\theta')^2 \propto I_n(\theta')^{-1}$ (choosing $q(\theta') = 0$ everywhere is not an option since we assume that there exist only N parameter values). By substituting this optimal q back into (7.13), we obtain

$$L_{\text{mml-p}}(x^n, \theta') \approx -\log P(x^n|\theta') + \log N + K, \tag{7.15}$$

where K depends only on N and n, and not on θ. The θ' which minimizes this, however, is simply the standard Bayes posterior mode (or equivalently, the ML estimator); from now on we denote it by θ'_{p}:

$$\theta'_{\text{p}} = \arg\max_{\theta \in \Gamma} P(x^n|\theta) \tag{7.16}$$

Thus, interestingly, we find that in our alternative derivation of MML estimators, if the uniform prior is used, then the optimal MML estimate is just the (Bayesian) MAP estimate. On the other hand, as we see from (7.14), the optimal precision quantum $q(\theta'_{\mathrm{p}})$ at point θ'_{p} remains inversely proportional to $\sqrt{I_n(\ddot{\theta})}$, just like in the original derivation by WF:

$$q(\theta'_{\mathrm{p}}) \propto \frac{1}{\sqrt{I_n(\theta'_{\mathrm{p}})}} \tag{7.17}$$

7.3.3 A Volumewise MML Estimator

Using C_{uni}, we can code the data by first stating a $\ddot{\theta}_i \in \ddot{\Gamma}_N$ using $\log N$ bits, and then stating x^n using $-\log P(x^n|\ddot{\theta}_i)$ bits. Using Bayes' rule, this can equivalently be recast as determining the posterior probability of $\ddot{\theta}_i$ given data x^n, using the uniform discrete prior $W(\ddot{\theta}_i) = 1/N$. We denote this posterior probability by P_W:

$$P_W(\ddot{\theta}_i|x^n) \propto P(x^n|\ddot{\theta}_i). \tag{7.18}$$

In this probabilistic formulation, (7.16) tells us that the *single* value θ'_{p} which maximizes the expected value of $P_W(\ddot{\theta}_i|x^n)$ (where $\ddot{\theta}_i$ is the truncated version of θ'_{p}), is given by the ML estimate. However, it tells us nothing about the width of the maximum attained at θ'_{p}. As we shall see, its width may depend on the region in the parameter space where θ'_{p} lies. It may therefore be more interesting to choose a small (but non-zero) width s, and look for the interval in Γ of width s with the maximal posterior probability mass (or, equivalently, the shortest code length) according to (7.18).

To obtain this interval, let us adapt an idea introduced by Rissanen in [129] in a somewhat different context, and look at the MML two-part code in another manner. We partition the parameter space Γ into a set of adjacent regions $R_1, \ldots, R_M$, each of width s, where s is such that $M \ll N$. Let us now determine the region R_i with maximum posterior probability mass $P_W(R_i|x^n)$. We first associate with each region R_i the element θ_i that lies in the center of R_i, so $R_i = [\theta_i - s/2, \theta_i + s/2]$. We can now extend the density $P(x^n|\theta_i)$, determined by a single value θ_i, to a probability determined by a region in the parameter space R_i in the obvious way by defining $P(x^n|R_i) = \int_{R_i} P(x^n|\theta)\pi(\theta)d\theta$, where π is an arbitrary proper prior with support R_i. In the limit for small s, we have $P(x^n|R_i) = P(x^n|\theta_i)$. This implies

$$P_W(R_i|x^n) \propto P(x^n|R_i)W(R_i) \quad \text{(if } s \text{ is small)}, \tag{7.19}$$

where W is the uniform prior. Marginalizing over the values $\ddot{\theta} \in \ddot{\Gamma}_N$ contained in R_i, we find that

$$W(R_i) = \frac{|R_i \cap \ddot{\Gamma}_N|}{N}. \tag{7.20}$$

In the limit for large N, we may select s as small as we want so that for all regions R_i, we can regard $P(x^n|\theta)$ and $\sqrt{I_n(\theta)}$ as approximately constant for all θ within the single interval R_i. For the optimal precision quantum $q(\theta)$, we have by Equation (7.17) that $q^2(\theta) \propto I_n(\theta)^{-1}$. Basing $\ddot{\Gamma}$ on this optimal precision quantum, it follows that in the limit for large N the density of parameter values $\ddot{\theta}$ in region R_i becomes proportional to $\sqrt{I_n(\theta_i)}$:

$$|R_i \cap \ddot{\Gamma}_N| \approx s\sqrt{I_n(\theta_i)} \cdot c \quad \text{for } N \text{ large}, \tag{7.21}$$

where c is a constant not depending on i. In the limit for large N, we have from (7.20) and (7.21) that

$$W(R_i) \propto s\sqrt{I_n(\theta_i)}. \tag{7.22}$$

We now conclude from (7.19), together with the fact that s does not depend on i, that

$$P_W(R_i|x^n) \propto P(x^n|\theta_i)\sqrt{I_n(\theta_i)}.$$

The region R_i which maximizes this is the most probable posterior region if the uniform prior is chosen. Assuming that I_n is a continuous function of θ, the $\theta \in \Gamma$ yielding the shortest expected code length *in its neighborhood* is thus given by

$$\theta'_{\mathrm{v}} = \arg\max_{\theta \in \Gamma}\{P(x^n|\theta)\sqrt{I_n(\theta)}\}. \tag{7.23}$$

We call this estimator the *revised volumewise MML estimator*. We see that in our alternative MML derivation, choosing the parameter value with the highest probability content in its neighborhood (7.23) gives us an estimate which maximizes the likelihood *times* $\sqrt{I_n(\theta)}$. This is in sharp contrast with the original MML estimate (7.10) which maximizes the likelihood *divided* by $\sqrt{I_n(\theta)}$! There is a caveat here; see the Discussion below. For simplicity, our derivations above have been only for the case where $\Gamma \subset \mathbf{R}^1$. The generalization to the multiple-parameter case is completely straightforward: it suffices to replace the intervals R_i of width s by *rectangles* R_i of *volume* s. In this case, the square root of the Fisher information $\sqrt{I_n(\theta)}$ becomes $\sqrt{|I_n(\theta)|}$, the square root of the determinant of the Fisher information matrix.

7.3.4 Discussion

Let us summarize what our foregoing derivations show and what they do not show. We *did* show in Section 7.3.2 that in general there is no reason to do as Wallace and Freeman and code parameter values using the code C_{wf} with expected lengths $-\log q(\theta') - \log w(\theta')$: another way of coding the parameter values may be better.

We provided an alternative coding scheme, but we did *not* show that it is in general better than Wallace and Freeman's code: we only showed that it leads to optimal estimators under *worst-case* assumptions. We should mention that there is a weakness[1] in the derivation of θ'_{v} as given by (7.23): in the derivation, we assumed that the optimal precision quantum was given by (7.17). However, (7.17) was derived in the derivation of the pointwise, not the volumewise MML estimator. It is not clear whether this remains the optimal precision quantum for the volumewise analysis. Hence, θ'_{v} has been proven to be 'volumewise optimal' *only* under the condition that the precision quantum (7.17) is used.

[1] Thanks to Kenji Yamanishi for pointing this out.

7.3.5 A Bold Assumption

We have derived our revised MML estimators θ'_{p} and θ'_{v} only for the special case of the uniform prior $w(\theta) = c$. In the following we shall simply *assume* that the derivations can also be done for the case of non-uniform priors. The definitions of θ'_{p} and θ'_{v} are therefore extended in the obvious way:

$$\theta'_{\mathrm{p}} = \arg\max_{\theta \in \Gamma} P(x^n|\theta)w(\theta) \tag{7.24}$$

$$\theta'_{\mathrm{v}} = \arg\max_{\theta \in \Gamma} \{P(x^n|\theta)w(\theta)\sqrt{I_n(\theta)}\}. \tag{7.25}$$

We should note that we have not proven the validity of these two formulas. However, all priors we shall use in our experiments are ESS-priors (see Section 7.5.2). As we shall see, using ESS-priors for data D is *always* equivalent to using the uniform prior for an extended data set $D \cup D^+$ (hence the ESS-priors embody some 'virtual data' on top of the actually given data). This gives an extra indication that the formulas (7.24) and (7.25) are theoretically valid for use in our experiments. Nevertheless, this is a very weak spot in our analysis, to which we shall return in the Conclusion.

7.4 Refined MDL Two-part Codes and their Relation to the Three MML Estimators

7.4.1 1996 MDL Two-part Codes

In Chapter 2, Section 2.1 we showed that the 2-part code $L_{2\text{-}p}$ is redundant. We used this fact as a motivation for introducing the stochastic complexity code as the code with the minimal worst-case regret (see Definition 2.8). The stochastic complexity code is 'one-part': no single parameter value is encoded explicitly. In his 1996 paper Rissanen [129] showed that one can revise the two-part code in such a way that the inherent redundancy gets removed, but the coding is still two-stage: first a parameter value is encoded, then the data is encoded with the help of that parameter. While the resulting code still has larger regret than the stochastic complexity code C_{sc} for all finite n, in the limit for large n the code lengths coincide.

We give a brief sketch of Rissanen's argument. For simplicity we assume that $\ddot{\Gamma}$ contains a finite number of parameter values, say N. We can thus write $\ddot{\Gamma} = \ddot{\Gamma}_N = \{\ddot{\theta}_1, \ldots \ddot{\theta}_N\}$. We also assume uniform code lengths for the models: $L_{C_1}(\ddot{\theta}_i) = \log N$ for all $\ddot{\theta}_i \in \ddot{\Gamma}_N$. Rissanen observed that a decoder, after having decoded $\ddot{\theta}_i$, already knows something about the data x^n whose description will follow. Namely, she knows that x^n must be a member of a proper subset $\mathbf{D}_i$ of the set of all possible data $\mathbf{E}^n$. This $\mathbf{D}_i$ is the set of all data x^n for which $\ddot{\theta}_i$ gives the shortest two-part code length:

$$\begin{aligned}
\mathbf{D}_i &= \{x^n \in \mathbf{E}^n \mid \ddot{\theta}_i = \arg\min_{\theta \in \ddot{\Gamma}_N}\{L_{C_1}(\theta) - \log P(x^n|\theta)\}\} \\
&= \{x^n \in \mathbf{E}^n \mid \ddot{\theta}_i = \arg\max_{\theta \in \ddot{\Gamma}_N}\{P(x^n|\theta)\}\}.
\end{aligned}$$

The reason for the decoder knowing that $x^n \in \mathbf{D}_i$ after decoding $\ddot{\theta}_i$ is the following: looking at Equation (7.1), we see that if $x^n \notin \mathbf{D}_i$, then the decoder would not have decoded $\ddot{\theta}_i$, but rather some other $\ddot{\theta}_j \neq \ddot{\theta}_i$. This fact can be exploited to change the code $C_2(\cdot|\theta)$ that was used in the original two-part code (7.1), to a code $C_2'(\cdot|\theta)$ with strictly shorter lengths. Using $C'_{2,\theta}$, we code x^n not by the code corresponding to probability distribution $P(x^n|\ddot{\theta}_i)$, but rather by the code based on the normalized probability distribution

$$\frac{P(x^n|\ddot{\theta}_i)}{\sum_{x^n \in \mathbf{D}_i} P(x^n|\ddot{\theta}_i)}.$$

In this case the total description length becomes

$$\log N - \log P(x^n|\ddot{\theta}_i) + \log \sum_{x^n \in \mathbf{D}_i} P(x^n|\ddot{\theta}_i)$$

rather than just $\log N - \log P(x^n|\ddot{\theta}_i)$ bits. In general, $\sum_{x^n \in \mathbf{D}_i} P(x^n|\ddot{\theta}_i) < 1$, which implies that the revised two-part code has a strictly shorter code length than the original one.

Rissanen showed [129] that the normalization trick described above can be optimally exploited (for large N) if, for every $\theta \in \Gamma$, the density of parameter values in $\ddot{\Gamma}_N$ in the neighborhood of θ is proportional to $\sqrt{|I(\theta)|}$, where $|I(\theta)|$ is the determinant of the Fisher information matrix $I(\theta) = I_1(\theta)$ as given by Equation (7.9) . This means that either the spacing between any two adjacent values $\ddot{\theta}_i$ and $\ddot{\theta}_{i+1}$ in $\ddot{\Gamma}_N$ should be made proportional to $1/\sqrt{|I(\theta)|}$, or the code giving code lengths $\log N$ to every $\ddot{\theta}_i \in \ddot{\Gamma}_N$ should be changed. Rissanen chooses the second option, but explicitly mentions that the first one is possible too [129, page 43].

7.4.2 Relation between the Four Approaches

We mentioned in the introduction that it has been – wrongly – claimed that Rissanen's considerations in [129] are already implicit in Wallace and Freeman's work on MML estimators. We now discern a possible reason for this confusion: according to Wallace and Freeman, the optimal precision quantum (width between adjacent parameter values in $\ddot{\Gamma}$) for encoding parameter values in the neighborhood of a point $\theta' \in \Gamma$ is given by $q(\theta') \propto 1/\sqrt{|I_n(\theta')|}$ (Equation 7.17). For i.i.d. model classes, $|I_n(\theta)| = n|I_1(\theta)|$ for all $\theta \in \Gamma$ [17]. This implies that for such model classes, both according to Rissanen (see above) and according to Wallace and Freeman one should pick the width between two adjacent parameter values proportional to $1/\sqrt{|I_n(\theta')|}$. In this sense they seem to reach the same conclusion. However, this same width was chosen for a very different reason. What is more, as we shall see in Sections 7.5 and 7.6, making predictions of future data on the basis of Wallace and Freeman's (1987) MML-estimators can be quite different from making predictions on the basis of Rissanen's 1996 refinement of MDL.

To understand how the same optimal precision can lead to different estimators and predictions, it is instructive to study the graph of $\sqrt{|I(\theta)|}$. As an example, in Figure 7.1 we depict this graph for the case of the Bernoulli model class. Observe that the optimal

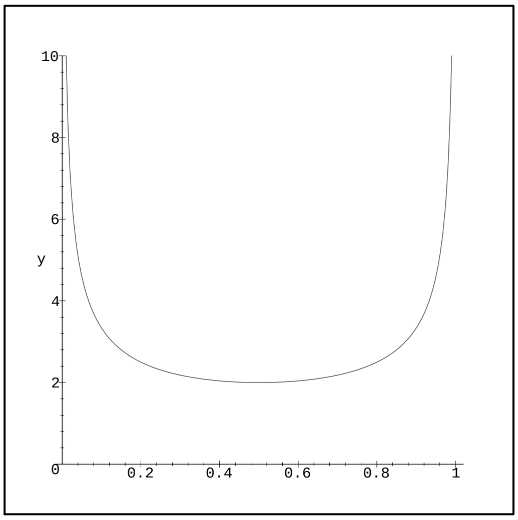


Figure 7.1: $\sqrt{|I(\theta)|}$ as a function of θ for the Bernoulli model defined by $P(X=1)=\theta$. This graph can be interpreted as giving for each θ the optimal density of quantized parameter values $\ddot{\theta}$ in the region around θ.

width between two truncated parameter values at point θ is inversely proportional to $\sqrt{|I(\theta')|}$. Therefore, the optimal *density* of truncated parameter values near θ is proportional to $\sqrt{|I_n(\theta')|}$. The graph of $\sqrt{|I(\theta)|}$ as a function of θ can therefore be interpreted as giving, for each θ, the density of the parameters in $\ddot{\Gamma}_N$ in the region around θ. In Wallace and Freeman's derivation of MML estimators, a 'subjective' prior density w defined for all θ is implicitly transformed into a probability mass function W defined for all $\ddot{\theta} \in \ddot{\Gamma}_N$. In our analysis we restrict ourselves to the situation where the subjective prior density w is chosen to be uniform (which is the only situation for which we have actually proven anything). In this case, in Wallace and Freeman's original derivation, it is implicitly assumed that regions R in the parameter space of equal volume should obtain equal prior probability mass $\sum_{\ddot{\theta} \in R \cap \ddot{\Gamma}_N} W(\ddot{\theta})$. If we choose $\ddot{\Gamma}_N$ with the optimal widths between the parameter values, then in regions with high $\sqrt{|I(\theta)|}$, there will be many truncated values $\ddot{\theta} \in \ddot{\Gamma}_N$. To obtain equal prior densities in regions of equal volume, the prior probability mass assigned to a *single* $\ddot{\theta} \in \ddot{\Gamma}_N$ must be made *inversely* proportional to $\sqrt{|I(\theta)|}$.

In contrast, both in our volumewise MML-estimator (with uniform subjective prior) and in Rissanen's revised two-part code all *single* truncated parameter values $\ddot{\theta} \in \ddot{\Gamma}_N$ receive *equal* prior probability $1/N$. In this case, for large N, the graph of $\sqrt{|I(\theta)|}$ can also be interpreted as giving, for each θ, the prior density of the region around θ: the prior probability becomes proportional, instead of inversely proportional, to $\sqrt{|I(\theta)|}$.

7.5 MML and MDL in Practice

According to both the MML and MDL *Principles*, it should be the case that the more we compress the data at hand, the better we are able to predict future data coming from the same source. We therefore expect the following: θ'_{v} gives better predictions than θ'_{p}, which in turn should give better predictions than θ'_{wf}. From an MDL (not MML) point of view we additionally expect that prediction based on P_{jef}, the evidence distribution with Jeffreys' prior will work even better, since P_{jef} can be interpreted as a form of stochastic complexity, and hence leads to maximal compression of the data; see Chapter 6, Section 6.2.4.

However, there are two possible caveats: first, as discussed in Section 7.3.4, θ'_{v} and θ'_{p} will only give shorter codelengths than θ'_{wf} under worst-case assumptions regarding the value of θ'_{p}, and it is not clear whether these are justified. Second, for the i.i.d. model classes we have $|I_n(\theta)| = n|I_1(\theta)|$. This implies that the influence of the 'correction factors' $\sqrt{|I_n(\theta')|}$ and $1/\sqrt{|I_n(\theta')|}$ in equations 7.23 and 7.10, respectively, becomes negligible as n grows to infinity. Therefore, asymptotically predictions based on θ'_{wf}, θ'_{p} and θ'_{v} all coincide with the Bayesian MAP prediction, which in turn asymptotically coincides with Bayesian evidence prediction for all reasonable priors w; see Chapter 6, Section 6.5.1. Consequently, the differences between the three MML estimators and the MDL (stochastic complexity) approach are relevant only for small sample sizes. Unfortunately, since both Rissanen's [129] and our derivations are asymptotic in nature, they do not say very much about this situation. It is therefore an interesting empirical question, whether either Rissanen's MDL approach or our revised MML estimators lead to a more accurate predictive distribution than the WF estimator in cases where only a limited amount of data is available. Below we describe how the four methods considered can be used to arrive at predictive distributions. In the next section, we study the predictive performance of these different predictive distributions empirically by using small, real-world datasets.

7.5.1 Predictive Distributions based on MML and MDL

Henceforth we assume the setting of Chapter 6, Section 6.2.1: our outcome space $\mathbf{E}$ can be written $\mathbf{E} = \mathbf{E}_1 \times \ldots \times \mathbf{E}_n$ and outcomes are written as *vectors* $\vec{x} \in \mathbf{E}$. Recall from Chapter 6 that in prediction and classification problems for discrete data, we are given some training data D which we use to arrive at predictions of a single *test vector* $\vec{x}$. The predictions are done using some *predictive distribution* $\mathcal{P}$. We now define the predictive distributions based on the three different MML estimators:

$$\mathcal{P}_{\text{mml-wf}}(\vec{x}|D) := P(\vec{x}|\theta'_{\mathrm{wf}}(D)) \tag{7.26}$$
$$\mathcal{P}_{\text{mml-p}}(\vec{x}|D) := P(\vec{x}|\theta'_{\mathrm{p}}(D)) = \mathcal{P}_{map}(\vec{x}|D) \tag{7.27}$$
$$\mathcal{P}_{\text{mml-v}}(\vec{x}|D) := P(\vec{x}|\theta'_{\mathrm{v}}(D)) \tag{7.28}$$

The MDL-based predictive distribution $\mathcal{P}_{\mathrm{mdl}}$ is defined to be the evidence predictive distribution with the prior set to Jeffreys' prior $\pi(\theta) \propto \sqrt{|I(\theta)|}$. In Chapter 6, Section 6.2.4, Equation 6.12 this distribution was introduced under the name $\mathcal{P}_{jef}$. Hence:

$$\mathcal{P}_{\mathrm{mdl}}(\vec{x}|D) := \mathcal{P}_{jef}(\vec{x}|D) = \frac{P_{jef}(\vec{x}, D)}{P_{jef}(D)} \tag{7.29}$$

Keeping D fixed and regarding $\mathcal{P}_{\mathrm{mdl}}(\cdot|D)$ as a function of $\vec{x}$, the denominator in (7.29) becomes a constant and we see that

$$\mathcal{P}_{\mathrm{mdl}}(\vec{x}|D) \propto \int P(\vec{x}, D|\theta)\pi(\theta)d\theta \tag{7.30}$$

It is interesting to contrast this with the WF MML estimator and the volumewise MML estimator when a uniform subjective prior w is used. In that case:

$$\theta'_{\mathrm{v}}(D) = \arg\max_{\theta} \{P(D|\theta)\pi(\theta)\} \quad \text{(if } w \text{ uniform)} \tag{7.31}$$

$$\theta'_{\mathrm{wf}}(D) = \arg\max_{\theta} \{P(D|\theta)\frac{1}{\pi(\theta)}\} \quad \text{(if } w \text{ uniform)} \tag{7.32}$$

and we see that, while the MDL predictions are based on the Bayesian evidence-predictions with Jeffreys' prior, the volumewise MML predictions are equivalent the Bayesian MAP-predictions with Jeffreys' prior and the WF MML predictions are based on the MAP-predictions with an inverse Jeffreys' prior.

7.5.2 The Model Class and the Subjective Prior

We wanted to apply our predictive distributions for a model class $\mathcal{M}$ sophisticated enough to yield good predictive performance in principle, while simple enough to allow the predictive distributions to be computed efficiently. We decided to use the Naive Bayes model class which has exactly these properties. We introduced Naive Bayes in the previous chapter (page 146) where (Equation 6.32) we also showed how to compute $\sqrt{|I(\theta)|}$ for this class. We repeat the formula for Jeffreys' prior $\pi(\theta)$ and for $\sqrt{|I(\theta)|}$ where θ is a member of the class of Naive Bayes models for sample space $\mathbf{E} = \mathbf{E}_1 \times \ldots \times \mathbf{E}_m$; for details about the notation we refer to the previous chapter):

$$\pi(\theta) \propto \sqrt{|I(\theta)|} = \prod_{k=1}^{K} (\theta_k^m)^{\frac{1}{2}(\sum_{i=1}^{m-1}(k_i-1)-1)} \prod_{i=1}^{m-1} \prod_{k'=1}^{K} \prod_{l=1}^{k_i} (\theta^i_{l|k'})^{-\frac{1}{2}} \tag{7.33}$$

From Equation (7.29) we see $\mathcal{P}_{\mathrm{mdl}} = \mathcal{P}_{jef}$. This implies that the instantiation of $\mathcal{P}_{jef}$ for the Naive Bayes case has been calculated already in Chapter 6. It is given by Equation (6.22).

The MML predictive distributions depend on a subjective prior. As candidate priors we used the so-called *ESS (Equivalent Sample Size) priors* [70]. ESS priors have a clear interpretation in terms of prior knowledge: ESS priors are indexed by a *virtual data set* D'. The ESS(D')-prior is defined such that, for every data set D and every test vector $\vec{x}$, the evidence prediction $P_{av}(\vec{x}|D)$ with prior ESS(D') is identical to the maximum likelihood prediction $P(\vec{x}|\hat{\theta}(D \cup D'))$. In other words, the ESS(D')-prior can be regarded as a 'summary' of data D': if, in a previous experiment, data D' was observed, then the corresponding prior can be used to combine this with the data D from the present experiment. Here we consider ESS(D') only for the case where D' contains all possible $\vec{x} \in \mathbf{E}$ equally often, say f times. We will allow fractional f. It is straightforward to show the following proposition; we omit the proof. In the proposition we freely use the notation introduced in Chapter 6, Sections 6.3 and 6.5.1.

Proposition 7.1 *Let $\mathcal{M}_{\mathcal{G}}$ be a Naive Bayes model class for random variables $X_1, \ldots, X_m$. Let*

$$a = \prod_{i=1}^{m-1} k_i \quad \textit{and} \quad b_i = \frac{\prod_{j=1}^{m-1} k_j}{k_i}.$$

The ESS prior $w(\theta)$ corresponding to a sample containing each $x \in \mathbf{E}$ exactly f times is given by

$$w(\theta) \propto \prod_{k=1}^{K} (\theta_k^m)^{f \cdot a - 1} \prod_{i=1}^{m-1} \prod_{k'=1}^{K} \prod_{l=1}^{k_i} (\theta_{l|k'}^i)^{f \cdot b_i - 1} \tag{7.34}$$

Consequently, the ESS-prior densities we consider are proportional to a product of parameters, which means they are Dirichlet densities (Definition 6.5). From (7.33) we see that $\sqrt{|I(\theta)|}$ is also proportional to a product of the parameters. The factors by which the likelihood is multiplied in determining the three MML estimators are $w(\theta)(\sqrt{|I(\theta)|})^{-1}$ for θ'_{wf}, $w(\theta)$ for θ'_{p} and $w(\theta)\sqrt{|I(\theta)|})$ for θ'_{v} respectively. Equations (7.33) and (7.34) together imply that these factors can be written as a simple product of parameters. Therefore, they may themselves be treated as Dirichlet densities. This means that the predictive distributions $\mathcal{P}_{\text{mml-wf}}$, $\mathcal{P}_{\text{mml-p}}$ and $\mathcal{P}_{\text{mml-v}}$ may all be reinterpreted as MAP predictive distributions with Dirichlet priors. We can therefore directly apply the formula (6.21) (page 143) to compute all three MML predictive distributions.

Why we cannot use the uniform prior Let

$$\text{WF}(\theta) = P(x^n|\theta)w(\theta)(\sqrt{|I(\theta)|})^{-1}.$$

θ'_{wf} is equal to the θ maximizing WF(θ). Observe that WF(θ) can be written as a simple product of parameters. We see from (7.33) that the exponent of θ_k^m in $(\sqrt{|I(\theta)|})^{-1}$ may become negative up to an amount of $-\frac{1}{2}(\sum_{i=1}^{m-1}(k_i - 1) - 1)$. Therefore, even with reasonably large training set sizes, WF(θ) may still contain two factors $\theta_{k_1}^m$ and $\theta_{k_2}^m$ with a negative exponent. In such a case, WF(θ) does not have a maximum. To see this, fix all components of θ except $\theta_{k_1}^m$ and $\theta_{k_2}^m$. We can now view WF(θ) as a function of $\theta_{k_1}^m$ as follows: as $\theta_{k_1}^m$ is varied, all parameter values in θ remain constant except $\theta_{k_1}^m$ and $\theta_{k_2}^m$. The latter must be varied along to keep $\sum_i \theta_i^m = 1$. We see immediately that

$$\lim_{\theta_{k_1}^m \downarrow 0} \text{WF}(\theta) = \infty \text{ and } \lim_{\theta_{k_1}^m \uparrow G} \text{WF}(\theta) = \infty \text{ where } G = 1 - \sum_{i \neq k_1, k_2} \theta_i^m$$

while for $0 < \theta_{k_1}^m < G$, WF(θ) is a continuous function of $\theta_{k_1}^m$. It follows that the MML WF estimator θ'_{wf} is undefined.

In particular, this odd behaviour occurs if we pick as our subjective prior $w(\theta)$ the uniform prior. Therefore we decided to use in our experiments only ESS priors, where the fraction f was chosen to be the smallest possible number with which the above mentioned technical difficulty does not occur. Experiments with different ESS subjective priors seemed to produce similar results.

7.6 Empirical Results

For comparing empirically the four predictive distributions discussed in the previous section, we turned once more to the six public domain classification data sets that were already used in Chapter 6. These datasets are described in detail in Table 6.1 on page 147; see also page 125. We performed two sets of experiments.

Experiments with 0/1-Score

In the first set of experiments, we computed the crossvalidated 0/1-scores for each of the four methods by using 5-fold crossvalidation (following the testing scheme used in [50]). The definition of 0/1-score and n-fold crossvalidation can be found in Section 6.5.2 of the previous chapter. As the results appeared to be strongly dependent on the way the data was partitioned in the 5 folds to be used, we repeated the whole crossvalidation cycle 10000 times with different, randomly chosen partitionings of data. The results are given in Table 7.1. In the MML-WF case the predictive distribution $\mathcal{P}_{\text{mml-wf}}$ was used; MML-P stands for $\mathcal{P}_{\text{mml-p}}$, MML-V stands for $\mathcal{P}_{\text{mml-v}}$ and MDL stands for $\mathcal{P}_{\text{mdl}}$.

We can observe that, first of all, with respect to the 0/1-score, there seems to be

	MIN				MEAN				MAX			
	MML				MML				MML			
	WF	P	V	MDL	WF	P	V	MDL	WF	P	V	MDL
AU	83.5	**83.6**	83.5	**83.6**	**84.9**	**84.9**	84.8	**84.9**	**86.2**	86.1	86.1	**86.2**
DB	73.4	**73.6**	73.3	73.2	**75.5**	75.4	75.4	75.3	77.1	77.2	77.2	**77.3**
GL	56.5	56.5	56.1	**58.4**	62.6	62.6	62.7	**64.9**	67.3	68.2	67.3	**70.1**
HD	**81.9**	**81.9**	**81.9**	81.1	**84.5**	84.4	84.4	84.1	**87.0**	**87.0**	**87.0**	**87.0**
IR	**93.3**	92.7	92.7	92.0	**94.4**	94.3	94.3	**94.4**	96.0	96.0	**96.7**	**96.7**
LY	**79.1**	78.4	78.4	**79.1**	**84.2**	84.0	83.6	83.9	**88.5**	87.8	**88.5**	**88.5**

Table 7.1: Classification 0/1-scores with 10000 independent 5-fold crossvalidation runs.

no clear winner between the different predictive distributions used, and the differences between the results are usually small. Secondly, for all data sets used, the average results obtained here (i.e. the figures in the middle column) are very close to the optimal results for the Naive Bayes class that can be found in the literature (which are often arrived at using other inference methods). In two cases ('diabetes' and 'heart disease') our results seem to be optimal. For extensive references on results obtained by other methods we refer to [154] and [50]. Apparently, the minimum encoding approaches work very well independently of exactly which of the four versions is used.

Experiments with log-score

In the second set of experiments, instead of predicting only the value of the class variable X_m, we used the predictive distributions for computing the joint probability for the unseen testing vectors as a whole. In this case there is no useful analogue to the 0/1-score, so the accuracy of the methods was now measured by using the log-score, which is simply the logarithmic loss; see Chapter 2, page 44 for several interpretations of this loss function.

To prevent the large fluctuation in the results, we used in this experiment the leave-one-out form of crossvalidation, where the task is at each stage to predict one data vector, given all the others. The results of this experiment can be found in Table 7.2. From these results we can now see that the MDL approach produced consistently the best score, and of the MML estimators considered here, the MML-V estimator was more accurate than the MML-P estimator, which performed better than MML-WF (with the exception of the Lymphography database).

	10% training data				100% training data			
	MMLWF	MMLP	MMLV	MDL	MMLWF	MMLP	MMLV	MDL
AU	16.68	16.61	16.54	**14.98**	14.64	14.63	14.62	**14.44**
DB	13.80	13.79	13.77	**13.65**	13.25	13.25	13.24	**13.23**
GL	14.22	14.19	14.13	**12.14**	11.38	11.34	11.30	**10.25**
HD	12.99	12.95	12.90	**12.41**	11.75	11.74	11.73	**11.67**
IR	4.34	4.22	4.08	**3.60**	3.20	3.17	3.14	**3.07**
LY	19.27	19.27	19.25	**16.78**	15.85	15.89	15.90	**14.73**

Table 7.2: Leave-one-out crossvalidated log-scores in the joint probability estimation task. The lower the log-score, the better the predictive performance.

To study the small sample behavior of the methods in more detail, we rerun the leave-one-out crossvalidation experiments, but used at each stage only s (randomly chosen) vectors of the available $n-1$ vectors for producing the predictive distribution, where s varied between 1 and $n-1$. In this case, the results obtained are quite similar with all six datasets: the results with $s = 0.1n$ can be found in Table 7.2.

In figures 7.2-7.4 we plotted the behavior of the methods as a function of the amount s of training data. In these figures, the log-scores are scaled with respect to the score produced by the MMLWF method so that the MMLWF method gets always a score 0, and a positive score means that the actual log-score was better than the MMLWF log-score by the corresponding amount.

From figures 7.2-7.4 we now see two interesting things: firstly, the different predictive distributions seem to converge with increasing amount of training data, as was expected from the discussion in Section 7.5. Secondly, the relative differences between the methods seem to grow when the amount of available data is decreased. We see in particular that for *extremely* small sample sizes, the three MML approaches are comparable while the MDL approach may even be a bit worse. Then, for still very (but not extremely) small sample sizes all datasets show the same pattern: MDL outperforms MML-V, which in turn outperforms MML-P which in turn outperforms MML-WF. As the sample size grows, the ordering between the four approaches remains unchanged (except for the lymphography database) but the differences become smaller.

The fact that the results are consistent over datasets pertaining to six completely different domains suggests that the differences between the various approaches presented here may be practically significant in cases with small amount of data.

We note that the difference in performance between MDL and the three MML approaches is probably largely due to the fact that the MDL predictive distribution is based on 'using all models in the class at once' (which is achieved by averaging over the models) while the other three are based on prediction using a *single* model; we saw in the previous chapter that, with the uniform prior, there is, for very small sample sizes, much to be gained by predicting using the evidence and predicting using a single model (Section 6.5.3). Different experiments we reported elsewhere [87] suggest the same phenomenon. This means that the results reported here cannot be simply interpreted as telling us that 'MDL works better than MML'. They do suggest the following though: the influence of using Fisher information in (what we consider) the wrong way, as done in MML-WF, versus not using it at all (as in MML-P = MAP) vs. using it in the 'right' way (as done in MML-V) is really very small, even for small sample sizes. Small as it is though, it has fairly consistent effects (MML-V performing better than MML-P performing better than MML-WF) and is thus in agreement with our asymptotic theoretical analysis which depended on taking a worst-case point of view and assuming a uniform subjective prior.

Looking at figures 7.2-7.4 again, we hypothesize that for extremely small sample sizes, our asymptotic results simply give no evidence; as more data arrives, we enter a region where they become significant and the performance of the three MML methods is as predicted by the asymptotic theory. Then, as the sample size grows truly large, the law of large numbers 'takes over' and the differences between the three methods become negligible.

7.7 Conclusion and Future Work

We have shown that the claimed similarity between Wallace and Freeman's MML approach and Rissanen's 1996 MDL approach is superficial, and that when applying the approaches for predictive modeling, we arrive at quite different methods in practice. We pointed out a technical oversight in the derivation of the WF MML estimator and introduced two revised versions of the MML estimator, of which the volumewise optimal estimator was shown to be related to Rissanen's MDL estimator.

Theoretically, the revised MML estimators should allow for more compression of the data than the WF MML estimator, at least in a worst-case sense. As this was theoretically shown to be the case only asymptotically and only for uniform subjective priors, this raised the question of the practical small sample behavior of these methods. To be able to study this question empirically, we tested the different prediction methods for the Naive Bayes model class.

In the empirical tests performed, it was observed that while in simple classification tasks the methods showed quite similar performance, in joint probability distribution estimation the MDL approach produced consistently the best results. Moreover, the revised MML estimators introduced here usually gave better results than the WF MML estimator. The differences were largest for small amounts of training data (except for *extremely* small ones). They became smaller with an increasing amount of training data, as was expected from the theory.

Future Work In future work we hope to be able to show that our formulas for the two revised MML estimators are also valid for subjective non-uniform priors. Also, it would be interesting to consider alternative, more sophisticated ways of testing the small sample behaviour of the predictive distributions based on these estimators.

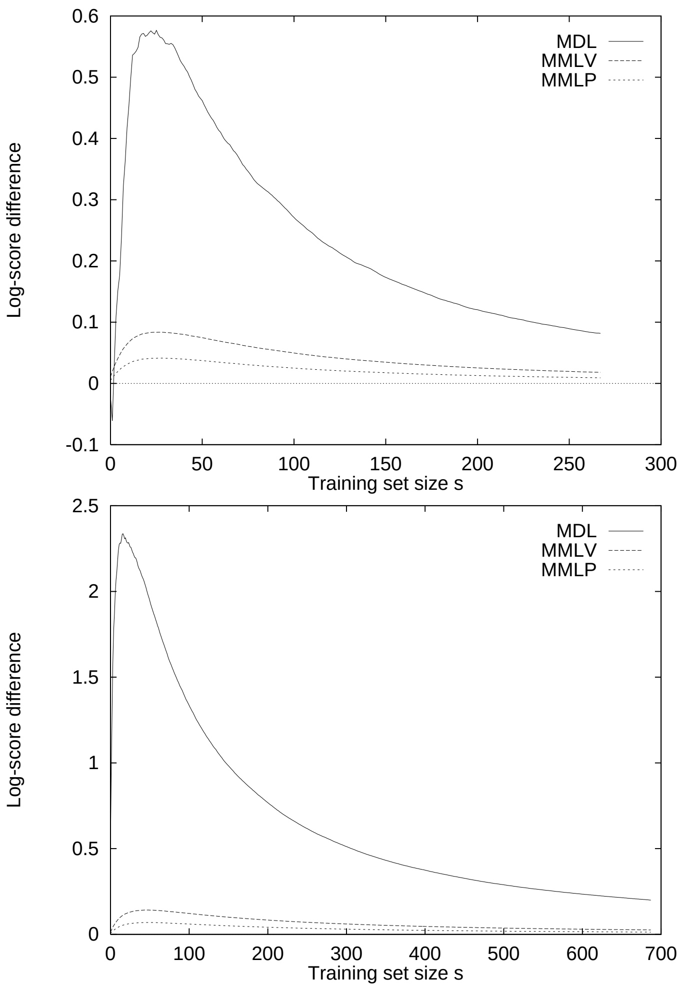


Figure 7.2: The upper picture shows the HD dataset leave-one-out crossvalidated results as a function of the training data available, scaled with respect to the MMLWF score. The lower picture shows the same function for the AU dataset. The higher the score, the better the prediction accuracy.

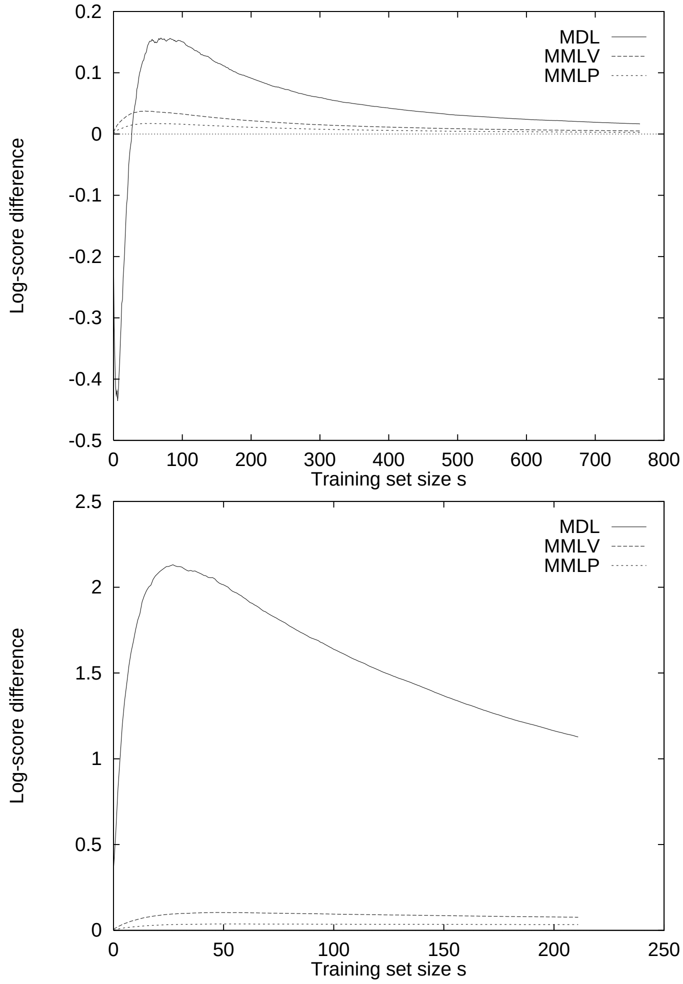
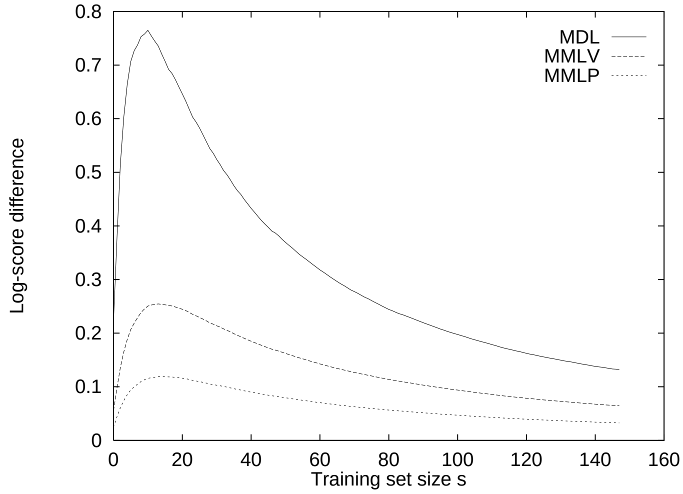


Figure 7.3: The upper picture shows the DB dataset leave-one-out crossvalidated results as a function of the training data available, scaled with respect to the MMLWF score. The lower picture shows the same function for the GL dataset. The higher the score, the better the prediction accuracy.



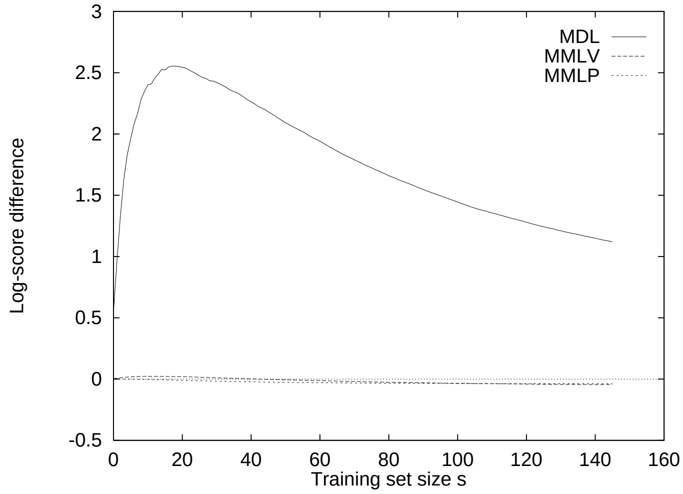


Figure 7.4: The upper picture shows the IR dataset leave-one-out crossvalidated results as a function of the training data available, scaled with respect to the MMLWF score. The lower picture shows the same function for the LY dataset. The higher the score, the better the prediction accuracy.

Part III

Reasoning under Uncertainty

Introduction to Part III

The previous parts of this thesis have been concerned with inductive inference. Inductive inference can be seen as an aspect of *reasoning under uncertainty*, the problem of how to make rational inferences and decisions in the presence of incomplete and unreliable information.

Default Reasoning

In part III of this thesis we explore other kinds of reasoning under uncertainty. We will mostly be concerned with *default reasoning*. Default reasoning is concerned with rules that typically (by *default*) hold but that may in some exceptional cases be violated. Below we give an informal introduction to default reasoning. In an epilogue to part III (page 269) we explore connections between default reasoning and MDL.

The prototypical example of default reasoning involves the rule ([56], page 2) 'Typically, birds fly'. If we are told that Tweety is a bird, and we are then asked whether it can fly or not, we would be inclined to say that it can. If, later on, we are told that Tweety is a penguin, we would like to retract our conclusion that it can fly and instead conclude that it cannot fly after all.

Common Sense Reasoning

It has always been one of the major goals of Artificial Intelligence to formalize *common sense reasoning*, the kind of every-day reasoning that we humans perform when making plans or decisions. An important part of common sense reasoning is really default reasoning; we give a little example to illustrate this general point. Imagine a situation in which your friend gives you a phone call and asks whether you can meet him in an hour. You reason as follows: the drive to your friend takes about 45 minutes; you have to drive to the gas station first which will take an additional 10 minutes. Therefore, you say that you will be able to make it.

In your reasoning, you did not take into account any of the following eventualities: your car keys may be lost; your car may be stolen or towed away; the gas station may be closed; the road may be blocked; you may get a traffic accident etc. If you would consider all preconditions that must hold in order for you to arrive in time, you will never be able to conclude that you actually will arrive on time. Yet in practice, you do jump to this conclusion. This means that even though you do not know whether your car has been stolen, you implicitly assume that it has not; even though you do not know whether the road is blocked, you assume it is not etc. These are all default assumptions: *typically* roads are not blocked, cars are not stolen etc., which leads you to assume that your own car has not been stolen, that you will not encounter a road block etc. Assumptions of this kind are ubiquitous in common sense reasoning.

The Frame Problem; Nonmonotonic Temporal Reasoning

The following default assumption is central to common sense reasoning: most properties of the world are *persistent*, that is, they usually do not change over time. The problem of how to properly formalize this 'law of persistence' or equivalently, *inertia*, is known as the 'frame problem' [107]. We have already encountered several instantiations of this law: in the reasoning described above, you implicitly assumed that your car would still be in the same location as where you last parked it; you implicitly assumed that your car keys would still be in the same location as where you last put them; you assumed that your car would still work etc.

Solving the frame problem, that is, formalizing the law of persistence, has turned out to be a surprisingly difficult task. It is one of the main challenges of the subfield of Artificial Intelligence that is known as 'nonmonotonic temporal reasoning' or 'common sense reasoning about action and change' [135]. In the next chapters we present a fairly general theory of nonmonotonic temporal reasoning.

Formalizing Nonmonotonic Reasoning: Nonmonotonic Logic

Default rules cannot be readily expressed in classical (propositional or first-order) logic. The reason is that classical logic is *monotonic*: as your set of knowledge grows, so does the set of conclusions that can be drawn from it. On the other hand, the inferences we made above are *nonmonotonic*; in the 'birds fly' example, if we are only told that Tweety is a bird, we conclude that it can fly, whereas if we are given *additional* knowledge (namely, that Tweety is a penguin) we would like to retract our previous conclusion.

This has led researchers to search for an extension of classical logic that allows for a correct treatment of default rules [106, 109, 124, 56, 21]. The resulting logics are called *nonmonotonic*. Most nonmonotonic logics can be viewed from a unifying perspective if one characterizes them in terms of their semantics [142]: they modify classical logic by introducing a *preference order* on models. In standard logic, when given a theory T, a sentence Γ is entailed by T iff Γ holds in all models of T. In nonmonotonic logic, the models for T are (roughly speaking) ordered according to some partial order and a sentence Γ is entailed by T iff it holds in the models that are the most preferred models in this order.

The order on models induced by such a 'preferential semantics' (a term due to Shoham [142]) is often determined by one or more *abnormality predicates*. We use the 'birds fly' example for illustration. In this case we need only one abnormality predicate Ab, where $Ab(x)$ stands for the fact that x is in some sense 'abnormal', that is, 'atypical' or 'unusual'. The rule 'typically birds fly' would be expressed as follows:

$$\forall x.\ [\ \mathit{Bird}(x) \wedge \neg Ab(x)\ ] \supset \mathit{Can\_Fly}(x) \tag{7.35}$$

($\supset$ standing for standard material implication[2]). In words, if x is a bird and x is not abnormal, then x can fly. Let us consider a logical theory consisting of (7.35) together with the fact that Tweety is a bird:

$$\mathit{Bird}(\mathit{Tweety}) \tag{7.36}$$

According to standard first-order logic, there exist models for this theory in which Tweety is not an abnormal bird and can therefore fly ($\neg Ab(\mathit{Tweety}) \wedge \mathit{Can\_Fly}(\mathit{Tweety})$)

[2] So $A \supset B$ is equivalent to $\neg A \vee B$.

while there also exist models in which Tweety is abnormal and cannot fly. The idea behind 'preferential semantics' is to order the models of the theory according to how 'normal' they are. More precisely, model $\mathcal{M}_1$ is preferred over $\mathcal{M}_2$ if and only if the interpretation of Ab in model $\mathcal{M}_1$ is a proper subset of the interpretation of Ab in model $\mathcal{M}_2$. The set of allowed conclusions is then restricted to those sentences that hold in all most-preferred models. In our example, there exist models $\mathcal{M}$ in which no abnormalities at all are the case ($\mathcal{M} \models \forall x \neg Ab(x)$). Such models will be preferred over all other models. Therefore, if $\mathcal{M}$ is a most preferred model, then $\mathcal{M} \models \neg Ab(\mathit{Tweety})$ and hence, by axiom (7.35), $\mathcal{M} \models \mathit{Can\_Fly}(\mathit{Tweety})$.

Does this work?

Can nonmonotonic logics be successfully applied for any but the simplest reasoning domains? The answer depends on one's point of view.

When nonmonotonic logic was introduced in the late 1970s, it was hoped that a large class of reasoning problems could be successfully handled by variations of the simple mechanism introduced above: first add a large amount of abnormality predicates $Ab_1, Ab_2, \ldots$ to one's language, then express all default rules by axioms of the form $a(x) \wedge \neg Ab_i(x) \supset b(x)$ and then choose the models for the theory with the smallest interpretations of $Ab_i(x)$. However, in the 1980s it gradually became clear that this simple idea of minimizing abnormalities only works for a very restricted class of reasoning domains. One of the most famous counterexamples is Hanks and McDermott's 'Yale Shooting Problem' (YSP) [67], which is concerned with the law of persistence ('things usually stay the same') introduced above. It shows that a straightforward formalization of this law using abnormalities will lead to counterintuitive results for all but the simplest reasoning domains. To make things worse, the YSP turned out to be just the first member of a long chain of problematic examples (this will be discussed at length in the next chapter). When taken together, these examples strongly suggest that the original goal of nonmonotonic reasoning - employing a single, simple extension to standard logic for handling a wide variety of default rules - is impossible to obtain. In this sense nonmonotonic logic turned out to be a failure (incidentally causing one of its originators, D. McDermott, to be so disappointed that he quit the field altogether [108]). On the other hand, if one views nonmonotonic formalisms merely as *technical tools* and uses them to extend classical logic in various different ways, not just by rules of the form (7.35), then they turn out to be reasonably successful, as the slow but steady progress in the field of nonmonotonic temporal reasoning shows [151, 12, 104].

Connection to Probability and MDL; what do defaults stand for?

From the subjectivist (Bayesian) view on probability theory, probabilities can be used to model 'degrees of belief' [77, 38]: if an agent (for example, a human or a robot) assigns a high probability to proposition A, this means he (it) has a high degree of belief in A. This leads to the identification of 'abnormalities' with 'events having small probability'. Whether probability theory should or should not be used to model default reasoning is still a matter of debate (see [25] for a review of the arguments for and against probability; see also [56, 106, 116]). The appropriateness of probability is related to the question of what exactly one is trying to model when one is using default rules. In one view, nonmonotonic reasoning systems should be used by (implemented in) 'agents' (for example, robots) which have some knowledge about the world (given by axioms) and which receive additional information through their sensors. One assumes that this information is preprocessed so that it can also be encoded as axioms. The agent can then use its nonmonotonic reasoning system to arrive at sensible conclusions from its given knowledge. In this view, 'abnormalities' can be quite literally seen as 'things that happen with small probability': if the robot sees that a car is in a parking lot at time t, it should be able to infer that it will still be there at time $t = t + 1$, since the probability that this will be the case is very high (at least if the unit of time is chosen appropriately). In this view, default rules should be seen as being analogous to laws of physics: they describe how the world works. The only difference to what we usually call the 'laws of physics' is that the description is at a different level of abstraction and deals with different phenomena.

According to another point of view, defaults may also stand for *conventions* [121, 1] that humans implicitly use in communication: if one reads a story in which a bird is mentioned and no clues are given as to whether or not it can fly, the reader will typically conclude that it can fly. If one regards defaults as conventions, then it is not so clear whether or not they should be treated probabilistically.

In a very influential book [116], J. Pearl argues that uncertainty should be handled probabilistically (even when one is modeling conventions rather than 'physical laws'), and he shows how many of the problems associated with a probabilistic interpretation of defaults can be overcome. However, he concedes that in practical applications, it may often be useful to work with nonmonotonic logic instead of probability theory. He then proposes to use nonmonotonic logic after all, but with a semantics that is consistent with probability theory in the sense that it can be given a probabilistic interpretation.

In our own work on nonmonotonic temporal reasoning, to be presented in chapters 9 and 10, we implicitly take a similar (yet different) view: though all nonmonotonicity will be dealt with in entirely non-probabilistic terms, most of it can in principle be interpreted probabilistically. The reason that a probabilistic interpretation is not available for *all* our uses of nonmonotonicity will be explained in the Epilogue to part III of this thesis. There we shall argue that when nonmonotonic logic is used to model *defaults*, it should indeed have a probabilistic interpretation. However, we shall also argue that in applied work, nonmonotonicity is often used only to arrive at concise descriptions of a domain; in such cases, a probabilistic semantics is inappropriate. In the Epilogue we shall review our own theory of chapters 9 and 10 and show that indeed, all *defaults* used there can be interpreted probabilistically. We shall also demonstrate that the connection between defaults and probability arises most naturally if probabilities are treated from an MDL perspective, thereby connecting part III to parts I and II of this thesis.

Chapter 8

Introduction to Nonmonotonic Temporal Reasoning

8.1 Introduction

In this chapter we give a short overview of the field of Nonmonotonic Temporal Reasoning (NMTR). We will discuss the central problems, the main goals and the methodology of the field. The main purpose is to provide the necessary background for the following two chapters 9 and 10, in which we will develop our own theory of NMTR.

We start (Section 8.2) by introducing the field by means of an historical overview. In Section 8.3 we give a critical discussion of the current research methodology and the main research goals of the field. In Section 8.4 we discuss our own view on these issues and indicate how some of the problems raised in Section 8.3 can be overcome. We end with a summary of challenges faced by the field. For more details about both the history and the problems of the field we refer to Sandewall and Shoham's overview article [135] and Shanahan's recent book [139].

8.2 Introduction to and History of the Field

Nonmonotonic Temporal Reasoning, also called *common-sense reasoning about action and change* is a subfield of Artificial Intelligence (AI). The original goal of this field was to come up with a logic that allows for common-sense reasoning about action and change. This would be part of the grander goal of developing a general 'logic of common-sense reasoning', the central goal in the logicist view of AI [108]; see also Section 8.3. Before we start our historical overview of the field, we introduce a formalism for representing statements about action and change as sentences in first-order logic.

The Language We use a many-sorted first-order language $\mathcal{L}$. $\mathcal{L}$ contains three sorts: *fluents* (variables of the sort denoted by f), *events* (e) and *time points* (t). Fluents and events correspond to the basic entities of our domains. Fluents are those properties of a domain which usually do not change over time. Fluents can *hold* ('be the case') or not at any point in time t. If fluent f holds at time t we write $Ho(f,t)$. Actions or *events* influence the (truth) value of fluents[1]. If an event e takes place between time t and $t+1$, we will also say that e 'holds' at time t and write $Ho(e,t)$. Time-points are identified with the integers. Apart from Ho, $\mathcal{L}$ contains the following functions and predicates: '+', '=' and Ab. '+' and '=' will receive their usual interpretation. Ab is defined over fluent-time pairs. $Ab(f,t)$ denotes that there is something 'abnormal' about fluent f at time t. We will assume all formulas to be implicitly universally quantified.

The Frame Problem The field got started in 1969 when McCarthy and Hayes [107] first formulated the *frame problem.* This is the problem of how to properly formalize the common-sense law of *inertia* or *persistence*, which says that most properties of the world do not change over time, unless there is some action that affects them. This notion turns out to be surprisingly difficult to model.

As an example of what we mean by 'persistence', consider a very simple reasoning domain consisting only of a person who can either be walking or not (denoted by fluent *Walking*) and who can be alive or not (denoted by fluent *Alive*). Only one action may take place in the domain, which is to sit down (denoted by event *Sit_down*). The effect of sitting down is that the person is not *Walking* any more. This can be formalized as

$$Ho(\mathit{Sit\_down},t) \supset \neg Ho(\mathit{Walking},t+1) \tag{8.1}$$

Such an axiom is usually called an *effect axiom.* Let it further be given that (1) the person is alive and walking at $t=0$, and (2) the person takes a seat at time $t=2$; no other events take place. This is formalized as

$$Ho(\mathit{Walking},0) \wedge Ho(\mathit{Alive},0) \tag{8.2}$$
$$Ho(\mathit{Sit\_down},2) \wedge (t \neq 2 \supset \neg\exists e.Ho(e,t)) \tag{8.3}$$

These axioms are usually called *observation axioms.*

We want our domain to be subject to persistence. Hence we would like to be able to infer that the person remains *Walking* until $t=2$, remains $\neg$*Walking* thereafter and remains *Alive* throughout. Clearly, this cannot be inferred from axioms (8.1)–(8.3) alone. We could enforce persistence by adding the following so-called *frame axioms*:

$$Ho(\mathit{Alive},t) \equiv Ho(\mathit{Alive},t+1) \tag{8.4}$$
$$\neg Ho(\mathit{Sit\_down},t) \supset [Ho(\mathit{Walking},t) \equiv Ho(\mathit{Walking},t+1)] \tag{8.5}$$

However, handling persistence in this way leads to theories that are unwieldy and brittle (see [139], page 10): *unwieldy* in the sense that for each fluent, we have to

[1] In the simple domains we will encounter in this chapter we need not make any distinction between 'actions' and 'events'.

specify the complete list of actions that do *not* interfere with its persistence. In the present case this was achieved by the simple axioms (8.4) and (8.5), but when we move on to domains consisting of thousands of actions and fluents, it will make our theories inconveniently large. Our theories would also be *brittle* in two ways: first, if we want to add knowledge to a pre-existing theory, we may have to change most axioms (rather than just adding new ones). Second, our theories would become *inconsistent* if the observations would contradict persistence – as will sometimes happen in the real world (see Example 8.4, page 191). If a theory is inconsistent then all reasoning breaks down; we would prefer a system that concludes that something unexpected has happened and keeps making rational inferences about other aspects of a domain.

It seems a better idea to express the law of persistence by a single mechanism, to which only axioms of the form (8.1) and (8.3) but *not* of the form (8.4)-(8.5) have to be supplied. The obvious idea is to use a simple form of nonmonotonic reasoning (page 184) that allows one to infer that fluents do not change value whenever there is no evidence to the contrary. In terms of a preferred model semantics, this amounts to selecting only those models for a theory in which the smallest number of changes take place. However, in their classic 1986 paper, Hanks and McDermott introduced the Yale Shooting Problem (YSP) which essentially shows that this idea fails for all but the simplest reasoning domains. In order to discuss the YSP, we first need to extend our formalism so that it can handle nonmonotonicity.

Models and Preferred Models A structure $\mathcal{M}$ for our language $\mathcal{L}$ consists of universes for all the sorts and interpretations for all function and predicate constants in $\mathcal{L}$. For a function or predicate constant K, we write $\mathcal{M}[K]$ to denote the interpretation of K in $\mathcal{M}$ (which is then a function or a set, respectively). A model of a set of sentences Γ is any structure $\mathcal{M}$ such that Γ is true in $\mathcal{M}$, where truth of sets of sentences is defined as usual. If $\mathcal{M}$ is a model of Γ, we write $\mathcal{M} \models \Gamma$. To keep things simple we will only consider models in which all time points are interpreted as integers and all fluent and event constants as themselves. Moreover, we will assume that only one event happens at a time (the latter restriction will be dropped in later chapters). Our nonmonotonic formalism is simply to prefer the models of a theory T with the fewest (in the subset sense) abnormalities. Summarizing:

Definition 8.1 *A* candidate model $\mathcal{M}$ *for a theory* T *is a model satisfying (1)* $\mathcal{M} \models T$, *(2)* $\mathcal{M}$ *interprets time points as the integers and all event and fluent constants as themselves, and (3) (*$\exists!$ *denoting 'there exists exactly one):*

$$\mathcal{M} \models \forall t \exists! e.\ Ho(e, t)$$

Definition 8.2 *A model* $\mathcal{M}$ *is a* preferred model *for theory* T *iff* $\mathcal{M}$ *is a candidate model for* T *and there is no other candidate model* $\mathcal{M}'$ *for* T *with* $\mathcal{M}'[Ab] \subsetneq \mathcal{M}[Ab]$.

In what follows, we assume the set of fluent and event constants to contain just those elements that are mentioned in the relevant axioms. When we write '$\mathcal{M}'$ has fewer abnormalities than $\mathcal{M}$' we mean 'fewer' in the subset sense, i.e. $\mathcal{M}'[Ab] \subsetneq \mathcal{M}[Ab]$.

8.2.1 The Yale Shooting Problem

Using the formalism just introduced, a straightforward formalization of the law of persistence would look as follows:

$$\forall f, t \; . \; \neg Ab(f,t) \supset [Ho(f,t) \equiv Ho(f,t+1)] \tag{8.6}$$

In words, (8.6) says that if property f holds at time t and there is nothing abnormal about f at time t, then it will still hold at time $t+1$. Combining this with axioms (8.1)-(8.3) and using Definition 8.2 to pick the preferred models, we essentially choose the models with the fewest possible changes. We will call the combination of Definition 8.2 with (8.6) the *naive persistence formalization.*

Example 8.3 (Yale Shooting Problem) Assume again there is a person (in this context typically named Fred) that can be either *Alive* or not. Furthermore someone points a gun at Fred that can either be *Loaded* or not. There are three possible actions in the domain: *Load*, *Wait* and *Shoot*. A *Load* action loads the gun; a *Wait* action has no effects whatsoever; a *Shoot* action kills Fred if it is performed when the gun is loaded. We are given the following information: initially, Fred is alive and the gun is unloaded. At $t = 0$, the gun is loaded. At $t = 1$, nothing happens (a *Wait* action takes place) and at $t = 2$ Fred is shot at. We will assume our domain to be subject to persistence; this means that intuitively, the gun remains loaded during the *Wait* action; therefore, Fred is shot at with a loaded gun and will therefore not be alive any more at time $t = 3$.

The following theory $T_{\text{H-M}}$ is a straightforward formalization of our domain. $T_{\text{H-M}}$ is the union of Axiom (8.6) (persistence) together with axioms (8.7)–(8.10) below.

$$Ho(Load, t) \supset Ho(Loaded, t+1) \tag{8.7}$$

$$Ho(Loaded, t) \wedge Ho(Shoot, t) \supset \neg Ho(Alive, t+1) \tag{8.8}$$

$$Ho(Alive, 0) \wedge \neg Ho(Loaded, 0) \tag{8.9}$$

$$Ho(Load, 0) \wedge Ho(Wait, 1) \wedge Ho(Shoot, 2) \tag{8.10}$$

In their YSP paper Hanks and McDermott showed that the naive persistence formalization does not handle this domain correctly: the set of *preferred* models does not coincide with the set of *intended* models . By an 'intended' model we mean a model that is intuitively admissible. Intuitively we would conclude that Fred is not alive any more at $t = 3$. Hence for all intended models $\mathcal{M}$ for $T_{\text{H-M}}$ we have $\mathcal{M} \models \neg Ho(Alive, 3)$. There indeed exists a preferred model $\mathcal{M}_1$ for which this is the case. But there is also another preferred model $\mathcal{M}_2$ with $\mathcal{M}_2 \models Ho(Alive, 3)$. In the model $\mathcal{M}_1$ everything happens as we would intuitively expect. It contains two abnormalities, $Ab(Loaded, 0)$ (the gun becomes *Loaded* at time $t = 0$) and $Ab(Alive, 2)$ (Fred dying at $t = 2$). $\mathcal{M}_2$ also contains two abnormalities: $Ab(Loaded, 0)$ and additionally $Ab(Loaded, 1)$: the gun mysteriously becomes unloaded again during the waiting. Therefore, Axiom (8.8) does not apply and Fred remains alive at $t = 3$. Notice that there is nothing in our axioms which is in contradiction with the spurious change of *Loaded*; moreover, both $\mathcal{M}_1$ and $\mathcal{M}_2$ have two abnormalities and, as is easy to show, there is no model for $T_{\text{H-M}}$ with only one abnormality. Hence $\mathcal{M}_1$ and $\mathcal{M}_2$ are both preferred and the intended conclusion $\neg Ho(Alive, 3)$ cannot be drawn.

The YSP is not confined to the specific nonmonotonic formalism we introduced above; Hanks and McDermott showed that in most (but not all; see Section 8.2.3) standard nonmonotonic logics, a naive formalization of the law of persistence in terms of minimizing change leads to the same problem [67, 135]. Also, more sophisticated representations of time than the integer-valued time points we used here do not help [67, 135].

Solutions to the YSP The YSP has provoked a very large number of solutions [143, 83, 95, 96, 68, 7, 116, 4, 136]. In some of these a different pre-existing nonmonotonic logic was used for which the problem did not occur after all; in other approaches, more complicated persistence axioms involving new predicates were introduced and/or new and more complicated minimization policies were proposed. We mention *chronological minimization* [83, 96, 143], *causal minimization* [95, 68, 112, 98, 51], *logic programming* [4, 45], *counterfactual* (also known as 'state-based') [7, 8, 81] and *explanation closure* [136, 101] approaches; there are many more. Apart from these approaches, which were all developed within the nonmonotonic reasoning community, a fundamentally different approach based on Bayesian networks and Pearl's ideas on causality started to evolve in the probabilistic reasoning community [116, 51, 33].

The Cyclic Pattern All these approaches had two things in common: (1) they solved the original YSP, but (2) they were soon confronted with their own 'counterexamples', that is, examples of very simple reasoning domains for which they gave counterintuitive results. This led to a cyclic pattern, where counterexamples triggered new proposals, which in turn triggered new counterexamples etc. To give the reader a feel of what happened, we will briefly sketch two of the better known approaches that were proposed: *chronological minimization* and *logic programming* approaches.

8.2.2 Chronological Minimization

The first three proposals to solve the YSP [83, 143, 96] were all based on essentially the same idea: reasoning should follow the direction of time, that is, the default assumptions should be applied in temporal order. In our semantical setting, this 'chronological minimization' (a term due to Shoham [143]) amounts to the following: instead of simply choosing the models with the fewest abnormalities, we select the set of our preferred models in a 'pointwise' fashion: from the class of all classical models consistent with a theory T, we first select the set with the fewest abnormalities at time $t = 0$. Among the remaining models, we select those with the fewest abnormalities at time $t = 1$; we then further restrict ourselves to those with the fewest abnormalities at $t = 2$ etc. The reader may (easily) check that this minimization policy indeed solves the YSP, preferring only the model $\mathcal{M}_1$ with $\mathcal{M}_1 \models \neg Ho(Alive, 3)$. However, it fails to give the intended models for almost every other reasoning problem proposed thereafter. Here is a simple example:

Example 8.4 [Stolen Car Problem] In Kautz' Stolen Car scenario [83], a car is left behind in a parking lot at the initial point in time $t = 0$. The car driver leaves the lot and then returns at some later point in time, say, $t = 10$, to find that the car is not there any more (it might have been stolen, or towed away for example). We model this domain using a theory consisting of persistence axiom (8.6) and the following observation axiom:

$$Ho(Car\_In\_Lot, 0) \wedge \neg Ho(Car\_In\_Lot, 10)$$

If we determine our preferred models according to chronological minimization as defined above, we find that in all selected models we have $Ho(Car\_In\_Lot, t)$ for all $1 \leq t < 10$: we can infer that the car disappears as late as is consistent with the observations, which clearly does not correspond to our intuitive understanding of the situation.

Sandewall [131] proposed to overcome the Stolen Car and similar problems by supplying chronological minimization with a trick called 'filtered preferential entailment' [135]. This yielded the intended models for the Stolen Car scenario but it still failed on some other domains. For this reason, another concept called 'occlusion' was invented and put on top of chronological minimization and filtered preferential entailment. At that point, the development took an interesting turn to which we shall return later.

8.2.3 Logic Programming Approaches

In this example we assume the reader to be familiar with the basics of logic programming; see for example [3]. Most approaches to Logic Programming use *negation as failure* which is an inherently nonmonotonic construct; see [10]. However, the nonmonotonicity is achieved in a manner quite different from our simple minimization of abnormality; we may therefore wonder what happens when temporal reasoning domains are expressed as logic programs.

Interestingly, as noted independently by Apt and Bezem [4], Elkan [43] and Evans [45], if the Yale Shooting *domain* is formalized as a logic program then the Yale Shooting *problem* does not occur. We now show the logic program P, a variation of Apt and Bezem's formalization that has been adjusted to our use of integer time. We adopt PROLOG syntax: capitals stand for variables; `:-` stands for implication and `not` stands for negation-as-failure. `s(T)` denotes the successor of `T`, i.e. `T + 1`. `ho_f` denotes the *Ho*-predicate for fluents; `ho_e` denotes the *Ho*-predicate for events.

```
ho_f(F,s(T))      :- ho_f(F,T), not ab(F,T).

ho_f(loaded,s(T)) :- ho_e(load,T).
ho_f(dead,s(T))   :- ho_e(shoot,T), ho_f(loaded,T).
ab(alive,T)       :- ho_e(shoot,T), ho_f(loaded,T).

ho_f(alive,0).
ho_e(load,0).   ho_e(wait,s(0)).  ho_e(shoot,s(s(0))).
```

Anyone familiar with PROLOG will see that the answer to either of the queries `ho_f(alive,s(0))` and `ho_f(alive,s(s(0)))` will be 'yes' while the answer to the query `ho_f(alive,s(s(s(0))))` will be 'no'. Apt and Bezem showed that the program P also solves the shooting problem from a semantical point of view: several different ways of defining the semantics of logic programs all lead to the adoption of the same unique model $\mathcal{M}$ for P. This is the intended model, satisfying $\mathcal{M} \models \texttt{ho\_f}(\texttt{alive}, \texttt{s}(\texttt{s}(0)))$ and $\mathcal{M} \not\models \texttt{ho\_f}(\texttt{alive}, \texttt{s}(\texttt{s}(\texttt{s}(0))))$.

As formulated above, all negation is handled as 'negation as failure' in this approach. This is undesirable in general; for example, it is not immediately clear how to represent versions of the YSP in which it is not known whether the gun is initially loaded or not. To see why, note that in the program P listed above the answer to the query `ho_f(loaded,0)` will be 'no' though we have not specified anything about `loaded` at time 0 explicitly.

To address this kind of problems, Gelfond and Lifschitz [53] introduced 'extended logic programs'. These are capable of dealing with both negation-as-failure and classical negation. However, the class of reasoning domains that could be handled by a basic formalization of persistence in an extended logic program was still quite restricted (for example, it had problems with the Stolen Car domain). To be able to handle larger classes of reasoning domains, the logic programming community has gradually adopted a different methodology altogether, as we shall see in Section 8.3.1.

8.2.4 The Ramification Problem

We have just seen two typical examples of the kinds of solutions that were proposed, and one typical example of the kind of counterexamples (the Stolen Car problem) this led to. As the cycle between solutions and counterexamples continued, it became increasingly clear that the most challenging of these counterexamples were all interrelated: they were all instances of the *ramification problem* [135]. This is the problem of how to properly deal with the side effects of actions. Sometimes it is viewed as part of the frame problem, sometimes it is seen as a problem in its own right [135]. It occurs in domains where an action or event has only a small number of *direct* effects while it can have a very large number of *indirect effects*. For example, if a person is shot at, he/she will stop being alive (a direct effect), but he/she will also stop walking, breathing, eating etc.; these are indirect effects. Theories in which all such indirect effects of an action are stated explicitly would quickly become unmanageably large; one would prefer to state them in more general and implicit ways. Here we will give just one example (due to Baker [7]) of the ramification problem - there are many more, all of them highlighting different aspects of it; some of these will be discussed in the next chapter, Section 9.5.

Example 8.5 [The Walking Turkey - Ramification] Imagine a turkey that can be either *Alive* or not and that can be either *Walking* or not. If the turkey is shot at (we assume a gun that is always loaded here) it will stop being alive. Initially, the turkey is alive. Then somebody shoots at the turkey. We can formalize this as follows:

$$Ho(Alive, 0) \wedge Ho(Shoot, 0) \tag{8.11}$$

$$Ho(Shoot, t) \supset \neg Ho(Alive, t+1) \tag{8.12}$$

We now want to encode the additional rule that if the turkey stops being alive, it will also stop walking. In light of the discussion above, we do *not* want to add a rule of the form $Ho(Shoot, t) \supset \neg Ho(Walking, t+1)$; such a methodology would lead to overly large and inconvenient theories in domains where there are several actions with direct effect $\neg Ho(Alive, t)$ and/or where $\neg Ho(Alive, t)$ has many indirect effects. Instead, it seems more reasonable to try the following *state constraint axiom*:

$$\neg Ho(Alive, t) \supset \neg Ho(Walking, t) \tag{8.13}$$

Unfortunately this will not work in general. We note first that our naive persistence formalization will lead to counterintuitive results, as the reader may (easily) check: let T be the union of (8.11)–(8.13) and persistence axiom (8.6). By axioms (8.6), (8.11) and (8.12), the preferred models for T all have abnormality $Ab(Alive, 0)$. There is only one model $\mathcal{M}$ with no further abnormalities, therefore, $\mathcal{M}$ is the single preferred model. By (8.13), $\mathcal{M} \models \neg Ho(Walking, 1)$. By persistence, $\mathcal{M} \models \neg Ho(Walking, 0)$. This means that, while nothing is known about whether the Turkey is initially *Walking* or not, we can infer that it is not, clearly an unintended (overly strong) result. The two other approaches introduced here also have problems with this example, but we shall not discuss these further.

It seems we should look for a solution in which the fluent *Walking* is somehow 'released' that is, not subjected to persistence, whenever *Alive* is affected by an action. This kind of solution was indeed proposed [135], but was later found to be inadequate by several authors [99, 103]. Briefly, the reason is that there may be other fluents in a domain for which the domain constraint $\neg Ho(A, t) \supset \neg Ho(B, t)$ should also hold in all intended models, but for which B should *not* be released from persistence when A is affected by an action. In the next chapter, Section 9.5 this will be discussed in detail.

8.2.5 Ramification and the Resurgence of 'Causal' Approaches

Let us now return to the historical development. The ramification problem was identified as a problem shortly after the YSP was presented (1986). From that time onwards, reasoning domains involving it have been gradually accumulating. A few early solutions were proposed [7, 97], but none of them of any reasonable generality. Then, around 1995, several authors, apparently independently, succeeded in defining relatively simple and elegant formalisms that where sufficiently general to account for a substantial number of examples (including the Walking Turkey) [99, 103, 66]. These formalisms were all closely related to one another; in the following we concentrate on one of them, namely Lin's [99] approach. It is based on the intuition that some effects of actions *cause* other effects while for other effects, this is not the case. Lin introduces an additional predicate *Caused*. To give an example, in Lin's formalization, the fact that $\neg Walking$ is an indirect effect of $\neg Alive$ would be expressed as:

$$\neg Ho(Alive, t) \supset Caused(Walking, \text{FALSE}, t) \tag{8.14}$$

to be read as '$\neg$*Alive causes* $\neg$*Walking*'. This proposal is related to the early *causal minimization approaches* to the Yale Shooting Problem. These used a causal predicate and a philosophy similar to Lin's. However, there were a few small but essential differences which allowed Lin to solve a much wider class of domains than these forerunners. In Chapter 10 we shall treat Lin's proposal in detail.

Interestingly, around the same time as Lin's approach was published, McCain and Turner [103] independently made a very similar proposal. Moreover, several of the originally non-causal approaches to the frame problem seemed to develop naturally, as it were, into approaches that were closely related to Lin's approach. We already mentioned that in the *chronological minimization* school, the concept of 'occlusion' had been introduced. Later, Gustafsson and Doherty [66] showed that one can also treat persistence and ramification based *only* on occlusion *without* combining it with chronological minimization. They showed that the resulting theories are closely related to Lin's. In the *logic programming* community, people gradually came to adopt a different methodology which we shall describe in Section 8.3.1. The methodology formalizes domains in some 'high level action description language'. Such languages invariably have a primitive **causes** which functions in a way similar to Lin's *Caused*-predicate [54, 57, 12]. In Shanahan's*event calculus-based approach* [139], a construct *Initiates* is used in way similar to Lin's use of *Caused* ([139], Chapter 16). Finally, Lin's own approach really had developed as an extension of the *explanation closure approaches* of Schubert [136] and Lin and Reiter [101] rather than one of the original causal approaches.

The Present Situation Summarizing , many different research groups nowadays use a construct similar to Lin's *Caused*. Approaches to persistence developed in the probabilistic community are all based on some version or other of Pearl's ideas about causation. It seems that among all the originally proposed approaches to persistence, the causal approach is the clear winner.

However, *Caused* is just a predicate with a certain semantics. It has been called '*Caused*' because it can often successfully be used in contexts where $\mathcal{M} \models \mathit{Caused}(f,t)$ seems to correspond to the statement 'fluent f is caused to hold at time t'. Unfortunately, causality is a notoriously difficult concept and it seems quite unclear what such a statement really means. Gustafsson, Doherty [66] and Shanahan [139] do not interpret their *Caused*-like constructs as being necessarily related to causation. Thielscher [151] even shows that the *Caused* predicate can sometimes be fruitfully used in a context where it does *not* correspond to the use of 'causes' in natural language (see the next chapter, Section 9.5.3). At the time of writing this thesis, one of the main concerns of the field is the question whether the causal road is really the one to take.

8.3 Research Goals, Criticisms and Methodology

Having introduced the field by giving examples of the problems it is dealing with, we now stand back a little and discuss the major research goals and the methodology of the field. Both of these have been subject to much criticism [108, 135]. Recall that the original goal was to develop a theory of action and change as part of a grand 'logic of common-sense reasoning'. Later, when it was realized that this goal was hard or even impossible to achieve, many researchers scaled down their ambition.

The Research Goal The goal now became to find an *engineering solution*: to develop a formalism (preferably, but not necessarily a logic) capable of handling a reasonably large class of reasoning domains involving action and change. Here 'handling' is meant in the following sense: the domains are represented in a *clear* and *concise* manner, the inferences that can be made from a representation of a domain are all *correct* (in the sense that they agree with our intuitive understanding of the domain) and *efficient* (in the sense that relevant inferences can be made in a small number of computation steps). The system developed could then be installed as a specific component of the reasoning- and planning system of some intelligent agent (e.g. a robot). Even the achievability of this more modest goal can be questioned. The main problem is that for sufficiently complex reasoning domains, it is really not clear what a 'correct' representation of such a domain would be. We discuss two reasons:

Clashes of Intuitions A 'correct' representation of a domain should allow us to make those inferences that we consider *intuitively* correct. Unfortunately, different researchers often make incompatible statements about what are the admissible (intuitive) common-sense inferences for a given example. In the next chapter (page 218 and 221) we shall see examples of reasoning domains where different people think that different, conflicting inferences are admissible. More in general, one sometimes gets the feeling that the community really is trying to formulate a logic that permits those inferences that are judged admissible by the members of the community itself (and not necessarily by others) (see [156]).

Arbitrariness of Translations Formalizing a domain always involves a step where our intuitive understanding of the domain is translated into a formal representation of it. This step *itself* is, however, necessarily informal: there is no formally defined translation function that, when input a description of a domain in natural language, would output a description of the same domain in the many-sorted first-order language we use in this chapter. The same holds of course for any other formalism. The inherent informality makes it unclear what parts of a problem are really solved by a clever translation and what parts are solved by the formalism itself. It then, once more, becomes very hard to judge whether a given formalism can be considered adequate or not: if somebody dismisses it on the grounds that it leads to counterintuitive results, somebody else may always claim that the formalism can handle the task after all; one merely has to express the domain knowledge in a different way.

A third criticism is directed against the methodology of the field rather than its goals:

Chaotic Methodology Even *if* the overall goal of the field can be achieved, it is very doubtful whether we get any closer to it by accumulating more and more specific examples of problematic domains and adjusting existing formalisms to cope with them. All past proposals were soon confronted with simple counterexamples (page 191). This suggests that if an approach handles a bunch of specific examples correctly, this does not give any guarantee that it really handles a 'wide class' of domains correctly.

In reaction to these and similar criticisms, an alternative methodology, often called 'systematic' [135] has developed in recent years. As we shall see, this methodology partly (but only partly) resolves the issues raised above.

8.3.1 The Systematic Methodology

Originally proposed around 1992 [132, 102, 54], the systematic methodology nowadays underlies the work of a substantial part of the researchers in this field. There are two main variations of this methodology, one developed by Sandewall [132] and one by Gelfond and Lifschitz [54]. We will concentrate on the latter variation, which has mainly been developed in the context of the logic programming approaches but which by now has been applied to many other approaches as well [80]. Roughly, it works as follows: instead of directly formalizing domain knowledge as axioms in some logic, one uses a high-level 'action description language' (ADL) [54] with a restricted syntax. The idea is that by using a restricted, tailor-made language rather than a full first-order language, it is possible to endow such a language with very simple and clear semantics. We shall see an example of such a language, the language $\mathcal{L}_3$, in Chapter 10, Section 10.4.

ADLs together with their semantics are regarded as *definitions* of a class of reasoning domains. The language defines what reasoning domains belong to the class (namely, precisely those that can be represented by the language). The semantics determine for every element of the class what the correct inferences for that element are. For any given ADL+semantics and any given logical formalism L, one can define a formal translation function π mapping domain descriptions written in the ADL to domain descriptions written in the language of L. The ADL+its semantics + a translation function 'resolve' all three issues raised above: there can be no clash of intuitions about what is correct and what not, since this is *defined* by the semantics of the ADL. There is no informal translation step any more, for translations are done by a precisely defined function π. Finally, one can often formally *prove* that a proposal L in combination with π is correct for a whole *class* of reasoning domains by proving that for every domain description D written in the ADL, the models assigned to D (by the semantics of the ADL) correspond to the models assigned to $\pi(D)$ (by the semantics of L).

Begging the Question Unfortunately, this solution evades all the difficulties rather than faces them: they all re-occur one level higher, in the question of how to define the proper semantics for an ADL; but this question is simply moved outside the scope of investigation. Nevertheless, the idea of ADLs, in our opinion, has two merits: first, by the restricted syntax of an ADL, it becomes much easier (though still hard) to judge whether it assigns the 'correct' semantics to all the problems that can be represented in it. Second, the idea of formally translating descriptions of approach A to descriptions of approach B is, we think, fundamental to any advances of the field. It allows us, when introducing a new approach, to show exactly in what ways it is related to previous approaches.

8.4 An Alternative View: NMTR as Modeling

In the previous section we saw some serious criticisms of the goals and methods of our research field and we presented a methodology that partially answers them. As we see it, at least part of the criticisms can be better met by a more radical shift of the focus and goals of the field. In this new view, the theories to be developed should be *models* about the *physics* of action and change. To explain what we mean by this and why it is really different from the research goal as stated before, we need to say a little about modeling in general.

8.4.1 Two Aspects of Modeling

In our view, the field of NMTR is basically concerned with finding good Models for scenarios concerning action and change. A logical theory (i.e. a set of axioms) together with a nonmonotonic semantics can be considered as a *Model* of the domain of interest (from now on we write the word 'Model' with capital M to indicate that it is used not in the logician's, but rather in the general sense of a 'model of some phenomenon'. When written uncapitalized, it denotes the logician's notion of 'mathematical structure in which all axioms of some theory hold'). A Model of a domain can be specified by giving the list of laws to which the basic entities in the domain are subject (or, more correctly, *would be* subject if the Model where in any sense 'true'). However, for many interesting domains, this list will be very long. Yet it is often possible to specify a Model in a concise manner, by using some precisely defined convention about how the short specification can be translated into the full list of laws corresponding to it. In such a case, there are two fundamentally different aspects to the process of modeling:

1. Finding a good Model for a domain, that is, identifying a list of 'laws' to which the domain is subject.
2. Finding a method to specify such a Model in a compact manner.

Much of the NMTR literature can be characterized by a confusion of these two issues. Yet (at least in our view) they should be clearly separated in order to make any real progress in the field. Perhaps the best example of this confusion is given by what we called the 'naive persistence formalization', that is persistence axiom (8.6) together with the minimization of Ab. Let us stand back a little and analyze what rôle the researchers who proposed it had intended it to play. We see that it was meant to capture some quite different things at the same time:

a. *Modeling* persistence per se: if at time t no action takes place that affects fluent f, then the value of f usually remains unchanged at time $t+1$. For example, this says that if Fred is alive and none of the actions that can kill Fred take place at time t, then he will usually still be alive at time $t+1$.

b. *Expressing* that actions affect no more things in the world than those we explicitly state they affect. For example, if a *Load* action takes place, then this should have no effects on *Alive* ; if a *Wait* action takes place, this should neither have an effect on *Loaded* nor on *Alive*, etc.; yet none of this is stated explicitly. These laws about non-effects are assumed to be enforced by the persistence axiom and the minimization of Ab.

(a) can be seen as a kind of 'physical law'. It is part of a Model of the domain as referred to in item (1) above. (b) is an attempt to express some of these laws (namely, the laws involving the non-effects of actions) in a very concise manner. As such it is related to item (2) above. Item (1) is about determining which things are generally the case in our domain of interest; item (2) is about determining how to specify these things.

8.4.2 Models, models, and Two Uses of Nonmonotonicity

To clarify our distinction we will give an alternative, more precise characterization of what we mean by a Model and by a 'concise representation' of it.

We first define the notion of 'Model' in a simplistic way; a more realistic definition follows later. For now, a (modeler's) *Model* of a domain is *defined* as a *set* of (logician's) models. Intuitively, each model $\mathcal{M}$ in the Model $\mathbf{M}$ can be identified with a single instantiations of the domain. For example, the 'intended' model $\mathcal{M}_1$ with $\mathcal{M}_1 \models \neg Ho(Alive, 3)$ (page 190) should be an element of any reasonable Model of the Yale Shooting domain.

Let $\mathbf{M}$ be a Model of some domain defined in this way. For a sentence Γ we write $\mathbf{M} \models \Gamma$ if for each model $\mathcal{M} \in \mathbf{M}$ we have $\mathcal{M} \models \Gamma$.

The Contingent and the General; Generic Models Now note that the problem domains we deal with in NMTR can usually have several instantiations: typically there are *general laws* (that hold in all possible instantiations of the domain) and *contingent facts* (whose truth value differs from instantiation to instantiation). For example, in the Yale Shooting domain, a sentence like $Ho(Loaded, 0)$ is a contingent fact. A sentence like

$$Ho(Load, t) \supset Ho(Loaded, t + 1)$$

is a general law.

For simplicity, we will only consider 'generic' Models of domains. These are Models in which none of the contingent facts have been instantiated (for example, a 'generic' Model of the Yale Shooting domain should contain both a model $\mathcal{M}_a$ with $\mathcal{M}_a \models Ho(Alive, 0)$ and a model $\mathcal{M}_b$ with $\mathcal{M}_b \models \neg Ho(Alive, 0)$).

Theories as Specifications of Models Generic Models can be characterized by a set of logician's models, but, at least if they are not too complicated, also by a logical theory. We call a theory T a *sound specification* of a Model $\mathbf{M}$ if, for all sentences Γ,

$$T \models \Gamma \Rightarrow \mathbf{M} \models \Gamma \tag{8.15}$$

We call a theory T a *complete specification* of a Model $\mathbf{M}$ if, for all sentences Γ,

$$\mathbf{M} \models \Gamma \Rightarrow T \models \Gamma \tag{8.16}$$

We call T an incomplete specification of $\mathbf{M}$ iff it is not a complete specification of $\mathbf{M}$.

One of the main goals of NMTR is to specify the Models of our reasoning domains in a compact manner (recall that we wanted to avoid using all the 'frame axioms' (page 188)). This is almost always done by giving a theory T that is a sound but incomplete specification of the Model $\mathbf{M}$ we have in mind. This theory is then turned into a complete specification by using some nonmonotonic entailment relation $\approx$ ($T \approx \Gamma$ denoting that Γ holds in all most-preferred models of T). Hence, in the practice of NMTR one uses a theory T to specify $\mathbf{M}$ which is such that (8.15) holds, (8.16) does not hold, but it *does* hold for all sentences Γ that

$$T \approx \Gamma \Leftrightarrow \mathbf{M} \models \Gamma \tag{8.17}$$

There would be nothing wrong with this, were it not for the fact that a good Model for the kind of domains we are actually interested in is usually not so simple that it can be identified with a plain set of logician's models. We often want such a Model to have a preference structure over the possible instantiations (logician's models) of the domain. In such a case, a Model $\mathbf{M}$ becomes an *ordered* set of logician's models rather than a plain set. For example, the generic Model of the Stolen Car domain (page 191) should contain a model $\mathcal{M}$ with $\mathcal{M} \models \forall t.\neg Ho(Car\_In\_Lot, t)$ among its most-preferred models; but, among the less preferred models, it should also contain a model $\mathcal{M}'$ with $\mathcal{M}' \models Ho(Car\_In\_Lot, 0) \wedge \neg Ho(Car\_In\_Lot, 10)$: after all, if we want to model the fact that persistence *may* be broken without discernible reason, then $\mathcal{M}'$ *should* be a possible instantiation of the domain.

Hence, more realistic Models $\mathbf{M}$ can be identified with a set of logician's models together with a preference ordering over these models. Let us call the preference ordering of this kind that belongs to a Model $\mathbf{M}$ the *internal preference structure of* $\mathbf{M}$. In contrast, we will call the nonmonotonic mapping performed by the relation $\approx$ of Equivalence 8.17 a *completion mapping*, since it is meant to turn theories that are incomplete specifications of a Model $\mathcal{M}$ into complete specifications of $\mathcal{M}$.

The Confusion Both completion mappings and internal preference structures are nonmonotonic constructs. In many approaches to NMTR, they are implemented by the same, single, mechanism. We already saw the example of the naive persistence formalization, where all nonmonotonicity is handled by minimization of Ab. We claim the same thing happens in chronological minimization, early causal approaches and in state-based approaches [83, 96, 143, 95, 7]). In our view, the problematic behaviour of these approaches is partly due to their treating both uses of nonmonotonicity by a single mechanism. For these two uses are really very different: the nonmonotonicity inherent in the completion mapping is concerned with how the Model of the domain is to be specified, while the internal preference structure (which is defined relative to a model $\mathbf{M}$) is *part of that Model itself.*

Further Uses of the Distinction In the next chapter, we shall present our own theory of nonmonotonic reasoning. In this theory the two different uses of nonmonotonicity will be strictly separated. In the conclusion to the next chapter (page 234) this will be verified and some examples will be given. In the Epilogue to part III, we shall further see that the use of nonmonotonicity to make incomplete specifications complete is *not related to default reasoning.* Finally, we shall point out there how the distinction between the two uses helps in understanding the relationship between probability theory and nonmonotonic logic.

8.4.3 A New Research Goal

The distinction introduced above leads to a different view of the goals of the field. The challenge now becomes:

Alternative Research Goal First, to find good Models for domains of action and change; and second, to find concise and computationally efficient representations of these Models.

A Model should constrain the possible instantiations of a domain so that they obey some set of 'laws of nature'. A concise and efficient representation of a model **M** MAY BE a logical theory T that incompletely specifies $\mathcal{M}$ together with a completion preference ordering $\models_{\approx}$, but we see no reason why to constrain ourselves in this way.

The 'laws of nature' implied by a model should be transparent: it should be clear what assumptions one has to make about the world in order for them to hold. For example, the persistence law that we will formulate in our own theory of NMTR (Section 9.6) can be read as follows:

> If a fluent f holds at time t and there are no interventions in the domain at time t that can influence the value of f, then with high probability f still holds at time $t+1$.

We call this a 'physical law' since it describes what really happens in the world if it is regarded at a certain level of abstraction. To clarify, we give an example of a 'law' used in the existing literature that clearly can*not* be viewed as a physical law:

> A fluent f changes value as late as possible.

This is just 'chronological minimization[2]' discussed in Section 8.2.2, page 191. Whereas our own formalization of persistence applies for many properties f and events e that exist in the real world, the second formalization, at least in our view, does not.

If, as in chronological minimization, one does not separate the issues of concise representation and Modeling, then the preference ordering over models one comes up with needs to serve two purposes at once; therefore, even if a single preference ordering of this kind exists that leads to intuitive results, it will probably not correspond in any way to 'what happens in the real world'.

The 'modeling' view of the field is adopted not only in our own work. It is also implicit in the works of some of the other researchers in the field; we mention Sandewall's 'trajectory semantics' [133] and Thielscher's work on causality [151]. The work on action and change done in the probabilistic community [33, 34, 52] invariably takes a 'modeling' perspective.

[2]Of course, the inventors of chronological minimization did not believe that fluents change as late as possible in the real world. Rather, they wanted to formalize the idea that when people *reason* about action and time, they sometimes do so in temporal order. However, the law that fluents change as late as possible *is* a direct consequence of the formal definition of chronological minimization.

8.4.4 The Clash of Intuitions Revisited

In Section 8.3 the field was criticized because of the inherent difficulty in determining whether a proposed formalism is really 'correct', the reason being that different people may have conflicting intuitions about what is and what is not an admissible inference for some given domain.

We think these 'clashes of intuitions' are better understood, and can be partially alleviated, if one adopts the alternative research goal as stated on the previous page. In our view, there are really two different kinds of 'clashes of intuitions':

1. Clashes concerning the question 'what is a *good Model* for a given domain', which is the central question of the first part of the research goal.

2. Clashes concerning the question 'what is a *concise yet intuitive way to specify* a Model of a domain?', which is a question related to the second part of the research goal.

The clashes of intuitions of kind (1) are problematic, but they seem not much worse than clashes of intuitions that appear when Models are developed in fields different from NMTR, i.e. economics, biology. As in these fields, one can use, at least in principle, empirical evidence to determine what a good Model is: when in doubt, one can think of a robot, equipped with the Model under investigation. The question of whether a Model is 'correct' then becomes: will the robot draw sensible conclusions and function sensibly, in the domains we intend to place it in?

The clashes of kind (2) are more serious; they are clashes of a kind that simply does not seem to occur in other fields concerned with modeling (at least in statistics, Models are *always* described by a *complete* specification). In our view, there are *bound* to be differing intuitions here. For this reason, we think it is a good idea to drop the requirement of intuitive specifications: in our view, in a proper approach to the field, the mapping from concise representations to Models should just be *simple* and *precisely defined*, but not necessarily intuitive - simplicity and precision are all we can ask for.

Let us explain this a bit further. We think that the actual clashes of intuitions thus far encountered in the literature are really mainly about clashes of kind (2). Recall that, in order to arrive at a compact representation, the general approach of the field has been to specify only *some* of the laws that should hold according to the Model. To make this work, one needs precisely defined rules (for example, a nonmonotonic semantics) to map an incompletely specified theory to the Model it is supposed to correspond to. Something similar occurs in natural language, where it is commonplace to leave out certain things from a description. Most speakers of a language use approximately the same *narrative conventions* [1] to (unconsciously) determine what somebody really means if he/she gives an incomplete description of a situation. But when an incomplete description of a Model is expressed in the language of first-order logic (i.e. as a logical theory), it seems quite strange to assume that different users of this (artificial) language will share the same intuitions about how incomplete descriptions should be mapped to Models: first-order logic is a recent, man-made invention. It would be really surprising if a set of 'logical narrative conventions' existed that would be judged intuitive by most or all people. This is especially so since, as we shall see in the next chapter, the rule 'if an effect of an action is not specified, then it is not an effect', is not enough; as domains get more complicated, much more complicated conventions are needed, at least if concise theories (descriptions) are what is asked for.

Summarizing, we think that by simply not attempting to do the compact specification of Models in an intuitive manner, the clashes of intuitions are reduced to those occurring for complete specifications; but there, they will not be much more serious than in other fields, and, at least in principle, there are empirical means (by testing a Model or an actual robot equipped with it in actual domains) to help resolve such clashes.

8.5 Summary; Main Challenges

We end this chapter with a summary and an overview of what we regard the main challenges of the field. We have seen that NMTR is based on an elusive goal, the 'logic of common-sense'. Many problematic reasoning domains show that this goal is not at all easy to achieve. Most modern approaches dealing with these problems are interrelated in that they are based on explicitly expressing causal relationships. One of the main current challenges of the field is therefore, in our view:

- Find out the exact relations between all these approaches. Gain a better understanding of the use they make of causality.

More generally one may question some of the assumptions behind the goals and methods of the field. This leads to what we personally regard as another fundamental challenge (though, unlike the first, most researchers seem not to be too concerned about this issue):

- Search for a better methodology; gain a better understanding of the possibilities and limitations of the field.

In Section 8.4 we proposed a different, more *modeling*-oriented view of the field, based on separating the modeling of a domain and finding a concise representation of a Model. In the next chapter, we shall present a theory of NMTR that adheres to this separation. Its treatment of causality is based on a successful approach developed outside the field. Hence, our approach will have implications for both challenges mentioned here.

Chapter 9

Causal Theories for Nonmonotonic Temporal Reasoning

In this chapter we present our own theory of Nonmonotonic Temporal Reasoning. Our theory is based on an approach to causality that has been developed in a statistical rather than an NMTR context. Nevertheless it turns out to be closely related to several recent approaches developed within the NMTR community. Before presenting our theory, we give an introduction in which we discuss what we wish to achieve with our theory and what its basic features are.

9.1 Introduction

By and large, the most successful existing approaches to NMTR are 'causal' in the sense that they use a predicate or logical connective to explicitly express causal relationships. [103, 105, 104, 99, 151, 12, 52]; we discussed this at length in the previous chapter (page 194). Causal approaches seem required especially in domains involving the ramification problem. It has even been argued [151] that the explicit modeling of causal relationships is a necessary ingredient of any solution to the ramification problem. Still, witness the frequently heated debates on the issue taking place during workshops and conferences, it seems that many researchers remain skeptical about the power and applicability of causal approaches. A reason for this may be that hardly any of these approaches has addressed the basic issue of causality directly: what has to be the case in the world in order for the assertion 'A causes B' to be valid? In other words, it is not attempted to define causal relationships in non-causal terms. Instead, causal domain knowledge is formalized explicitly, typically as rules of the form 'A causes B' – and the approaches rely on the assumption that their semantics will yield the right inferences from such assertions.

On the other hand, J. Pearl has provided an empirical semantics for causation that has met with considerable success in statistical applications [117, 118, 119, 120]. However, as will be explained below, Pearl's theory as such cannot be directly applied to common-sense reasoning about action and change. The theory we present in this chapter extends Pearl's theory in a straightforward way so as to overcome this hindrance. In this way we arrive at a causal theory that has the advantage that it can also be understood in non-causal terms, namely as a theory that gives a certain semantics to 'actions'. The theory will be capable of handling wide classes of reasoning domains, specifically involving *persistence*, (complicated) *ramifications* (page 193), actions with *disjunctive and non-deterministic effects*, domains with incomplete specification of the events that occur and *concurrent events*, furthermore *unknown fluent dependencies*, *causal chains of events* and *surprises*. All these terms will be explained in due course. In this chapter we will show that the theory gives intuitive results for specific examples of domains with all these features; in the next chapter we shall provide evidence that it deals well with complete *classes* of reasoning domains involving most of these features.

As we will see, the theory turns out to be closely related to, yet different from several other existing approaches. The theory will therefore allow us to interpret these approaches (and specifically, the use they make of causality), in a new way. Summarizing:

Research Goal Our aim is to develop a theory that allows us to

1. gain a better understanding of the rôle of causality in NMTR by showing its relation to Pearl's theory of causation; specifically, we will show *how causal notions can be understood in non-causal terms.*

2. deal with a wide range of domains involving action and change; moreover, to provide evidence that the way we deal with these domains is, in some sense 'correct'.

3. show how some of the existing approaches are interrelated and explain their successes and failures.

These goals will be pursued in the context of the more general research goal we formulated in the previous chapter (Chapter 8, Section 8.4): we separate the issues of finding a good model for a domain and compactly specifying such a model.

Structure of this Chapter

In the present chapter we focus on the development of our theory. We show how it arises as an extension of Pearl's theory and we illustrate its use by applying it to several reasoning domains. At the end of the chapter we shall return to the goals mentioned above and see whether they have been achieved. A truely detailed comparison to other approaches will be the subject of the next chapter.

In the next sections we will gradually build up our theory. In Section 9.2 we provide an informal introduction to Pearl's semantics of causation. We then introduce 'causal theories' which are a simple, propositional extension of Pearl's causal theories. In Section 9.4 we reach our first 'milestone' when we instantiate our theories to domains involving persistence. The resulting theories are capable of representing and dealing with most aspects of the ramification problem as well as the problem of handling disjunctive and indeterministic effects of actions. We illustrate this by applying the theories to several examples in Section 9.5.

In order to represent an arbitrary number of time points, our causal theories must be extended to the first-order case. This allows us to treat a much wider variety of domains. Section 9.6 discusses the extension to the first order case; it is shown how this allows us to handle the original YSP and related problems, 'dependent' fluents, concurrent events, causal chains of events and surprises. This is followed by a conclusion. The chapter ends with an appendix in which we show formally how our theory arises as an extension of Pearl's.

The End Before the Beginning

Since we will develop our theory gradually, starting with a simple version and then adding more and more features as we encounter more and more complicated examples, it may be useful to first give an idea of what we will end up with. Our general theory will be instantiated to several 'causal theories'. The most sophisticated version of causal theories will be given in Definition 9.27 on page 232. In the box on page 208 we provide an outline of how these sophisticated causal theories will look like.

9.2 Informal Introduction to Causal Theories

9.2.1 Actions Remove Some Dependencies, but Keep Others Intact!

Causal theories are about domains of variables whose values may be influenced by *interventions*. Each variable X in the domain depends in some deterministic manner on a subset $S(X)$ of all the other variables; here 'deterministic' means that the value of X is completely determined by the values of the variables in $S(X)$. Now if an intervention takes place that sets the value of a variable X to some value, then some of the variables in $S(X)$ become independent of X while the rest of the variables in $S(X)$ keep their dependence. If we equate an *action* with a *set of interventions*, this induces a semantics for the concept of 'action'. This resulting semantics, which will be discussed in more detail below, has sometimes been called *the sufficient cause principle* [34, 33]. An action that sets the value of a variable X is then called a 'sufficient cause' for X.

We give a simple example: let us denote by $\mathit{Alive}(t)$ whether or not some turkey is alive at time t. Normally, i.e. if no interventions take place, we have that if it is given that the turkey is alive at time $t+1$, we can deduce that it will also be alive at time $t+2$ and that it already had been alive at time t. Now if an intervention takes place that sets $\mathit{Alive}(t+1)$ to FALSE (for example, if somebody shoots at the turkey), then the value of Alive at time t becomes independent of the value of Alive at time $t+1$: we can no longer infer anything on whether or not the turkey was alive at time t (disappearance of a dependency). On the other hand, the dependency between

Causal Theories The basic entities of first-order causal theories are *fluents* (properties of the world that persist), *events* and (nonnegative integer) *time-points*. The main predicates used are *Ho*, *Do* and *Ab*.

- $Ho(f,t)$ denotes that fluent f holds at time t; $Ho(e,t)$ denotes that event e takes place at time t.
- $Do(x,b,t)$ denotes that 'an intervention takes place that sets fluent or event x to value b' at time t.
- $Ab_1(x,t), Ab_2(x,t)$ both denote that something abnormal is the case concerning fluent or event x at time t. In our use of Ab_1 and Ab_2, an 'abnormality' can *always* be seen as something *unexpected* or *surprising*.

Causal Models We determine the 'causal' models of causal theories in several steps. We start with a theory consisting of two sets of axioms: CONS and EQ. CONS contains all axioms specific to the domain under consideration; EQ contains general axioms involving persistence and the effects of interventions. Then we:

1. circumscribe (minimize) the *Do*-predicate in CONS, keeping Ho, Ab_1 and Ab_2 fixed.
2. add the axioms in EQ to the theory resulting from (1).
3. minimize the *Ab*-predicates in the theory resulting from (2).

The models resulting from step (3) are the *preferred causal models* of the causal theory defined by CONS and EQ. Note that we use nonmonotonic inference in steps (1) and (3).

Essential Features of our Theory Our theory is characterized by two features:

1. It follows the *alternative research goal* stated in Section 8.4 of the previous chapter, page 197. The distinction introduced there is reflected in our use of nonmonotonicity: in step (1) above, nonmonotonicity serves to enable a concise representation of causal theories. In step (3), nonmonotonicity serves to determine the 'most expected', or 'least surprising' models of a fully specified causal theory.
2. It follows *Pearl's theory of causation*. This is embodied in the semantics and the use of *Do*. The semantics of *Do* is derived from Pearl's 'sufficient cause principle'. The use of *Do* involves one of the central ideas in Pearl's theory: *observations and interventions should be modeled differently*. Observations are modeled using *Ho*, interventions are modeled using *Do*.

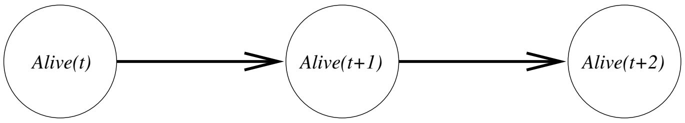


Figure 9.1: A Very Simple Causal Graph.

Alive$(t+1)$ and *Alive*$(t+2)$ has not disappeared: we can still infer that the turkey will *not* be alive at time $t+2$.

Hence if we have a causal theory for a domain, and we know the interventions that take place in the domain, we end up with a new, updated causal theory. We now discuss two equivalent ways of defining how such a new theory may be arrived at.

Causal Graphs and Structural Equations

We can depict the dependencies between the different variables of a given causal theory as a (possibly cyclic) *causal graph*[1] [34, 33]. Figure 9.1 shows the causal graph for the example above. The semantics of actions can be characterized in terms of these causal graphs: if an action takes place that sets the variable X to some value, then that variable becomes independent of all its non-descendants in the graph. Hence the intervention changes the theory into a new, updated one that is characterized by the causal graph in which all *incoming* arcs into the node corresponding to X have been removed while all other arcs remain. The graph does not give any information about the exact functional relationships between the variables in the domain, but only about which variables become independent in the presence of an intervention; we can now either regard causal theories as causal graphs together with a description of the functional relations between parents and children in the graph, or, as has been done by Pearl in his more recent papers [119, 120] we can avoid using graphs by writing the functional relations in the form of *structural equations*: these are equations with a directionality attached to them. A set of structural equations for our example would look as follows: $\{Alive(1) = Alive(0)\ ;\ Alive(2) = Alive(1)\ ; \ldots\}$. Our action semantics, i.e. the sufficient cause principle, can now be rephrased like this: if an intervention takes place that sets variable X to value x, then the structural equation $X = \ldots$ is replaced by the new equation $X = x$. In our example, if just one intervention takes place, and this intervention sets *Alive*(2) to FALSE, then the equation $Alive(2) = Alive(1)$ gets replaced by $Alive(2) = \text{FALSE}$, while all other structural equations remain unchanged.

In Pearl's terminology, each structural equation can be seen as a *micro theory* [34]. It describes a single 'mechanism' whose input and output are given by, respectively, the right-hand and the left-hand side of the equation. All these mechanisms are coupled to form a complete theory. But if an intervention takes place, some of the mechanisms get *decoupled* from the rest and the output they normally provide is replaced by the result of the intervention. In other words, in Pearl's theory interventions are defined as *surgeries* on the set of mechanisms: an intervention keeps some of the mechanisms

[1] Historically, causal graphs have arisen as an extension of the Bayesian networks we saw in Chapter 6.

intact, while 'turning off' others, thereby changing the behaviour of the system of mechanisms as a whole.

Why we need to extend Pearl's theories

In common-sense reasoning about action, we are typically interested in domains where the fact whether or not an intervention takes place may depend on the values of some of the variables in the domain; this is impossible to model in Pearl's [119, 120] basic theory. Pearl's theory also lacks the representational power needed to express *global* domain constraints. For example, we may have $Alive(t) \equiv \neg Dead(t)$ irrespective of any action that influences the value of $Alive$; this formula should always hold in our domains and should not be replaced when $Alive$ is set to FALSE. It is therefore not a structural equation. All constraints between variables in Pearl's basic theory must be given in the form of structural equations, so it is impossible to model such a global constraint. In our work we extend Pearlian causal theories in a way that allows us to represent both kinds of domain knowledge mentioned here. We do this by simply adding to the set of structural equations a set of *global constraints*, consisting of formulas that should hold in all models of our theories irrespective of what interventions take place. These global constraints are also allowed to mention interventions. To be able to express those, we introduce for each propositional variable X in our domain two additional propositional variables $Do(X, \text{TRUE})$ and $Do(X, \text{FALSE})$, representing interventions setting X either to TRUE or to FALSE.

9.3 Propositional Causal Theories

Our propositional causal theories are a straightforward extension of Pearl's causal theories. In an appendix to this chapter (page 235) we show exactly how they are related to Pearl's causal theories as defined in [119, 120].

Preliminaries

Let $\mathbf{B} = \{\text{TRUE}, \text{FALSE}\}$. A *truth value* is an element of $\mathbf{B}$. Let $\mathbf{X}$ be a set of propositional variables, i.e. variables taking on truth values. We assume a standard propositional language for $\mathbf{X}$ containing the additional symbols TRUE and FALSE which are abbreviations of $X \vee \neg X$ and $X \wedge \neg X$ respectively; here X is any variable in $\mathbf{X}$. A *valuation* or *interpretation* of $\mathbf{X}$ is an assignment of truth values to the variables in $\mathbf{X}$. A valuation $\mathcal{M}$ is a *model* for a set of propositional formulas Γ (written as $\mathcal{M} \models \Gamma$) if Γ is true in $\mathcal{M}$; here truth of propositional formulas with respect to interpretations is defined as usual. Hence, from now on,'$\models$' will stand for standard propositional entailment, later it may also stand for standard first-order entailment.

Causal Theories and their Models

Here is our initial definition of causal theories:

Definition 9.1 *A propositional causal theory T is a 4-tuple* $T = \langle \mathbf{V}, \mathbf{U}, \text{EQ}, \text{CONS} \rangle$ *where*

1. $\mathbf{V} = \{X_1, \ldots, X_n\}$ *is a set of observable propositional variables*
2. $\mathbf{U} = \{U_1, \ldots, U_m\}$ *is a set of unobserved propositional variables*
3. EQ *is a set of* structural equations, *i.e. propositional formulas of the form*

$$X_i \equiv \Phi_i(X_1, \ldots, X_n, U_1, \ldots, U_m) \tag{9.1}$$

where Φ_i *is a formula involving zero or more of the variables* $X_1, \ldots, X_n$, $U_1, \ldots, U_m$. *For each* $X_i \in \mathbf{V}$, EQ *contains at most one structural equation with* X_i *on the left-hand side.*

4. CONS, *the set of* constraints, *is a finite set of propositional formulas over variables* $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$. *Here* $\mathcal{A}(\mathbf{V})$ *is defined as the set of propositional variables*

$$\{Do(X_i, \text{TRUE}), Do(X_i, \text{FALSE}) \mid X_i \in \mathbf{V}\}$$

Notice that expressions of the form '$Do(X_i, b)$' simply stand for propositional variables here. The set $\mathbf{U}$ can be used to model external influences we are ignorant about or that we consider unlikely; we will only need it in Section 9.5.4.

Example 9.2 Let $T = \langle \mathbf{V}, \mathbf{U}, \text{EQ}, \text{CONS}\rangle$ be a causal theory with $\mathbf{V} = \{Alive(0), Alive(1)\}$, $\mathbf{U} = \emptyset$, $\text{EQ} = \{Alive(1) \equiv Alive(0)\}$, $\text{CONS} = \{Alive(0), \neg Do(Alive(1), \text{FALSE}), \neg Do(Alive(1), \text{TRUE})\}$. The equation in EQ expresses that the value of *Alive* persists (from time 0 to 1) if there are no interventions; the formulas in CONS ensure that $Alive(0)$ should hold in all models of T and that there is no intervention at time 1.

We now introduce the notion of models for causal theories. Condition (1) below simply ensures that all constraints hold in all models; condition (2) implements the 'sufficient cause principle'.

Definition 9.3 *A causal model for a propositional causal theory* $T = \langle \mathbf{V}, \mathbf{U}, \text{EQ}, \text{CONS}\rangle$ *is a valuation* $\mathcal{M}$ *for the variables in* $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$ *such that*

1. $\mathcal{M} \models \text{CONS}$
2. *The restriction of* $\mathcal{M}$ *to the variables in* $\mathbf{V} \cup \mathbf{U}$ *is a model for the set of equations* EQ$'$, *i.e.* $\mathcal{M} \models \text{EQ}'$. *Here* EQ$'$ *is obtained from* EQ *and* $\mathcal{M}$ *as follows:*

   For all $X_i \in \mathbf{V}, b \in \mathbf{B}$ *such that* $\mathcal{M} \models Do(X_i, b)$, *we* delete *(if present) from* EQ *the equation* $X_i \equiv \Phi_i(\ldots)$ *and we* add *the equation* $X_i \equiv b$.

If $\mathcal{M}$ is a causal model for T and it is clear from the context that T is a causal theory rather than a set of propositional formulas, then we simply write '$\mathcal{M}$ is a model for T'.

Example 9.2, continued Let T be as in Example 9.2 above. Condition (1) of Definition 9.3 ensures that all causal models $\mathcal{M}$ for T have $\mathcal{M} \models \mathit{Alive}(0)$ and that for all $b \in \mathbf{B}$, $\mathcal{M} \models \neg Do(\mathit{Alive}(1), b)$. By condition (2), $\text{EQ}' = \text{EQ}$ for all models and so we also have $\mathit{Alive}(1)$ in all models for T: *Alive* persists. Now consider a theory T' identical to T except that CONS is now equal to $\text{CONS} = \{\mathit{Alive}(0), Do(\mathit{Alive}(1), \text{FALSE})\}$. By condition 1 of Definition 9.3 we now have $\mathcal{M} \models \mathit{Alive}(0) \wedge Do(\mathit{Alive}(1), \text{FALSE})$ for all causal models $\mathcal{M}$. By condition 2 of that definition, we have $\text{EQ}' = \{\mathit{Alive}(1) \equiv \text{FALSE}\}$ for all these models and thus all models for T' must have $\neg \mathit{Alive}(1)$.

9.4 Causal Theories Involving Persistence

Throughout this and the next chapter we consider domains in which two basic kinds of entities exist: *fluents*, which are properties of the world that, if no interventions take place, do not change over time, and *events*, which will stand for the 'triggers' of interventions. We use the term 'event' rather than 'action' here since our 'events' do not necessarily have to be performed by some agent – the difference however is not crucial in what follows. Causal theories for our domains of interest will thus be defined with respect to a set of fluents $\mathbf{F}$ and a set of events $\mathbf{E}$; we assume these sets to be finite. Specifically, from now on we assume that for any causal theory for the sets $\mathbf{E}$ and $\mathbf{F}$, the set of observables $\mathbf{V}$ can be partitioned into two subsets: $\mathbf{V_E}$, the set of *event – time pairs*, and $\mathbf{V_F}$, the set of *fluent – time pairs*. In this section and the next, we are only interested in the behaviour of our domains directly before and directly after some actions happen. We therefore restrict ourselves to *2-state causal theories* which represent domains using only two points in time: the initial point in time $t = 0$ and the final point in time $t = 1$. Given the set $\mathbf{F} = \{F_1, \ldots, F_n\}$, $\mathbf{V_F}$ can now be written as $\mathbf{V_F} = \{F_1(0), \ldots, F_n(0), F_1(1), \ldots, F_n(1)\}$. Here $F_i(t)$ denotes fluent i at time t. We assume that all actions we are interested in happen at the same time, namely directly after the initial point in time. Hence we do not have to index actions by time points and can simply let $\mathbf{V_E} = \mathbf{E}$.

We define a fluent literal to be an expression of the form F or $\neg F$ where F is some fluent in $\mathbf{F}$. The *initial state* of a causal model $\mathcal{M}$ for a causal theory for sets $\mathbf{E}$ and $\mathbf{F}$ is the set $\{F \mid \mathcal{M} \models F(0)\} \cup \{\neg F \mid \mathcal{M} \models \neg F(0)\}$. The *final state* of a causal model $\mathcal{M}$ is the set $\{F \mid \mathcal{M} \models F(1)\} \cup \{\neg F \mid \mathcal{M} \models \neg F(1)\}$.

Example 9.4 We now extend our turkey domain with a new fluent *Dark*. $\mathit{Dark}(t)$ will denote that it is dark at time t; shooting at the turkey will have its intended effect only if it is not dark (so that one can aim properly). For this, let T be a causal theory such that $\mathbf{V_E} = \{\mathit{Shoot}\}$, $\mathbf{V_F} = \{\mathit{Alive}(0), \mathit{Dark}(0), \mathit{Alive}(1), \mathit{Dark}(1)\}$; $\mathbf{U} = \emptyset$. EQ contains two equations: $\{\mathit{Alive}(1) \equiv \mathit{Alive}(0)\ ; \mathit{Dark}(1) \equiv \mathit{Dark}(0)\}$. CONS contains the single axiom

$$[\ \neg \mathit{Dark}(0) \wedge \mathit{Shoot}\ ] \supset Do(\mathit{Alive}(1), \text{FALSE}) \tag{9.2}$$

There are potentially four different initial and final states for T, corresponding to the four possible interpretations of *Alive* and *Dark*. Let us look at the set of valuations $\mathbf{M}$ in which no *Shoot*-event and no interventions take place; i.e. we have $\neg \mathit{Shoot}$

and $\neg Do(X, b)$ for any $X \in \mathbf{V}, b \in \mathbf{B}$. Clearly, for any member $\mathcal{M}$ of $\mathbf{M}$ we have $\mathcal{M} \models$ CONS; hence condition (1) of Definition 9.3 is satisfied. Since, according to condition (2) of that definition, the updated set of equations EQ$'$ is equal to EQ, the valuations in $\mathbf{M}$ which also satisfy this condition are clearly exactly those in which the values of both *Alive* and *Dark* persist. Thus $\mathbf{M}$ contains four models, each of which has the same initial and final state. Hence in models in which no interventions take place, we have persistence, which is in accord with intuition.

9.4.1 We need more...

Let us now look at the set of all causal models $\mathbf{M}'$ for T in which the *Shoot*-event *does* take place and in which it is not dark in the initial situation. Hence for all $\mathcal{M} \in \mathbf{M}'$ it holds $\mathcal{M} \models \neg Dark(0) \wedge Shoot$. By axiom (9.2) all $\mathcal{M} \in \mathbf{M}'$ have $\mathcal{M} \models Do(Alive(1), \text{FALSE})$. The set EQ$'$ for these models must therefore contain $Alive(1) \equiv$ FALSE so each $\mathcal{M} \in \mathbf{M}'$ also has $\mathcal{M} \models \neg Alive(1)$. But notice that there is also a model $\mathcal{M}' \in \mathbf{M}'$ with

$$\mathcal{M}' \models \neg Dark(0) \wedge Dark(1) \wedge Do(Dark(1), \text{TRUE}),$$

since both conditions (1) and (2) of Definition 9.3 are satisfied for $\mathcal{M}'$.

This is not what we would intuitively expect! Intuitively, we would reason as follows: (1) there are no interventions except those triggered by the *Shoot*-event; (2) *Shoot* does not affect *Dark*, so the value of *Dark* should remain unchanged after the *Shoot*-action.

What we have yet forgotten to model are the implicit assumptions behind (1) and (2), namely that (a) *there are no external interventions* and that (b) *actions affect no more things in the world than those we explicitly state they affect.* Notice that, unlike assumption (a) and the assumption of persistence, assumption (b) is *not* an assumption about the *physics* of our reasoning domains, i.e. it is not an assumption about how the world works. Rather it is an assumption about *what we really mean when we specify our domain knowledge in a certain way.* It allows us to specify domains in a compact way and as such corresponds to the second part of the distinction indicated in Chapter 8, Section 8.4 (page 198). In contrast, the persistence assumption (expressed by the axioms in EQ), the effect laws (expressed by the axioms in CONS) and the sufficient cause principle (expressed by the semantics of causal theories, i.e. Definition 9.3) are directly about the 'physics' of our domains: they describe how the domains under consideration really work, and as such belong to the first part of the distinction made on page 198.

In order to find a way to deal with assumption (b), we need to ask ourselves when exactly an 'intervention' takes place, or equivalently: under what conditions is a variable *set* to a value? Variables are always set to a value in a specific *context*: for example, $Alive(1)$ is set to FALSE only in a context in which *Shoot* and $\neg Dark(0)$ is true. Now suppose we are in one specific context; let us call it C. If there is a model $\mathcal{M}_1$ with this context and with $\mathcal{M}_1 \models Do(X_i, b)$ while there is also a model $\mathcal{M}_2$ with the same context C and with $\mathcal{M}_2 \models \neg Do(X_i, b)$, then we can be sure that nothing in our axioms states that in context C the fluent X_i is set to value b. By assumptions

(a) and (b) above, we should now prefer model $\mathcal{M}_2$. Apparently, we should partition our models into classes sharing the same context, and then within each such class pick the model which has $\neg Do(X_i, b)$ for as many X_i and b as possible.

Now what does it mean that two models 'have the same context'? Clearly, the context should contain at least *every fact in the world that can possibly influence whether or not an intervention takes place.* Since we assume (assumption (a)) that there are no external events, we may safely say that two models in which exactly the same events take place and the same fluents hold at the same time share the same context. Any two such models interpret all the variables in $\mathbf{V}$ and $\mathbf{U}$ the same; hence we should partition our models into equivalence classes where each class corresponds to one particular interpretation of the variables in $\mathbf{V} \cup \mathbf{U}$ and contains all models with that particular interpretation. For each equivalence class we should then pick the models which have a minimal interpretation (see below) of *Do*. But notice that we must do all this *before* we replace the structural equation set EQ by EQ′ (step 2 of Definition 9.3), since this replacement will only work if the *Do*-propositions already have the right interpretations in each model.

We thus have to make precise the notion of 'models that are minimal within a context'. First we need some notation: let $\mathbf{X}$ be any set of propositional variables; let $\mathbf{Y}$ be some subset of $\mathbf{X}$ and let $\mathcal{M}$ be any interpretation of the variables in $\mathbf{X}$. We write $\mathcal{M}\rfloor_{\mathbf{Y}}$ to denote the restriction of $\mathcal{M}$ to $\mathbf{Y}$, i.e. the set of assignments that $\mathcal{M}$ attaches to the variables in $\mathbf{Y}$.

Now let again $\mathbf{X}$ be any set of propositional variables; let $\mathbf{Y}$ and $\mathbf{Z}$ be subsets of $\mathbf{X}$ with $\mathbf{Y} \cap \mathbf{Z} = \emptyset$. Let $\mathcal{M}$ and $\mathcal{C}$ be two interpretations of the variables in $\mathbf{X}$ and let Γ be a set of formulas over $\mathbf{X}$.

Definition 9.5 *We call $\mathcal{M}$ a minimal model for Γ of the variables $\mathbf{Y}$ within context $\mathcal{C}\rfloor_{\mathbf{Z}}$ iff:*

1. $\mathcal{M} \models \Gamma$ *and* $\mathcal{M}\rfloor_{\mathbf{Z}} = \mathcal{C}\rfloor_{\mathbf{Z}}$.
2. *there is no* $\mathcal{M}'$ *with* $\mathcal{M}' \models \Gamma$, $\mathcal{M}'\rfloor_{\mathbf{Z}} = \mathcal{C}\rfloor_{\mathbf{Z}}$ *and*

$$\{Y \in \mathbf{Y} \mid \mathcal{M}' \models Y\} \subsetneqq \{Y \in \mathbf{Y} \mid \mathcal{M} \models Y\}$$

We will thus be looking for the models $\mathcal{M}$ for CONS with a minimal interpretation of $\mathcal{A}(\mathbf{V})$ in context $\mathcal{M}\rfloor_{\mathbf{V}\cup\mathbf{U}}$. It is important to realize that the minimization can never completely rule out any interpretations of $\mathbf{V}$ and $\mathbf{U}$: if $\mathcal{M} \models$ CONS, then there *must* exist some minimal model $\mathcal{M}'$ for CONS of $\mathcal{A}(V)$ with the same context, i.e. with $\mathcal{M}'\rfloor_{\mathbf{V}\cup\mathbf{U}} = \mathcal{M}\rfloor_{\mathbf{V}\cup\mathbf{U}}$

We call causal theories in which *Do* still has to be minimized *partially specified* (since it is only specified which fluents are affected by events and it is not specified which fluents are not). In both this chapter and the next, we will assume that all our causal theories are actually partially specified ones. This brings us to the final definition of 2-point causal theories and their models. Definition 9.6 defines 2-point causal theories as a special case of the general propositional causal theories as defined in Definition 9.1: each 2-point causal theory is also a propositional causal theory.

Definition 9.6 *A 2-point causal theory for the set of fluents* $\mathbf{F}$ *and the set of events* $\mathbf{E}$ *is a tuple* $T = \langle \mathbf{V}, \mathbf{U}, \text{EQ}, \text{CONS} \rangle$ *such that*

- $\mathbf{V} = \mathbf{V_F} \cup \mathbf{V_E}$ *with* $\mathbf{V_F} = \{F_i(0), F_i(1) \mid F_i \in \mathbf{F}\}$ *and* $\mathbf{V_E} = \mathbf{E}$.
- $\mathbf{U} = \{U_1, \ldots, U_m\}$ *is a set of unobserved propositional variables*
- $\text{EQ} = \{F_i(1) \equiv F_i(0) \mid F_i \in \mathbf{F}\}$
- CONS*, the set of* constraints*, is a finite set of propositional formulas over variables* $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$*. Here* $\mathcal{A}(\mathbf{V})$ *is defined as the set of propositional variables*

$$\{Do(F, \text{TRUE}), Do(F, \text{FALSE}) \mid F \in \mathbf{V}\}$$

Notice that the following definition only differs from Definition 9.3 in its first condition.

Definition 9.7 *A causal model for a 2-point causal theory* $T = \langle \mathbf{V}, \mathbf{U}, \text{EQ}, \text{CONS}\rangle$ *is a valuation* $\mathcal{M}$ *for the variables in* $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$ *such that*

1. $\mathcal{M}$ *is a minimal model for* CONS *of the variables* $\mathcal{A}(\mathbf{V})$ *within context* $\mathcal{M}\rfloor_{\mathbf{V}\cup\mathbf{U}}$.
2. *The restriction of* $\mathcal{M}$ *to the variables in* $\mathbf{V} \cup \mathbf{U}$ *is a model for the set of equations* EQ'*, where* EQ' *is obtained from* EQ *and* $\mathcal{M}$ *as follows:*

   For all $X_i \in X, b \in \mathbf{B}$ *such that* $\mathcal{M} \models Do(X_i, b)$*, we* delete *(if present) from* EQ *the equation* $X_i \equiv \Phi_i(\ldots)$ *and we* add *the equation* $X_i \equiv b$.

If $\mathcal{M}$ *is a causal model for a 2-point causal theory* T*, we write* $\mathcal{M} \models_{\mathbf{c}} T$.

We have now reached our first 'milestone'; that is, a theory that can be successfully applied.

9.5 The Power of Two-Point Causal Theories

2-point causal theories allow us to handle many interesting problem domains, specifically involving *ramifications* and actions with *disjunctive* or *non-deterministic* effects. In this section we will give several examples to illustrate this fact. We will briefly indicate how other approaches handle these examples and how our approach avoids some of their difficulties. Problem domains which involve more than two points in time have to wait until the next section, where our theory is extended to deal with them.

The key to all the examples presented in this section is that we are allowed the use of *Do* in domain constraints and effect axioms. Using *Do*, we are able to express the difference between an observation (e.g. $\neg Alive(1)$) and an intervention (e.g. $Do(Alive(1), \text{FALSE})$).

9.5.1 Ramifications

The *ramification problem* concerns the problem of concisely and correctly representing indirect effects of actions. It was introduced in Chapter 8, Section 8.2.4. The key to our handling of ramifications are axioms of the form

$$Do(F_1, b) \supset Do(F_2, b') \tag{9.3}$$

Such axioms can be interpreted as saying that *any event in the domain that sets the value of F_1 to b also sets the value of F_2 to b'*. Forms of (9.3) will be used in all the examples we are about to present.

Before turning to more complicated issues, we start with two standard examples involving ramifications: the extended 'walking turkey' problem and the 'two switches problem'.

Example 9.8 [**The Walking Turkey**] We introduced the 'walking turkey' in Chapter 8, Example 8.5, page 193. Recall that it was about a turkey that may be *Alive* or not and that may be *Walking* or not. In the basic version of the domain, we want to express that if a turkey stops being alive (for example, because it is shot at), then it should also stop walking: $\neg$ *Walking* is a *ramification* of $\neg$*Alive*. Later, a more complicated version of the domain was proposed by McCain and Turner [103]. They noted that if the turkey is not alive and somebody tries to make it walking (for example, by performing the action *Entice*), it will not suddenly become alive! *Alive* is *not* a ramification of the effect of *Entice*; rather, *Alive* is a *qualification* of *Entice* (by 'A is a qualification of B' we mean that the action B will have its usual effect only in situations in which A is the case). Clearly, a sentence of the form $\neg Alive(t) \supset \neg Walking(t)$ is not enough to express this knowledge, since it treats $Alive(t)$ and $\neg Walking(t)$ in an equivalent manner. Therefore it cannot represent the different 'dynamics' of *Alive* and $\neg$ *Walking*.

Let us formalize our domain. Consider the 2-point causal theory T_{WT} for the event and fluent sets $\mathbf{E} = \{Shoot, Entice\}$ and $\mathbf{F} = \{Alive, Walking\}$. $\mathbf{U} = \emptyset$. $\textsc{eq} = \{Alive(1) \equiv Alive(0)\ ;\ Walking(1) \equiv Walking(0)\}$. $\textsc{cons}$ consists of the following four axioms:

$$Shoot \supset Do(Alive(1), \textsc{false}) \tag{9.4}$$

$$Entice \supset Do(Walking(1), \textsc{true}) \tag{9.5}$$

$$\neg Alive(t) \supset \neg Walking(t) \tag{9.6}$$

$$Do(Alive(t), \textsc{false}) \supset Do(Walking(t), \textsc{false}) \tag{9.7}$$

Here axioms of the form $\phi(t)$ should be read as $\phi(0) \wedge \phi(1)$. At first sight, axiom (9.7) seems a consequence of (9.6). However, axiom (9.6) denotes a *static* domain constraint ('there can be no state in which a turkey is both walking and not alive') while axiom (9.7) denotes its corresponding 'dynamics'. Axiom (9.7) will allow us to infer that interventions that set *Alive* to false also set *Walking* to false. Note that it is an example of an axiom of form (9.3). By the minimization of *Do* in Definition 9.7, we will *not* be able to conclude that an intervention which sets *Walking* to true also sets *Alive* to true. We now show this in detail.

Let us denote by $\mathbf{M}$ the set of causal models for T_{WT} in which *Shoot* takes place while *Entice* does not; by axioms (9.4) and (9.7) we have $Do(Alive(1), \textsc{false})$ and $Do(Walking(1), \textsc{false})$ in all such models. By condition (1) of Definition 9.7, we also have $\neg Do(X_i, b)$ for all other (X_i, b). Hence, the updated set of structural equations $\textsc{eq}'$ for the models in $\mathbf{M}$ (Definition 9.6) becomes:

$$Alive(1) \equiv \textsc{false} \quad ; \quad Walking(1) \equiv \textsc{false}$$

Hence all models in $\mathbf{M}$ have as their final state $\{\neg Alive, \neg Walking\}$. We also see (by checking whether all the axioms in CONS and EQ$'$ hold) that $\mathbf{M}$ contains models for three different initial states: $\{Alive, Walking\}$, $\{Alive, \neg Walking\}$, $\{\neg Alive, \neg Walking\}$. So the case in which the domain constraints should entail a ramification of the *Shoot*-event works fine. Now for the models in which no *Shoot* but an *Entice* event takes place. All such models have $Do(Walking(1), \text{TRUE})$. By the minimization of *Do* in condition (1) in Definition 9.7 they also have $\neg Do(Alive(1), \text{TRUE})$. Hence the set EQ$'$ in condition (2) of that definition becomes

$$Alive(1) \equiv Alive(0) \quad ; \quad Walking(1) \equiv \text{TRUE}$$

By axiom (9.6) all models $\mathcal{M}$ must then also have $\mathcal{M} \models Alive(1)$, and, since $\mathcal{M} \models \text{EQ}'$ also $\mathcal{M} \models Alive(0)$. This means that there are no causal models for T_{WT} with both $\neg Alive(0)$ and *Entice*: *Entice* has *Alive* as an *implicit qualification*.

Our next example is due to Lifschitz' [97]. It is closely related to Lin's [99] 'suitcase problem' and to Ginsberg's 'stuffy room problem' [135].

Example 9.9 [The Suitcase & Switches Problem] Imagine a light that is connected to two switches; the light is only on if both of the switches are in the *on*-position: any event that puts a switch into the *on*-position in a context in which the other switch is on, will have as a ramification that the light goes on. However, if the event takes place in a context in which the other switch is not on, the light will not be affected. The main importance of this example is that some previous approaches cannot express the domain properly: they cannot rule out models in which, as a result of turning on one switch in a context in which the other one is on already, the other switch may jump into the *off*-position while the light remains out; see Lin [99] for details. We formalize the domain using event and fluent sets $\mathbf{E} = \emptyset$ and $\mathbf{F} = \{Swi_1, Swi_2, Light\}$. Swi_i denotes that switch i is in the *on*-position. Let T_{SW} be a causal theory for $\mathbf{E}$ and $\mathbf{F}$ with $\mathbf{U} = \emptyset$, EQ as usual and CONS as follows:

$$\begin{aligned}
[\ Swi_1(t) \wedge Do(Swi_2(t), \text{TRUE})\ ] &\supset Do(Light(t), \text{TRUE}) && (9.8)\\
[\ Swi_2(t) \wedge Do(Swi_1(t), \text{TRUE})\ ] &\supset Do(Light(t), \text{TRUE}) && (9.9)\\
Do(Swi_1(t), \text{FALSE}) &\supset Do(Light(t), \text{FALSE}) && (9.10)\\
Do(Swi_2(t), \text{FALSE}) &\supset Do(Light(t), \text{FALSE}) && (9.11)\\
(Swi_1(t) \wedge Swi_2(t)) &\equiv Light(t) && (9.12)
\end{aligned}$$

Here axiom (9.12) describes a domain constraint that must hold in all models of the domain; the other axioms describe the 'dynamics' of the domain. Let us see whether we get the expected models for T_{SW}: suppose first we turn on the first switch while the second one is off; i.e. we add the axiom $\neg Swi_2(0) \wedge Do(Swi_1(1), \text{TRUE})$ to CONS. By the minimization of *Do*, we get $\neg Do(Swi_2(1), b)$ for all b in all models. Hence EQ$'$ contains $Swi_2(1) \equiv Swi_2(0)$, so all models for T_{SW} must have $\neg Swi_2(1)$. But in all models with $\neg Swi_2(1)$, we already have (again by the minimization of *Do*) $\neg Do(Light(1), \text{TRUE})$ and $\neg Light(1)$. It follows that all models will have $\neg Light$ in their final state; it is easy to see that such models indeed exist. One can show in a similar manner that one obtains the intended models in all other possible scenarios.

9.5.2 A First Glimpse at Other Approaches

We have seen in the introduction that, in Pearl's terminology, the proposition $Do(X_i, b)$ can be read as 'there is a sufficient cause for X_i to take on the value b'. Indeed, usage of 'Do' in axioms often corresponds to the colloquial use of the word 'causes'. It turns out that if the word Do is replaced by the word `causes`, then our axioms start to look very similar to those used in various other approaches. We leave detailed comparisons for Chapter 10 and give only a suggestive example for the time being. This will serve to make clear that existing approaches solve the ramification problem in a quite similar, yet subtly different manner. The difference causes problems for domains involving 'causal cycles', which we are about to present.

In Thielscher's work, the equivalent of axiom (9.7) looks as follows ([151], page 330):

$$\neg Alive \texttt{ causes } \neg Walking$$

In Lin's work [99, 100], we would get:

$$Caused(Alive, \text{FALSE}, s) \supset Caused(Walking, \text{FALSE}, s)$$

In McCain & Turner's approach [103, 104, 105], it would be formalized as:

$$\neg Alive \Rightarrow \neg Walking \tag{9.13}$$

where '$\Rightarrow$' is to be read as 'if $\neg Alive$, then the fact that $\neg Walking$ is caused' [104]. McCain & Turner's semantics interprets (9.13) as something like

$$\neg Alive(t) \supset Do(Walking(t), \text{FALSE}) \tag{9.14}$$

In the turkey example, replacing (9.7) by (9.14) does not make any difference: it is easy to show that one obtains exactly the same models in both cases. However, as we will see below, in general, $(X \equiv b_1) \supset Do(Y, b_2)$ is *not* equivalent to $Do(X, b_1) \supset Do(Y, b_2)$.

9.5.3 Causal Cycles

We have just seen that $Do(X, \text{TRUE}) \supset Do(Y, \text{TRUE})$ may often be read as 'X causes Y'. As pointed out by Sandewall and Gustafsson & Doherty [66, 134], we do not always want to reason in the 'causal direction' that is implied by the above; for example, consider the switches domain (Example 9.9) again: if we know that the light has been put off, we may want to conclude that (at least) one of the two switches has been put off too. We model this as follows:

Example 9.10 Let $T_{\text{SW},2}$ be as T_{SW} but with the following axioms added to CONS:

$$\begin{aligned}
Do(Light(t), \text{FALSE}) &\supset [\ Swi_1(t) \supset Do(Swi_2(t), \text{FALSE})\ ] && (9.15)\\
Do(Light(t), \text{FALSE}) &\supset [\ Swi_2(t) \supset Do(Swi_1(t), \text{FALSE})\ ] && (9.16)\\
Do(Light(t), \text{TRUE}) &\supset Do(Swi_1(t), \text{TRUE}) && (9.17)\\
Do(Light(t), \text{TRUE}) &\supset Do(Swi_2(t), \text{TRUE}) && (9.18)
\end{aligned}$$

We show first that if we take CONS as above and do not add anything else, then we get persistence: one easily checks that for *any* interpretation $\mathcal{M}$ with $\mathcal{M} \models$ CONS, there is an interpretation $\mathcal{M}'$ that interprets all variables in $\mathbf{V} \cup \mathbf{U}$ the same way but which has $\mathcal{M}' \models \neg Do(X_i, b)$ for all X_i and b. So the minimization of Do in the definition of causal models will make sure that all models have $\neg Do(X_i, b)$ for all X_i and b. Hence $\text{EQ}' = \text{EQ}$ for all models and all fluents in our domain must persist.

Now suppose we turn on the light in an initial state with both switches *off*, i.e. we further add the following axiom to CONS:

$$Do(Light(1), \text{TRUE}) \wedge \neg Swi_1(0) \wedge \neg Swi_2(0) \tag{9.19}$$

Axioms (9.17) and (9.18) ensure that we have $Do(F(1), \text{TRUE})$ for $F \in \{Swi_1, Swi_2, Light\}$ in all models; so EQ' becomes $F(1) \equiv \text{TRUE}$ for all these F in all models, and all models get final state $\{Swi_1, Swi_2, Light\}$; it is easy to see that models with such a final state indeed exist, so we get exactly the result we wanted. What happens if we turn off the light will be discussed in Example 9.12.

The example above leads to a problem for some of the earlier causal approaches (i.e. those of Gustafsson & Doherty and McCain & Turner; see [66] for the details). The reason is that in these approaches, constructions similar to $X \supset Do(Y, \text{TRUE})$ are used at places where one should use $Do(X, \text{TRUE}) \supset Do(Y, \text{TRUE})$; see the remark at the end of Section 9.5.2. Had we chosen the first possibility, axioms (9.8) and (9.9) would have been replaced by the single axiom

$$[\ Swi_1(t) \wedge Swi_2(t)\ ] \supset Do(Light(t), \text{TRUE}) \tag{9.20}$$

while axioms (9.17) and (9.18) would have become:

$$Light(t) \supset [\ Do(Swi_1(t), \text{TRUE}) \wedge Do(Swi_2(t), \text{TRUE})] \tag{9.21}$$

and axioms (9.15) and (9.16) would have become:

$$\neg Light(t) \supset \\ [\ [Swi_1(t) \supset Do(Swi_2(t), \text{FALSE})] \wedge [Swi_2(t) \supset Do(Swi_1(t), \text{FALSE})]\ ] \tag{9.22}$$

By axioms (9.20) and (9.21) we would then get that each model with final state $S = \{Light\ ;\ Swi_1\ ;\ Swi_2\}$ also has $Do(F(1), \text{TRUE})$ for F equal to any of the three fluents. Since (a) there is nothing in our axioms contradicting models with final state S and (b) the minimization of Do cannot rule out particular interpretations of $\mathbf{V} \cup \mathbf{U}$, such models exist and are not ruled out by condition (1) of Definition 9.7. Therefore, for such models the set EQ' in condition (2) of Definition 9.7 would become $F \equiv \text{TRUE}$ for all three F and all three persistence relations would be broken: there is an 'automatic' intervention that sets their value to TRUE. It follows that no matter what the initial state is, there is always a model with the final state above: there is a cycle involved in that the fact that the light is on implies an intervention that puts both switches on, while the fact that the switches are on implies an intervention that puts the light on. We have already seen that in our own formalization (the one using axioms (9.8)-(9.12) and (9.15)-(9.18)) this cannot happen.

Thielscher ([151], page 329) solves the problem in a way similar to ours, but since his equivalent of *Do* is called `causes`, the corresponding rules look somewhat strange: he obtains rules like '*Light* `causes` Swi_1', while, intuitively, turning on the light does not 'cause' turning on the switch. Related observations have led some authors to claim that there is more to ramification than mere causal relationships – as stated in Gustafsson & Doherty, 'physical causality is simply one of several reasons one might set up dependencies between fluents' [66].

We think that our 'Pearlian' approach sheds some new light on this issue: we define causal relations (i.e. relations involving interventions) fully in non-causal terms; this is reflected in choosing the name *Do* rather than *Causes* for our interventions. Formulas of the form $Do(A, \textsc{true}) \supset Do(B, \textsc{true})$ may or may not correspond to the colloquial statement 'A causes B' or to any notion of 'physical' causality. The only thing we care about is how the world works at the level of detail at which we want to formalize it - and if at that level of detail, we have $Do(A, \textsc{true}) \supset Do(B, \textsc{true})$, then we may say that A causes B within the constraints of our domain – but if you do not like the word 'causes', you do not have to use it - the important thing about our theory is the semantics that it gives to interventions.

9.5.4 Disjunctive Effects and Nondeterminism

Example 9.10 raises the interesting question of how to formalize actions whose effect is a *disjunction* of fluents. Recently, some authors have, either explicitly or implicitly, begun to take up this question [105, 151]. A special case of this question concerns the more well-known issue of representing actions with non-deterministic effects [11, 105, 135]. If an event occurs that sets the value of a proposition X to TRUE, where $X \equiv Y \vee Z$, it is not immediately clear what should happen: should we exempt both Y and Z from persistence, or should we keep as much persistence as is logically consistent? In Example 9.10 we implicitly chose the latter option: setting *Light* to FALSE in an initial situation with both switches on will always keep one switch in the *on*-position: as will be shown below, there will be no final states with both $\neg Swi_1$ and $\neg Swi_2$. But is this correct, i.e. is it what we intuitively expect? A small poll conducted by the present author reveals that people have differing intuitions about this: some think that the position of both switches should be exempted from persistence, while others prefer the 'inert models' where there are as few changes as possible. We think that this clash of intuitions is not so surprising in light of Pearl's theory: if we intervene to set the value of a variable X that can be regarded as a disjunction of variables Y and Z, then we simply have an *incomplete specification* of our problem: it is not clear what should happen to the structural equations for Y and Z – should they both be removed from the set EQ, or should we remove as few equations as possible? In our view, this may change from domain to domain and should therefore be reflected in the domain axioms. We will therefore *not* attempt to extend the semantics of *Do* to constructions of the form $Do(Y \vee Z, b)$, but rather always make explicit what should happen to the structural equations of Y and Z if an intervention takes place. We can do so in our causal theories by using the *assumption symbols* in $\mathbf{U}$ (see Definition 9.1).

We illustrate their use by considering a domain[2] which clearly asks for eliminating all structural equations involved in the disjunction.

Example 9.11 We drop a pencil on a table with a piece of paper on it. As a result, the pencil may either lie fully on the paper, or touch both part of the paper and part of the table's surface, or touch only the table's surface. We write $\mathit{Touches\_table}(t)$ ($\mathit{Touches\_paper}(t)$) iff the pencil touches the surface of the table (the paper) at time t. We can model this using a causal theory with $\mathbf{E} = \{\mathit{Drop}\}$, $\mathbf{F} = \{\mathit{Touches\_table}, \mathit{Touches\_paper}\}$, $\mathbf{U} = \{U_1, U_2\}$, $\mathbf{V} = \mathbf{V_E} \cup \mathbf{V_F}$, EQ as in Definition 9.6 and CONS as follows:

$$\neg\mathit{Touches\_table}(0) \wedge \neg\mathit{Touches\_paper}(0) \tag{9.23}$$

$$\begin{aligned} \mathit{Drop} \;\supset\; [\;\; & (\; \mathit{Do}(\mathit{Touches\_table}(1), \textsc{true}) \wedge U_1 \;) \;\vee \\ & (\; \mathit{Do}(\mathit{Touches\_paper}(1), \textsc{true}) \wedge U_2 \;) \;\;] \end{aligned} \tag{9.24}$$

In this case, among the models with *Drop*, there will be three subclasses, corresponding to the three extensions of (U_1, U_2) that are consistent with (9.24). By the minimization of *Do* in Definition 9.7 and by axiom (9.24), we get the following interpretation for each class of models:

`class`	`corresponding interpretation of` *Do*
$U_1 \wedge U_2$	$\mathit{Do}(\mathit{Touches\_table}(1), \textsc{true}) \wedge \mathit{Do}(\mathit{Touches\_paper}(1), \textsc{true})$
$U_1 \wedge \neg U_2$	$\mathit{Do}(\mathit{Touches\_table}(1), \textsc{true}) \wedge \neg\mathit{Do}(\mathit{Touches\_paper}(1), \textsc{true})$
$\neg U_1 \wedge U_2$	$\neg\mathit{Do}(\mathit{Touches\_table}(1), \textsc{true}) \wedge \mathit{Do}(\mathit{Touches\_paper}(1), \textsc{true})$

(9.25)

It follows that there will be both models where only the persistence of *Touches_table* or *Touches_paper* is broken and models in which the persistence of both is broken.

Example 9.12 Let us now consider a case in which we want 'as much persistence as possible'. Let us say there is an intervention which sets the value of variable X where $X \equiv Y \vee Z$. If we let $X \equiv \neg\mathit{Light}$, $Y \equiv \neg\mathit{Swi}_1$ and $Z \equiv \neg\mathit{Swi}_2$, then we can see that we have already implicitly treated this case in the extended switches domain, Example[3] 9.10. Consider a scenario in the context of that example with an intervention setting *Light* to FALSE, i.e. we add the following axiom to the set CONS of $T_{\mathrm{SW},2}$:

$$\mathit{Do}(\mathit{Light}(1), \textsc{false}) \wedge \mathit{Light}(0)$$

We see that for the subclass of models with $\mathit{Swi}_1(1)$ and $\neg\mathit{Swi}_2(1)$ we have $\mathit{Do}(\mathit{Swi}_2(1), \textsc{false})$ (by axiom (9.15)) and $\neg\mathit{Do}(\mathit{Swi}_1(1), \textsc{false})$ (by the minimization of *Do*). For these models EQ$'$ contains $\mathit{Swi}_2(1) \equiv \textsc{false}$ and $\mathit{Swi}_1(1) \equiv \mathit{Swi}_1(0)$. It is now easy

[2]Example 9.11 is an adaptation of a scenario that, according to M. Shanahan [139], is due to R. Reiter.

[3]We would like to stress that we do not say that the switches domain *should* be formalized so as to keep 'as much persistence as possible' – we formalized it this way just for illustrative purposes, and we feel that one may equally well decide to formalize it along the lines of Example 9.11.

to see that we obtain a model with final state $\{\neg \mathit{Light}, \mathit{Swi}_1, \neg \mathit{Swi}_2\}$. Similarly, we can show there is a final state with $\neg \mathit{Swi}_1$ and Swi_2. But now suppose there is a model $\mathcal{M}'$ with final state containing $\neg \mathit{Swi}_1$ and $\neg \mathit{Swi}_2$. It is easy to check that by the minimization of *Do*, we must have $\neg \mathit{Do}(\mathit{Swi}_1(1), \textsc{false})$ and $\neg \mathit{Do}(\mathit{Swi}_2(1), \textsc{false})$ in such a model. This means the set $\textsc{eq}'$ will contain persistence relations for both Swi_1 and Swi_2. Since $\mathit{Swi}_1(0)$ and $\mathit{Swi}_2(0)$ must hold in all models, we must have $\mathcal{M}' \models \mathit{Swi}_1(1) \wedge \mathit{Swi}_2(1)$ too and we have a contradiction. So indeed only one of the switches will change position.

Example 9.13 [Exclusive Non-Deterministic Effects] The classic example of an action with a non-deterministic effect is the tossing of a coin: if we toss a coin, it will come up either heads (*Heads*) or tails, but we do not know which. However, we do know that whether or not we have *Heads* after tossing is independent of the fact whether or not we had *Heads* before tossing, thus there is no persistence. In light of the previous examples, this can clearly be modeled by a causal theory containing the following CONS:

$$\mathit{Toss} \supset [\ \mathit{Do}(\mathit{Heads}(1), \textsc{true}) \equiv \neg \mathit{Do}(\mathit{Heads}(1), \textsc{false})\ ]$$

In this way, in the models for our causal theory in which a *Toss*-event takes place, we either have an intervention that sets the value of *Heads* to TRUE or an intervention that sets its value to FALSE, but not both.

9.6 Handling many Time-points, Events & and Surprises

In this section we extend our 2-point causal theories to handle arbitrarily many points in time. This means we will have to move to a first-order language. In the subsections to come, we first (Section 9.6.1) show how first-order theories arise as a natural extension of 2-point theories. In Section 9.6.2 we introduce 'basic' first-order causal theories. We then instantiate these theories in more and more complicated ways: in Section 9.6.3, we start with simple instantiations for handling domains in which exactly one event happens at a time. Section 9.6.4 extends our theories to handle 'dependent fluents' which form yet another instance of the ramification problem. Then (Section 9.6.5) we move over to domains in which more than one event may happen at the same time and (Section 9.6.6) we show how this allows us to formalize causal chains of events. Finally, we extend our domains to handle complete surprises (Section 9.6.7).

9.6.1 From Propositional to First-Order Causal Theories

We want to extend propositional causal theories to handle countably many points in time. It seems that we would get an infinite number of structural equations '$F_i(t+1) = F_i(t)$', one for each fluent $F_i \in \mathbf{F}$ and one for each t. This suggests using predicate logic and universally quantifying our structural equations over time points. However, in that case Pearl's semantics gets undefined: it is not immediately clear how to perform a replacement of structural equations if the structural equations are quantified over. But it turns out that, by changing our formalism slightly, we can *mimic* the replacement of equations *within* causal theories. Once we have done this, the generalization to the first order case is completely straightforward. The general idea of this mimicking operation is based on the following translation:

For any propositional causal theory $T = \langle \mathbf{V}, \mathbf{U}, \text{EQ}, \text{CONS} \rangle$ defined according to Definition 9.1, let T' be the propositional theory over variables $\mathbf{U} \cup \mathbf{V} \cup \mathcal{A}(V)$ such that $T' = \text{CONS} \cup \text{EQ}_{\text{new}}$. Here EQ_{new} results from EQ by replacing any structural equation $X_i \equiv \Phi$ in EQ by the following three axioms:

$$[\ \neg Do(X_i, \text{TRUE}) \wedge \neg Do(X_i, \text{FALSE})\ ] \ \supset (X_i \equiv \Phi) \tag{9.26}$$
$$Do(X_i, \text{TRUE}) \supset X_i \tag{9.27}$$
$$Do(X_i, \text{FALSE}) \supset \neg X_i \tag{9.28}$$

Here the first axiom represents the original structural equation: if no interventions take place, then X_i should still be equivalent to Φ (and hence the structural equation applies). The second and third axioms represent the effect of actions. To give an example, if $\mathcal{M} \models Do(X_i, \text{TRUE})$ (and hence 'X_i is set to true in $\mathcal{M}$'), then the left-hand side of (9.26) does not hold in $\mathcal{M}$, and $\mathcal{M}$ automatically satisfies (9.26). On the other hand, the left hand side of (9.27) does hold in $\mathcal{M}$, hence $\mathcal{M}$ must satisfy $\mathcal{M} \models X_i$.

The following theorem shows that the notion of 'causal model' for theories $T = \langle \mathbf{V}, \mathbf{U}, \text{EQ}, \text{CONS} \rangle$ is simply equivalent to the notion of classical model for theories $T' = \text{CONS} \cup \text{EQ}_{\text{new}}$.

Theorem 9.14 *For any causal theory T and propositional theory T' obtained from T as described above, we have*

$$\mathcal{M} \textit{ is a model for causal theory } T \ \Leftrightarrow \mathcal{M} \models T'$$

The proof (which is almost trivial) can be found in [64].

The theorem shows that propositional causal theories can be equivalently modeled as 'plain' propositional theories. But these can be extended in a straightforward way to first-order theories:

Quantifying over Structural Equations

Let us suppose for the moment that we want to use causal theories for persistence involving $n+1$ points in time. Theorem 9.14 shows that in this case, all axioms in EQ can equivalently be represented by a set EQ_{new} which contains:

$$\begin{array}{lcl} \neg Do(F_i(1), \text{TRUE}) \wedge \neg Do(F_i(1), \text{FALSE}) & \supset & (F_i(1) \equiv F_i(0)) \\ \neg Do(F_i(2), \text{TRUE}) \wedge \neg Do(F_i(2), \text{FALSE}) & \supset & (F_i(2) \equiv F_i(1)) \\ & \vdots & \\ \neg Do(F_i(n), \text{TRUE}) \wedge \neg Do(F_i(n), \text{FALSE}) & \supset & (F_i(n) \equiv F_i(n-1)) \end{array} \tag{9.29}$$

for all $F_i \in \mathbf{F}$, with corresponding axioms

$$Do(F_i(t), b) \supset (F_i(t) \equiv b)$$

for all $F_i \in \mathbf{F}, b \in \mathbf{B}$ and $t \in \{0, 1, \ldots, n\}$.

It is now evident that this scheme of axioms can be extended to a countably infinite number of time points by universally quantifying over both groups of axioms and letting $Do(F_i(t), b)$ denote a predicate involving objects $F_i(t)$ and b rather than an atomic propositional variable. We thus end up with two classes of axioms. First,

$$\forall t > 0 \ . \ [\neg Do(F_i(t), \text{TRUE}) \wedge \neg Do(F_i(t), \text{FALSE})] \supset (F_i(t) = F_i(t-1)) \tag{9.30}$$

for all $F_i \in \mathbf{F}$ and second

$$\forall t \ . \ Do(F_i(t), \text{TRUE}) \quad \supset \quad (F_i(t) = \text{TRUE}) \tag{9.31}$$
$$\forall t \ . \ Do(F_i(t), \text{FALSE}) \quad \supset \quad (F_i(t) = \text{FALSE}) \tag{9.32}$$

Axioms of form (9.30) will be called *structural equation axioms*. Notice that they really contain an infinitude of structural equations. Axioms of form (9.32) will be called *intervention axioms*.

We will want to quantify not only over time points but also over fluents. This can easily be accomplished by introducing a new predicate Ho and writing $Ho(F_i, t)$ instead of $F_i(t)$, a common practice in common-sense temporal reasoning. This final extension immediately leads to the formal definitions of first-order theories that we will present in the next subsection.

9.6.2 First-Order Causal Theories

Preliminaries We use a many-sorted first-order language $\mathcal{L}$. Structures, interpretations, truth in a model and entailment are defined as usual; see Chapter 8, page 187. Specifically, from now on '$\models$' stands for first-order rather than propositional entailment. For a sort X indicated by the letter X, we write $|\mathcal{M}|_x$ to denote the universe of the sort X. The language $\mathcal{L}$ in turn depends on the sets $\mathbf{E}$ and $\mathbf{F}$. We therefore sometimes write $\mathcal{L}(\mathbf{E}, \mathbf{F})$. $\mathcal{L}(\mathbf{E}, \mathbf{F})$ contains three sorts: *Booleans* (variables of the sort will be denoted by b); *time points* (t) and *observables* (x). There are two 'subsorts' to observables: *fluents* (variables of the sort denoted by f) and *events* (e). We explain what we mean by 'subsort' below. The set $\mathbf{B}$ of Boolean constants contains two elements: $\mathbf{B} = \{\text{TRUE}, \text{FALSE}\}$. The set of time-point constants is $\mathbf{N}_0 = \{0, 1, 2, \ldots\}$, i.e. the set of nonnegative integers. The set of fluent constants coincides with $\mathbf{F}$; the set of event constants is identified with $\mathbf{E}$. $\mathcal{L}(\mathbf{E}, \mathbf{F})$ contains the following functions and predicates:

- Ho: $Ho(x, t)$ will denote that observable x *Ho*lds at time t.
- Do: $Do(x, b, t)$ will denote that the value of observable x at time t has been set to value b by some (unspecified) action.
- '$=, <, +$' which will receive their usual interpretation.

- Ab_1, Ab_2: $Ab_2(f,t)$ and $Ab_1(e,t)$ will stand for *abnormalities*. They will be used and explained only in Section 9.6.5.

By a 'subsort' we mean the following: events and fluents are really two different sorts. Whenever a predicate is defined for the sort of observables, it is really defined both for the sort events and the sort fluents. Whenever in a formula we quantify over x, for example, we have the axiom $\forall x \phi(x)$, we implicitly quantify over elements of both constituent sorts (i.e. the axiom should be read as $\forall e \phi(e) \wedge \forall f \phi(f)$). In what follows, we implicitly assume all those formulas that are listed without quantifiers to be universally quantified.

General First-Order Causal Theories

We are now ready to define first-order causal theories. In the appendix to this chapter (page 235), we show formally that the definitions below are indeed a straightforward extension of the propositional causal theories defined earlier.

Definition 9.15 *A first-order causal theory for a language $\mathcal{L}(\mathbf{E}, \mathbf{F})$ is a tuple $\langle \text{EQ}, \text{CONS} \rangle$ where*

1. EQ *is a set of sentences for $\mathcal{L}(\mathbf{E}, \mathbf{F})$ containing the 'intervention axioms'*

$$\forall x, t. \qquad Do(x, \text{TRUE}, t) \supset Ho(x,t) \tag{9.33}$$

$$\forall x, t. \qquad Do(x, \text{FALSE}, t) \supset \neg Ho(x,t) \tag{9.34}$$

   and, in addition, one or more axioms of the form

$$\forall\, x,t \;.\; [\ \neg Do(x, \text{TRUE}, t) \quad \wedge \quad \neg Do(x, \text{FALSE}, t)\ ] \ \supset \ [\ Ho(x,t) \equiv \ \Phi(x,t)\ ] \tag{9.35}$$

   Here x is a variable of sort events *or* fluents*; t is of sort time-points. We call the expression '$Ho(x,t) \equiv \Phi(x,t)$' a 'structural equation'.*

2. CONS *is a set of sentences for $\mathcal{L}(\mathbf{E}, \mathbf{F})$ containing at least uniqueness-of-names (UNA) and domain closure (DC) axioms[4] for the sets $\mathbf{B}, \mathbf{E}$ and $\mathbf{F}$.*

We will refer to the uniqueness-of-names axioms in CONS as the 'UNA-axioms' and to the domain closure axioms as the 'DC'-axioms.

We now give the definition of models for causal theories. Remember that in the propositional case, we were looking for the minimal interpretations of the set of *Do*-variables within each context, i.e. each interpretation of all other propositional variables of the theory. We will now do exactly the same thing in a first-order setting: now, we want the minimal interpretations of the *Do-predicate* for each context. Now, a context is any interpretation of all other *predicates* of the theory.

This 'minimization within a context' can be conveniently implemented using circumscription [94, 106]. For details about circumscription, we refer to [94]. In an

[4] A uniqueness-of-names (UNA) axiom for a finite set of constants $\mathbf{X} = \{X_1, \ldots, X_n\}$ is the formula $\bigwedge_{i \neq j} X_i \neq X_j$. A domain closure axiom for the set $\mathbf{X}$ is the axiom $\forall x\ x = X_1 \vee \ldots \vee x = X_n$.

appendix to this chapter (page 234) we give Lifschitz' characterization of circumscription in model-theoretic terms. From that definition, it can be seen immediately that circumscribing *Do* in CONS with all other functions and predicates *fixed* will give us exactly the models we want: if a predicate is kept fixed while circumscribing *Do*, each interpretation of the predicate will serve as a context within which *Do* is minimized. Also, just as in the propositional case, we have to minimize *Do* in CONS *before* we add the set of structural equations EQ to it – cf. the remark in Section 9.4.1. This is reflected in the following definition. The expression $Circum(\text{CONS}; Do)$ stands for the *circumscription of Do in* CONS *with all other functions and predicates kept fixed.*

Definition 9.16 *A structure $\mathcal{M}$ for the language $\mathcal{L}$ is a* model *for the first-order causal theory T (written as $\mathcal{M} \models_{\mathbf{c}} T$) iff*

1. $\mathcal{M} \models \text{EQ} \wedge Circum(\text{CONS}; Do)$.
2. *Time-points in $\mathcal{L}$ are interpreted as the integers; '+' and '<' are interpreted accordingly.*

Note that for any causal theory $T = \langle \text{EQ}, \text{CONS} \rangle$ and any $\mathcal{M}$ we have that if $\mathcal{M} \models_{\mathbf{c}} T$, then also $\mathcal{M} \models \text{CONS}$. Since by Definition 9.15 above CONS contains both UNA- and DC-axioms for $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{F}$, we easily see that the following proposition holds:

Proposition 9.17 *We may assume without loss of generality that for any model $\mathcal{M}$ for a causal theory T, we have*

1. $|\mathcal{M}|_b = \mathbf{B}$ *and* $|\mathcal{M}|_e = \mathbf{E}$ *and* $|\mathcal{M}|_f = \mathbf{F}$.
2. $\mathcal{M}$ *interprets all elements in* $\mathbf{B}$, $\mathbf{E}$ *and* $\mathbf{F}$ *as themselves.*

We now turn to the exact kind of rules of form (9.35) that we will need in order to model persistence. Fluents are supposed to behave as before: if no intervention takes place, they persist. This is modeled by the following axiom in EQ which we call our *persistence axiom*:

$$\begin{aligned}&\forall f, t\ .(t > 0) \supset \\ &\quad [\ \neg Do(f, \text{TRUE}, t) \ \ \wedge \ \ \neg Do(f, \text{FALSE}, t)\ ] \ \supset \ [\ Ho(f,t) \equiv \ Ho(f, t-1)\ ]\end{aligned} \tag{9.36}$$

which can easily be rewritten into the form prescribed by the definition of first-order causal theories above. Concerning events, we can opt for either one of two possibilities, The first is reminiscent of the way actions are treated in the standard situation calculus [107]: between each two 'time points', exactly one action happens. We can formalize this by adding an extra axiom to CONS ('$\exists!$' stands for 'there exists exactly one'):

$$\forall t \exists! e.\ Ho(e,t) \tag{9.37}$$

The formula $Ho(e,t)$ is to be interpreted as saying that event e takes place between time t and time $t+1$. The second, more complicated possibility is to allow for multiple actions happening at the same time; this is similar to what happens in the works of Morgenstern & Stein and Baral, Gelfond & Provetti [12, 147]. We first discuss the former possibility, delaying treatment of the latter until Section 9.6.5.

9.6.3 Handling Events like in the Situation Calculus

We first extend our definition of causal theories to incorporate axioms (9.36) and (9.37).

Definition 9.18 *A* first-order causal theory with persistence *for the tuple* $\langle \mathbf{E}, \mathbf{F} \rangle$ *is a tuple* $\langle \text{EQ}, \text{CONS} \rangle$ *where* EQ *and* CONS *are sets of sentences for the language* $\mathcal{L}(\mathbf{E}, \mathbf{F})$*,* EQ *consists of intervention axioms (9.33),(9.34) and the persistence axiom (9.36) while* CONS *contains at least UNA- and DC-axioms for* $\mathbf{B}, \mathbf{E}$ *and* $\mathbf{F}$ *and axiom (9.37).*

Example 9.19 [Yale Shooting Problem] Armed with this definition, we are able to handle most standard reasoning domains involving more than two time points. As an example, consider the Yale Shooting domain as described on page 190, where it was modeled in a 'naive' way. To model it instead as a causal theory, let T_{YSP} be a causal theory with persistence for sets $\mathbf{E} = \{\mathit{Load}, \mathit{Wait}, \mathit{Shoot}\}$, $\mathbf{F} = \{\mathit{Alive}, \mathit{Loaded}\}$. Apart from the axioms mentioned above, the set CONS further contains the following axioms:

$$\mathit{Ho}(\mathit{Load}, t) \supset \mathit{Do}(\mathit{Loaded}, \text{TRUE}, t+1) \tag{9.38}$$

$$\mathit{Ho}(\mathit{Loaded}, t) \wedge \mathit{Ho}(\mathit{Shoot}, t) \supset \mathit{Do}(\mathit{Alive}, \text{FALSE}, t+1) \tag{9.39}$$

$$\mathit{Ho}(\mathit{Alive}, 0) \wedge \neg \mathit{Ho}(\mathit{Loaded}, 0) \tag{9.40}$$

$$\mathit{Ho}(\mathit{Load}, 0) \wedge \mathit{Ho}(\mathit{Wait}, 1) \wedge \mathit{Ho}(\mathit{Shoot}, 2) \tag{9.41}$$

It is instructive to compare these axioms with the original axioms on page 190. Recall that with the original formulation, we obtained an unintended model in which Fred remained alive after the shooting. We will show that we now only get intended models. To see this, notice first that by axiom (9.37) we have that the only events taking place in *any* model at times $0, 1$ and 2 are those introduced in axiom (9.41). The circumscription of *Do* in the definition of causal models then makes sure that in all models of T_{YSP} we have $\neg \mathit{Do}(\mathit{Loaded}, b, t)$ for all b and $t \in \{0, 2\}$. On the other hand, by axioms (9.41) and (9.38) we have $\mathit{Do}(\mathit{Loaded}, \text{TRUE}, 1)$ in all models, too. It follows by axiom (9.33) that we have $\mathit{Ho}(\mathit{Loaded}, 1)$ in all models and by the persistence axiom (9.36) that we have $\mathit{Ho}(\mathit{Loaded}, 2)$ in all models. Since by axiom (9.41) we have $\mathit{Ho}(\mathit{Shoot}, 2)$ in all models, the antecedent of axiom (9.39) holds in all models for t instantiated to 2 and we have $\mathit{Do}(\mathit{Alive}, \text{FALSE}, 3)$ in all models. By axiom (9.36) we then have $\neg \mathit{Ho}(\mathit{Alive}, 3)$ in all models for T_{YSP}. It is easy to show that such models indeed exist.

We state without proof that we also handle the related 'Stanford Murder Mystery' [8]. The current version still has a problem if the schedule of actions that take place in the domain is incompletely specified. This problem will be discussed and solved in Section 9.6.5, where we show how to extend our theories for domains where arbitrarily many events may happen at the same time. First, we will discuss another well-known reasoning domain that we can already express with our current restrictions on the occurrence of events.

9.6.4 Ramifications Again: Dependent Fluents

Giunchiglia and Lifschitz [57] argue that we sometimes want to express dependencies between fluents where these dependencies may be partially unknown. Let us consider

Giunchiglia and Lifschitz' [57] motivating example: suppose we are in a room with a baby, an object and a table. The object may be dangerous to the baby (for example, it might be a hammer) but we are not sure about that. We do know however that if the object is placed on the table, it is out of reach of the baby, and hence it is safe. In other words, the safeness of an object depends on whether or not it is on the table, but we do not know exactly in what way: we know it is safe if it is on the table; if it is not on the table, we just know that this fact *determines* whether it is safe or not (see [57] for details). This implies that if we put an object on the table and then remove it again, we want there to be only two possibilities: either the object was safe both before and after it lay on the table or it was unsafe both before and after it lay on the table. Giunchiglia and Lifschitz introduce what they call the 'high-level action language $\mathcal{ARD}$' in order to deal with this kind of dependencies. It turns out that causal theories can be instantiated for 'dependent fluents' in a straightforward manner, the reason being that we are allowed the use of *assumption symbols* (unobserved variables) in causal theories.

We first extend our formalism by introducing the new subsort of *dependent fluents*; we will assume there are a finite number of them, listed in the set $\mathbf{D}$. Variables of the subsort will be indicated by d. Like regular fluents, dependent fluents may or may not persist. But unlike the situation for regular fluents, the behaviour of dependent fluents is fully defined in terms of other (regular) fluents and assumption symbols. We first extend our definitions of causal theories to deal with the subsort of dependent fluents. The language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F})$ is as the language $\mathcal{L}(\mathbf{E}, \mathbf{F})$ but now with dependent fluents as a new subsort of observables and with $\mathbf{D}$ indicating the constants of this new subsort.

Definition 9.20 *A* first-order causal theory with persistence and dependent fluents *for the tuple* $\langle \mathbf{D}, \mathbf{E}, \mathbf{F} \rangle$ *is a tuple* $\langle \text{EQ}, \text{CONS} \rangle$ *where* EQ *and* CONS *are sets of sentences for the language* $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F})$*,* EQ *consists of intervention axioms (9.33),(9.34) and the persistence axiom (9.36) while* CONS *contains at least UNA- and DC-axioms for* $\mathbf{B}, \mathbf{D}, \mathbf{E}$ *and* $\mathbf{F}$ *and axiom (9.37).*

Note that the only difference between causal theories as defined here and those defined as in the previous section (Definition 9.18) is the addition of UNA- and DC-axioms for dependent fluents. The new definition allows us to formalize the example described above:

Example 9.21 Let T_{DF} be a first-order causal theory with persistence for the language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F})$ containing the additional 0-ary predicate U_{Safe}. Here $\mathbf{D} = \{Safe\}$, $\mathbf{E} = \{Put\_On\_Table, Remove\_From\_Table\}$ and $\mathbf{F} = \{On\_Table\}$ and CONS contains the following additional axioms:

$$Ho(Put\_On\_Table, t) \supset Do(On\_Table, \text{TRUE}, t+1) \tag{9.42}$$

$$Ho(Remove\_From\_Table, t) \supset Do(On\_Table, \text{FALSE}, t+1) \tag{9.43}$$

$$Ho(Safe, t) \equiv [\ Ho(On\_Table, t) \vee (\neg Ho(On\_Table, t) \wedge U_{Safe})\ ] \tag{9.44}$$

U_{Safe} is called an *assumption symbol.* It determines the actual (but unknown) relationship between $\neg On\_Table$ and $Safe$. Suppose further that CONS contains the following observations:

$$\neg Ho(On\_Table, 0) \wedge Ho(Put\_On\_Table, 0) \wedge Ho(Remove\_From\_Table, 1) \qquad (9.45)$$

Notice that (9.45) together with (9.42) and (9.43) make sure that we have $Do(On\_Table, \text{TRUE}, 1)$ and $Do(On\_Table, \text{FALSE}, 2)$ in all models. The circumscription of Do in the definition of models for causal theories makes sure that we have $\neg Do(x, b, t)$ for all other x, b and $t \in \{0, 1\}$. It follows that all models for T_{DF} have

$$\neg Ho(On\_Table, 0) \wedge Ho(On\_Table, 1) \wedge \neg Ho(On\_Table, 2) \qquad (9.46)$$

Now in any model either U_{Safe} holds or it does not. For the class of models $\mathbf{M}$ with $\neg U_{Safe}$, we have by (9.44) and (9.46) that $\mathbf{M} \models \neg Ho(Safe, 0) \wedge \neg Ho(Safe, 2)$. For the class $\mathbf{M}'$ with U_{Safe}, we have $\mathbf{M}' \models Ho(Safe, 0) \wedge Ho(Safe, 2)$. It immediately follows that there can be no models with $Ho(Safe, 0) \equiv \neg Ho(Safe, 2)$. It remains to be shown that there do exist models for T_{DF} with U_{Safe} and models with $\neg U_{Safe}$; such models can indeed easily be constructed; we omit the details. Summarizing:

Proposition 9.22 *There exists a model $\mathcal{M}$ for T_{DF} with $\mathcal{M} \models Ho(Safe, 0) \wedge Ho(Safe, 1) \wedge Ho(Safe, 2)$. There is a model $\mathcal{M}'$ for T_{DF} with $\mathcal{M} \models \neg Ho(Safe, 0) \wedge Ho(Safe, 1) \wedge \neg Ho(Safe, 2)$. There are no models for T_{DF} with any other interpretation of $Ho(Safe, t)$ for $t \in \{0, 1, 2\}$.*

We remark that this allows us to specify what Giunchiglia and Lifschitz call 'non-Markovian' theories: the value of a fluent at time t may depend on its 'long-term' history, and not only on the situation in the world at time $t - 1$.

9.6.5 Minimizing Occurrence of Events

We have seen how to deal with domains in which we allow one event at a time to happen. Following Morgenstern & Stein [112, 147], Baral, Gelfond and Provetti [11, 12] and several other authors, we would like to extend this to the case where more than one event may happen at a time. We want our domains to be subject to the assumption that normally, events don't happen unless there is a specific reason for them to happen.

This is easy to model with our Pearlian theories: it turns out that we can properly model events, just like fluents, using structural equations. The advantage of using structural equations instead of axioms in CONS will become clear in Section 9.6.6. The new structural equations will say that 'if no intervention takes place that makes an event happen, and if nothing abnormal is the case, then the event will not happen'. The corresponding axiom in EQ will look as follows (the notation has been chosen to conform to (9.35)) :

$$\forall e, t \,.\, [\ \neg Do(e, \text{TRUE}, t) \ \ \wedge \ \ \neg Do(e, \text{FALSE}, t)\ ] \ \supset \ [\ Ho(e, t) \equiv \ Ab_1(e, t)\ ] \qquad (9.47)$$

This uses an 'abnormality predicate' $Ab_1(e, t)$ defined for all event-time pairs. An instantiated abnormality predicate plays a role similar to the 'assumption symbols' we have seen before. But unlike these, which represented things we were completely ignorant about, the abnormalities stand for things we consider abnormal, or, in other words, *unlikely*. For this, we extend the notion of causal model to 'preferred causal model'. We always prefer those models of our theory that are the least 'abnormal'. Since in the next subsection we will encounter domains involving both *rather* and *highly* abnormal eventualities, we need to introduce *two* abnormality predicates Ab_1 and Ab_2 in the definition below. Ab_1, standing for the 'weak' abnormality, is defined over event-time pairs (e, t). Ab_2, standing for 'strong' abnormality, over fluent-time pairs (f, t) (we will see how to use Ab_2 in the next section). The definition below says that the models with the smallest interpretations of the strong abnormality predicate Ab_2 should be preferred, and, among the remaining models, the models with the smallest interpretations of the weak abnormality predicate Ab_1.

Definition 9.23 *A model $\mathcal{M}$ is a preferred causal model for causal theory T if*

1. $\mathcal{M} \models_{\mathbf{c}} T$ *and*
2. *There is no other* $\mathcal{M}' \models_{\mathbf{c}} T$ *with* $\mathcal{M}'[Ab_2] \subsetneqq \mathcal{M}[Ab_2]$ *and*
3. *There is no other* $\mathcal{M}'' \models_{\mathbf{c}} T$ *with* $\mathcal{M}''[Ab_2] = \mathcal{M}[Ab_2]$ *and* $\mathcal{M}''[Ab_1] \subsetneqq \mathcal{M}[Ab_1]$

In the Epilogue to part III of this thesis (page 269) we show how the notion of 'preferred models' is connected to probability theory, and as such is already implicitly present in Pearl's original (probabilistic) theories. Having defined preferred models, we are in a position to give the definition of causal theories with concurrent events. The only difference to the previous definition (Definition 9.20) is that EQ must now also contain the 'no events'-axiom (9.47) while CONS does not contain the 'one-event-at-a-time' axiom (9.37) any more.

Definition 9.24 *A* first-order causal theory with persistence, dependent fluents and concurrent events *for the tuple $\langle \mathbf{D}, \mathbf{E}, \mathbf{F} \rangle$ is a tuple $\langle$*EQ, CONS*$\rangle$ where* EQ *and* CONS *are sets of sentences for the language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F})$,* EQ *consists of intervention axioms (9.33),(9.34), persistence axiom (9.36) and no-events axiom (9.47) while* CONS *contains at least UNA- and DC-axioms for $\mathbf{B}, \mathbf{D}, \mathbf{E}$ and $\mathbf{F}$.*

Note first that with this definition, we *still* handle standard reasoning domains like the Yale Shooting Problem:

Example 9.25 [YSP, continued] Let us consider the theory $T_{\text{YSP},2}$ which is as T_{YSP} but adapted to Definition 9.24: EQ now contains axioms (9.33), (9.34), (9.36) and (9.47), while CONS contains UNA- and DC-axioms and additionally axioms (9.38)–(9.41). One sees that the circumscription of CONS rules out all models with $Do(e, b, t)$ for any e, b and t. It follows by axioms (9.47) and (9.41) that all models for $T_{\text{YSP},2}$ have the abnormalities

$$Ab_1(\mathit{Load}, 0), Ab_1(\mathit{Wait}, 1), Ab_1(\mathit{Shoot}, 2) \tag{9.48}$$

Clearly, there will be no models with even more abnormalities: no more events will happen than those we specified to happen. It is easy to check that this means that in all preferred models for $T_{\text{YSP},2}$, Fred is not alive any more at $t = 3$. We can now also handle the case where at some points in time, no actions at all need to take place. If we replace (9.41) by the following axiom,

$$Ho(Load, 0) \wedge \; \exists t_1, t_2. \; [\; Ho(Wait, t_1) \wedge Ho(Shoot, t_2) \wedge t_1 > 0 \wedge t_2 > t_1 \;] \tag{9.49}$$

then we will still only prefer models $\mathcal{M}$ with, for some t_1 and $t_2 > t_1 > 0$:

$$\mathcal{M} \models Ho(Wait, t_1) \wedge Ho(Alive, t_2) \wedge Ho(Shoot, t_2) \wedge \neg Ho(Alive, t_2 + 1).$$

9.6.6 Causal Chains of Events

To see why it makes sense to put the axiom expressing the non-occurrence of events in EQ rather than CONS, we now turn to 'causal chains of events'. The idea here is very simple: suppose an event A 'causes' another event B, i.e. the event A is always accompanied by an intervention that triggers event B. Then, if the event A happens, we do not consider the event B 'abnormal' any more. The structural equation which says that 'event B occurring at time t is abnormal' is replaced by another equation that says 'event B does occur at time t' - just like structural equations concerning the persistence of regular fluents get replaced if an intervention takes place.

Example 9.26 Imagine a domain where, if you push somebody, he or she falls down a moment later. We can formalize this using a causal theory according to Definition 9.24 such that CONS contains the axiom

$$Ho(Push, t) \supset Do(Fall, \text{TRUE}, t + 1)$$

Now suppose CONS further contains axiom $Ho(Push, 0)$. From inspection of axiom (9.47) and the intervention axioms (9.33) and (9.34), we see that all preferred models for this theory will have $Ho(Push, 0) \wedge Ho(Fall, 1)$. But if CONS had not contained $Ho(Push, 0)$, then the preferred models would be those in which no events at all take place.

9.6.7 Surprise, Surprise

What if a fluent changes value while we have no event in our domain which can account for that? This is the kind of situation that Sandewall and Shoham [135] call a 'surprise'; Lifschitz and Rabinov [98] call it a 'miracle'. In order to deal with it, we have to weaken our structural equations concerning regular fluents: persistence may now be broken not only by some intervention, but also by some 'abnormal' (unlikely) external influence. For such abnormalities, we will use our second abnormality predicate Ab_2. We change axiom (9.36) into the following *weakened persistence axiom*:

$$\begin{aligned} &\forall f, t \;.\; (t > 0) \;\supset\; [\; (\neg Do(f, \text{TRUE}, t) \;\;\wedge\;\; \neg Do(f, \text{FALSE}, t)) \;\supset \\ &\quad [\; Ho(f, t) \equiv \;[\; (Ho(f, t-1) \wedge \neg Ab_2(f, t)) \vee (\neg Ho(f, t-1) \wedge Ab_2(f, t)) \;]\;]\;] \end{aligned} \tag{9.50}$$

The notation (9.50) has been chosen so as to conform to the syntactic form we used in (9.36). An equivalent, perhaps more intuitive way of stating it is by the following two axioms, whose conjunction is equivalent to (9.50):

$$[\ (t>0) \wedge \neg Do(f, \text{TRUE}, t) \wedge \neg Do(f, \text{FALSE}, t) \wedge \neg Ab_2(f,t)\ ] \supset [\ Ho(f,t) \equiv Ho(f,t-1)\ ]$$

$$[\ (t>0) \wedge \neg Do(f, \text{TRUE}, t) \wedge \neg Do(f, \text{FALSE}, t) \wedge Ab_2(f,t)\ ] \supset \neg[\ Ho(f,t) \equiv Ho(f,t-1)\ ]$$

In other words, if no interventions and no abnormalities take place, then the value of a fluent persists. If no interventions take place and the value of the fluent does not persist, then something abnormal is the case. Here is our updated definition:

Definition 9.27 *A* first-order causal theory with persistence, dependent fluents, concurrent events and surprises *for the tuple* $\langle \mathbf{D}, \mathbf{E}, \mathbf{F} \rangle$ *is a tuple* $\langle \text{EQ}, \text{CONS} \rangle$ *where* EQ *and* CONS *are sets of sentences for the language* $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F})$*,* EQ *consists of intervention axioms (9.33),(9.34),* weakened *persistence axiom (9.50) and no-events axiom (9.47) while* CONS *contains at least UNA- and DC-axioms for* $\mathbf{B}, \mathbf{D}, \mathbf{E}$ *and* $\mathbf{F}$.

We see that the only difference to the previous Definition 9.24 is that persistence axiom (9.36) has been replaced by its weakened version (9.50). We illustrate the use of 'surprises' by Kautz' Stolen Car Domain which we introduced on page 191.

Example 9.28 [stolen car problems] Here we consider two variations of this domain, modeled by theories $T_{\text{SC},1} = \langle \text{EQ}, \text{CONS}_1 \rangle$ and $T_{\text{SC},2} = \langle \text{EQ}, \text{CONS}_2 \rangle$. Both are causal theories according to Definition 9.27 for the tuple $\langle \mathbf{D}, \mathbf{E}, \mathbf{F} \rangle$ with $\mathbf{D} = \emptyset$, $\mathbf{E} = \{\mathit{Steal\_Car}\}$, $\mathbf{F} = \{\mathit{Car\_In\_Lot}\}$. On top of the UNA- and DC-axioms, CONS_1 and CONS_2 both contain observations:

$$Ho(\mathit{Car\_In\_Lot}, 0) \wedge \neg Ho(\mathit{Car\_In\_Lot}, 10)$$

CONS_2 contains no further axioms, while CONS_1 additionally contains:

$$Ho(\mathit{Steal\_Car}, t) \supset Do(\mathit{Car\_In\_Lot}, \text{FALSE}, t+1)$$

Now let t^* be some element of $\{0, \ldots, 9\}$ and consider a model $\mathcal{M}$ with

1. $\mathcal{M} \models Ho(\mathit{Steal\_Car}, t^*)$.
2. $\mathcal{M} \models Do(\mathit{Car\_In\_Lot}, \text{FALSE}, t^*+1)$ and $\mathcal{M} \not\models Do(x, b, t)$ for any other x, b, t.
3. $\mathcal{M} \models \forall f, t\ .\ \neg Ab_2(f,t)$
4. $\mathcal{M} \models Ho(\mathit{Car\_In\_Lot}, t+1) \equiv Ho(\mathit{Car\_In\_Lot}, t)$ for all t except $t = t^*$.

Clearly, $\mathcal{M} \models \mathit{Circum}(\text{CONS}_1, Do)$. It can be easily checked that $\mathcal{M} \models \text{EQ}$ too, so $\mathcal{M} \models_{\mathbf{c}} T_{\text{SC},1}$. Since $\mathcal{M}$ has no abnormalities of the Ab_2-kind, it follows that $\mathcal{M}$ will be preferred over any model which does have these. On the other hand, we have $\mathcal{M} \models Ab_1(\mathit{Steal\_Car}, t^*)$ but since there is no model which has neither Ab_1- nor Ab_2-abnormalities it clearly follows that $\mathcal{M}$ is a *preferred* model for $T_{\text{SC},1}$.

In $T_{\text{SC},2}$ things look different: there is no event which can account for the disappearance of the car. By arguments similar to those above, we find that all preferred models for $T_{\text{SC},2}$ do have $Ab_2(Car\_In\_Lot, t)$ for some $t \in \{1, \ldots, 10\}$.

So we see that in domains where a change of fluents happens for which there *is* an action in the domain that accounts for it, it will preferably be assumed that this action takes place than that some external 'miracle' or 'surprise' happens.

9.7 Conclusion

We have shown that by extending Pearl's causal theories we arrive at a powerful approach to common sense reasoning about action and change. We had to extend Pearl's theory at several places; however, the basic idea behind Pearl's theory, i.e. the sufficient cause principle, remained unchanged. The main ingredient of our causal theories is the *Do*-operator, which allows us to express any propositional combinations of observations (in the propositional case, these are elements of $\mathbf{V}$; in the first-order case, instances of *Ho*) and interventions (instances of *Do*).

In the introduction to this chapter we set ourselves three goals. We said that some of them would be dealt with in the present chapter and some in the next. Let us see what we have achieved this far:

Our first goal was to gain a better understanding of the rôle of causality in NMTR. Our contribution towards this goal was given mainly in Section 9.5. There we showed in detail that, while

$$Do(F_1, \text{TRUE}) \supset Do(F_2, \text{TRUE}) \tag{9.51}$$

can often be read as 'F_1 causes F_2', it has a semantics that can be understood fully in non-causal terms. We showed that this makes it possible to use axioms like (9.51) even in contexts where they do not correspond to the colloquial usage of 'causes'. We also showed how other approaches to NMTR, which use a predicate called *Caused* (or similarly) with a semantics very similar to *Do*, express relationships like (9.51) in a different way, sometimes leading to unintended results (Section 9.5.3). Finally, in Section 9.6.5 we showed that the *Do*-predicate with its 'sufficient cause principle' semantics can be used to model causal chains of events in a natural way.

The second goal was to provide a theory that could deal with a wide class of reasoning domains. Towards this goal, we have given many specific examples of reasoning domains that are problematic for many other approaches, and we have shown that our theory deals with them in an intuitive way. However, as we stressed in Chapter 8, Section 8.3.1, this does not give much confidence that our causal theories will work well on any full *class* of reasoning domains. Providing evidence that they do work well for full classes too is one of the two main goals of the next chapter.

The third goal, which is to show how existing approaches are interrelated and to explain some of their successes and failures, will be the second of the two main goals of the next chapter.

The Overall Research Goal To end this chapter, we indicate how our causal theories fit in the general 'alternative research goal' we set ourselves in Chapter 8, Section 8.4. Let us concentrate on the most sophisticated version of our theories, i.e. Definition 9.27 on page 232. Recall that the alternative research goal concerned a fundamental distinction between finding a good Model of a domain and finding a concise representation of that Model ('Model', when written with capital M, is used in the general sense; not in the logician's – see page 198). In the present causal theories, this distinction is adhered to as follows. We specify our causal theories by two sets of axioms: EQ and CONS. EQ contains axioms embodying persistence and the sufficient cause principle, CONS contains axioms standing for state constraints and for effects of events (e.g. $Ho(Shoot, t) \supset Do(Alive, \text{FALSE}, t+1)$) and of interventions (e.g. $Do(Alive, \text{FALSE}, t) \supset Do(Walking, \text{FALSE}, t)$). Nowhere do we mention any sentences about the non-effects of events; nevertheless there are many of these which we want to hold in all our models. For example, in the YSP domain (page 227) as modeled by theory T_{YSP}, we surely want the following to hold in *all* models:

$$\neg Ho(Shoot, t) \supset \neg Do(Alive, \text{FALSE}, t+1)$$

Yet this is neither enforced by CONS nor by EQ. Hence, T_{YSP} is an incomplete specification (in the sense of page 199) of the Model we want to specify by the theory T_{YSP}. T_{YSP} is turned into a complete specification in the first step of the definition of causal models (Definition 9.16), by circumscribing Do in CONS and then adding EQ. This first step performs the 'completion mapping' we defined on page 200. The set $\mathbf{M}$ of (logician's) models of $Circum(\text{CONS}; Do) \wedge \text{EQ}$ corresponds to our intended Model of the Yale Shooting domain. The minimization of abnormalities was done with respect to this set of models $\mathbf{M}$. It only served to select the set of *least surprising* models within $\mathbf{M}$; *all* models in the set $\mathbf{M}$ stand for *possible* realizations of the domain we want to model. We see that both uses of nonmonotonicity are strictly separated, as required by our alternative research goal: the circumscription of Do corresponds to the completion mapping, the preference order of abnormalities to the internal preference structure (page 200) of $\mathbf{M}$.

9.8 Appendix: The Model-Theoretic Characterization of Circumscription

The following is all taken from [94].

Let T be a first-order theory for some language $\mathcal{L}$; let P be a tuple of predicate constants for $\mathcal{L}$ and let Z be a tuple of function and/or predicate constants for $\mathcal{L}$ disjoint with P. For any two structures $\mathcal{M}_1$ and $\mathcal{M}_2$ for the language L, we write $\mathcal{M}_1 \leq^{P;Z} \mathcal{M}_2$ if

(i) $|\mathcal{M}_1| = |\mathcal{M}_2|$
(ii) $\mathcal{M}_1[K] = \mathcal{M}_2[K]$ for every constant K not in P, Z
(iii) $\mathcal{M}_1[P_i] \subseteq \mathcal{M}_2[P_i]$ for every P_i in P.

If $\mathcal{M}_1 \leq^{P;Z} \mathcal{M}_2$ but not $\mathcal{M}_2 \leq^{P;Z} \mathcal{M}_1$ we write $M_1 <^{P;Z} M_2$. We call a structure $\mathcal{M}$ *minimal* in a class M of structures if $\mathcal{M} \in$ M and there is no structure $\mathcal{M}' \in$ M such that $\mathcal{M}' <^{P;Z} M$.

The 'circumscription of P in T with Z varied', written as $Circum(T; P; Z)$ is defined as a formula in second-order logic (we will not repeat this formula here, see [106, 94]). However, we may also characterize circumscription as follows:

Proposition 9.29 (Lifschitz 1985) *A structure $\mathcal{M}$ is a model of $Circum(T; P; Z)$ iff $\mathcal{M}$ is minimal in the class of models of T with respect to $<^{P;Z}$.*

9.9 Appendix: Pearl's Causal Theories

We claimed at various places that our theories are simple extensions of Pearl's. However, none of our definitions appear anywhere in Pearl's articles. Hence, in order to convince the sceptical reader we will establish a formal relationship between Pearl's original theories and ours. Both our propositional and first-order causal theories for dealing with persistence have their roots in definitions 9.1 and 9.3, so it is enough if we can show how these definitions are formally related to Pearl's original definitions. That is what we will do in this appendix: we show in detail how exactly to extend Pearl's causal theories in order to arrive at definitions 9.1 and 9.3. We use the version of Pearl's theories introduced in [119, 120]. We first give Pearl's original definitions. We then show how a set of global constraints CONS is introduced; this gives us a new kind of causal theories that are more powerful than Pearl's. We then show that these are equivalent to the propositional causal theories introduced in Section 9.3. We end the appendix by saying something about the probabilistic ingredient in Pearl's theories, which is ignored in our analysis.

Propositional Causal Theories

Here is the definition of causal theories as given by Pearl [119, 120]:

Definition 9.30 *A causal theory T is a 4-tuple $T = \langle \mathbf{V}, \mathbf{U}, P(\mathbf{u}), \{f_i\}\rangle$ where*

1. *$\mathbf{V} = \{X_1, \ldots, X_n\}$ is a set of observed variables*
2. *$\mathbf{U} = \{U_1, \ldots, U_m\}$ is a set of unobserved variables which represent disturbances, abnormalities or assumptions.*
3. *$P(\mathbf{u})$ is a distribution over $U_1, \ldots, U_m$, and*
4. *$\{f_i\}$ is a set of n deterministic functions, each of the form*

$$X_i = f_i(X_1, \ldots, X_n, U_1, \ldots, U_m) \tag{9.52}$$

Equations of the form (9.52) are called structural equations.

Usually, causal theories come together with one or more *actions* of the form $Do(X_i, x)$. In the following, we assume that $\{X_{i_1}, \ldots, X_{i_l}\}$ is an arbitrary subset of the variables $\mathbf{V}$ and that for all $1 \leq k \leq l$, x_k is a value in the domain of X_{i_k}. Here is how Pearl defines the effect of actions [119, 120]:

Definition 9.31 *(Effect of actions) The effect of the set of actions*

$$\mathbf{A} = \{Do(X_{i_1}, x_1), Do(X_{i_2}, x_2), \ldots, Do(X_{i_l}, x_l)\}$$

on a causal theory T is given by a subtheory $T(\mathbf{A})$ of T, where $T(\mathbf{A})$ obtains by deleting from T all equations corresponding to the X_{i_k} occurring in $\mathbf{A}$ and substituting the equations $X_{i_k} = x_k$ instead.

The definition of *models* for causal theories is straightforward:

Definition 9.32 *A valuation $\mathcal{M}$ for the variables in $\mathbf{V} \cup \mathbf{U}$ belonging to a causal theory T is called a* model *of T iff all of the equations (9.52) associated with T hold in $\mathcal{M}$. A valuation $\mathcal{M}$ is called* a model for the causal theory T and the set of actions $\mathbf{A}$ *iff $\mathcal{M}$ is a model for $T(\mathbf{A})$, where $T(\mathbf{A})$ is defined as in Definition 9.31.*

It is often assumed [120] that the set of equations (9.52) has a unique solution for $X_i, \ldots, X_n$, given any value of the disturbances $U_1, \ldots, U_m$; in other words, each set of values for the disturbances determines a unique model for T. Therefore the distribution $P(\mathbf{u})$ induces a unique distribution on the set of variables $\mathbf{U} \cup \mathbf{V}$, or, equivalently, on the set of *models*: a model $\mathcal{M}$ such that $U_1 = u_1, \ldots, U_m = u_m$ will receive probability $P\{\mathcal{M}\} = P\{U_1 = u_1, \ldots, U_m = u_m\}$. We can define the notion of *preferred* model in terms of its probability:

Definition 9.33 *For any causal theory T, any valuation $\mathcal{M}$ of the variables in $\mathbf{V} \cup \mathbf{U}$ with maximum probability $P\{\mathcal{M}\}$ will be called a* preferred model *for the causal theory.*

For the time being we just look at plain models for causal theories, i.e. we do not care about the distribution $P(\mathbf{u})$. In that case, the assumption that the set of equations (9.52) has a unique solution for $X_i, \ldots, X_n$, given any value of the disturbances $U_1, \ldots, U_m$ is not needed. We return to the use of $P(\mathbf{u})$ and to 'preferred models' at the end of this appendix (page 238).

We refer to causal theories as defined above as *Pearlian* or *basic* causal theories. Causal theories which are such that all the variables in $\mathbf{V}$ and $\mathbf{U}$ are propositional will be called *propositional*. For simplicity, we will focus on these propositional theories when defining our extension. Here it is:

Definition 9.34 *An extended propositional causal theory T_{EP} is a 5-tuple*

$$T_{EP} = \langle \mathbf{V}, \mathbf{U}, P(\mathbf{u}), \{f_i\}, \textsc{cons} \rangle$$

such that

1. *$\langle \mathbf{V}, \mathbf{U}, P(\mathbf{u}), \{f_i\} \rangle$ is a propositional basic causal theory as defined in Definition 9.30.*

2. *$\textsc{cons}$, the set of* constraints, *is a finite set of propositional formulas over variables $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$. Here $\mathcal{A}(\mathbf{V})$ is defined as the set of propositional variables*

$$\{Do(X_i, \textsc{true}), Do(X_i, \textsc{false}) \mid X_i \in \mathbf{V}\}$$

For any extended propositional causal theory $T_{EP} = \langle \mathbf{V}, \mathbf{U}, P(\mathbf{u}), \{f_i\}, \textsc{cons}\rangle$, the associated basic causal theory will be called T^*_{EP}. If all the constraints in CONS hold for a valuation $\mathcal{M}$ of the variables in $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$ we will write $\mathcal{M} \models \textsc{cons}$. Notice that a T_{EP} is just a causal theory together with a set of formulas that may directly involve interventions. For such theories, we also need to extend the definition of models. The idea here is that we want to make sure that the axioms in CONS hold for all models of causal theories while still, the sufficient cause principle dictates how interventions should be handled.

Definition 9.35 *A model for an extended causal theory* $T_{EP} = \langle \mathbf{V}, \mathbf{U}, P(\mathbf{u}), \{f_i\}\ ,$ $\textsc{cons}\rangle$ *is a valuation* $\mathcal{M}$ *for the variables in* $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$ *such that*

1. $\mathcal{M} \models \textsc{cons}$
2. *The restriction of* $\mathcal{M}$ *to the variables in* $\mathbf{V} \cup \mathbf{U}$ *is a model of the basic causal theory* $T^*_{EP}(\mathbf{A})$. *Here* $T^*_{EP}(\mathbf{A})$ *is the effect of the set of actions* $\mathbf{A}$ *on the basic causal theory* T^*_{EP}, *where*

$$\mathbf{A} = \{Do(X_i, x) \mid \mathcal{M} \models Do(X_i, x), X_i \in \mathbf{V}, x \in \mathbf{B}\}$$

A preferred model for T_{EP} *is defined as a valuation* $\mathcal{M}$ *for variables in* $\mathbf{V} \cup \mathbf{U} \cup \mathcal{A}(\mathbf{V})$ *such that (1)* $\mathcal{M} \models \textsc{cons}$ *and (2) the restriction of* $\mathcal{M}$ *to* $\mathbf{V} \cup \mathbf{U}$ *is a preferred model of* $T^*_{EP}(\mathbf{A})$.

Note that the set $\mathbf{A}$ in item 2 of this definition depends on the model $\mathcal{M}$, i.e. it can be different for different $\mathcal{M}$.

We would like to show that the definitions of T_{EP} and its models are equivalent to the definitions of causal theories and their models used in the main text (definitions 9.1 and 9.3). However, there are still a few small differences, the most important one being that, in contrast to the theories of the main text, theories T_{EP} must have one structural equation with X_i on the left hand side for *every* X_i in $\mathbf{V}$. But this difference is not crucial, as we proceed to show:

For any extended causal theory $T_{EP} = \langle \mathbf{V}, \mathbf{U}, P(\mathbf{u}), \{f_i\}, \textsc{cons}\rangle$, let T'_{EP} be the theory that results from deleting from T_{EP} all functions f_i in $\{f_i\}$ of the form $X_i = U$ where U is some element of $\mathbf{U}$. It is easy to see the following:

Proposition 9.36 $\mathcal{M}$ *is a model for* T_{EP} *iff* $\mathcal{M}$ *is a model for* T'_{EP}.

Hence when working with extended propositional theories T_{EP}, we do not have to worry about specifying functions f_i for *all* the $X_i \in X$ – if an f_i is left out, it just means that we are indifferent about the corresponding X_i. But now notice that, since we are working with propositional variables, we can replace the equality sign '=' in the structural equations (9.52) by logical equivalence '≡' without changing their meaning. From this, together with the Proposition 9.36 above and the definition of effects (Definition 9.31) it immediately follows that the definitions of extended propositional causal theories T_{EP} and their models (definitions 9.34 and 9.35, resp.) are completely equivalent to the definitions of causal theories and models given in Section 9.3 (definitions 9.1 and 9.3, resp.). We have thus shown how these definitions arise as natural extensions of Pearl's definitions, which was our goal in this appendix.

9.9.1 Conclusion; the use of $P(\mathbf{u})$

We have seen how our propositional causal theories can be viewed as extensions of Pearl's theories. In Section 9.6, we lifted these theories to the first-order case. In the end this led to Definition 9.27 (page 232), the most sophisticated version of our causal theories. These were given a non-monotonic semantics in Definition 9.23: we prefer (roughly speaking) the models of such theories with the least number of abnormalities. This minimization of abnormalities is also already implicit in Pearl's original theories, where it can be handled by the probability distribution $P(\mathbf{u})$ that is implicit in causal theories but that we neglected thus far (see the remark below Definition 9.33). One can show that, if one assumes the probability distribution $P(\mathbf{u})$ to satisfy certain natural requirements, then the 'preferred models' of a causal theory (Definition 9.33) coincide with the models with the least abnormalities. We shall return to this issue in the Epilogue to part III of this thesis, where we show the relation between abnormality and probability.

Chapter 10

Causal Theories and Other Approaches

In the previous chapter we introduced 'causal theories'. In this chapter we give a detailed comparison of causal theories to three existing 'state-of-the-art' approaches to reasoning about action: those of McCain & Turner (Section 10.2), Lin (Section 10.3) and Baral, Gelfond & Provetti (Section 10.4). We briefly mention similarities to some other approaches (Section 10.5) and we end with some conclusions. The chapter is followed by two appendices containing proofs of the theorems and propositions presented throughout the chapter.

10.1 Introduction

In this chapter we give detailed comparisons of causal theories to other, existing approaches. This will serve two goals:

1. Provide evidence that our approach can truly handle a large class of reasoning domains.
2. Understand how existing approaches are inter-related; understand their successes and failures.

Both goals will be dealt with using a 'pragmatic' variation of the 'systematic methodology' introduced in Chapter 8, Section 8.3.1: in order to compare our approach to some alternative approach A we will show *formally* that it yields the same inferences as A on a large *class* of reasoning domains. In some cases we will also provide a domain that falls outside this class, and for which our approach gives more intuitive results than approach A. Such a result provides evidence (but not proof of course) that our approach gives intuitive results for a superset of the domains for which approach A gives intuitive results. In this way, goal (1) above will be achieved. The comparisons and formal correspondences will connect alternative approaches not only to our approach, but also to Pearl's 'sufficient cause principle'. This will shed a new light on their treatment of causality, and as such, goal (2) will be achieved.

10.1.1 How The Comparisons Will Be Done

We will consider three approaches in detail: those of McCain & Turner (Section 10.2), Lin (Section 10.3) and Baral, Gelfond & Provetti (Section 10.4). Each of these will be compared to the instantiation of our theory that is conceptually closest to it. Hence McCain & Turner's approach, which has been specifically designed to handle the ramification problem, will be compared to our 2-point causal theories (page 214) which are complex enough to deal with ramifications but which contain no additional features that would only serve to complicate the comparison. In the same vein, Lin's approach will be compared to first-order theories for persistence in which only one event may happen at a time (page 225), and Baral, Gelfond & Provetti's method will be compared to the version of our theory that minimizes occurrences of events but that does not handle surprises (page 229).

In the case of McCain & Turner and Baral, Gelfond and Provetti we give and prove 'equivalence theorems' stating that for large classes of reasoning domains these approaches give the same results as our causal theories do. We will also give examples of reasoning domains that fall outside these classes for which, we claim, our approach works better than the one we are comparing it to.

In order to state our equivalence theorems we have to define 'correspondence relations' which we denote by $\sim$. We write $T_A \sim T_B$ iff domain descriptions T_A of approach A and T_B of approach B intuitively encode the same domain knowledge. $T_A \sim T_B$ should be pronounced as 'theory T_A *syntactically corresponds* to theory T_B'. We will define two correspondence relations. In the first correspondence relation, A will be our two-point causal theories while B will be McCain & Turner's approach. In the second relation, A will be our first-order causal theories while B will be Baral, Gelfond & Provetti's approach. All approaches we consider have as their basic objects fluents and events; proving that A and B are equivalent then amounts to showing that for all corresponding theories T_A and T_B (i.e. $T_A \sim T_B$) we have that approach A selects a model $\mathcal{M}_A$ with a particular history of what fluents and events hold at what time iff B selects a model $\mathcal{M}_B$ with the same interpretation of event/fluent-time pairs. We will say that such models *semantically correspond*, written as $\mathcal{M}_A \cong \mathcal{M}_B$. Since for many well-formed theories of both approaches, $\sim$ will not be defined, $\sim$ implicitly imposes constraints on the class of reasoning domains for which the equivalence holds.

The importance of the equivalence theorems is that they show that some approaches which at first glance look rather different from ours are actually quite similar; therefore they are also connected to Pearl's sufficient cause principle, albeit implicitly. In our comparison to Lin's [99, 100] approach, we have not bothered to try and prove an equivalence theorem, since it is easy to see, just by looking at the definitions, that his approach is almost equivalent to ours.

10.2 McCain & Turner's Theory of Ramifications and Qualifications

Recently, McCain & Turner (MT) have introduced a 'causal theory' that focuses on handling ramification constraints [103, 105, 104]. We will now compare their approach to our two-point causal theories as defined by Definition 9.6 and 9.7 on page 214.

MT consider theories that are triplets (S, E, C), defined for a propositional language where each atom stands for a fluent. The 'state of the world' S is an interpretation for all fluents. An interpretation is denoted by the set of literals true in it. E is a set of 'explicit effects', i.e. propositional combinations of fluents. Intuitively, they are the formulas that are explicitly caused to hold by some (unspecified) action. C is a set of 'causal laws' that determine the ramifications of effects. MT define the function[1] $\Pi_1(S, E, C)$ such that it gives the set of possible states of the world after an action with effects E has taken place in state S. The exact definition of $\Pi_1(S, E, C)$ can be found in Appendix 10.7.1; here we just give an example of its use:

$$\begin{aligned} S &= \{\mathit{Alive}, \mathit{Walking}\} \\ E &= \{\neg \mathit{Alive}\} \\ C &= \{\neg \mathit{Alive} \Rightarrow \neg \mathit{Walking}\} \end{aligned} \tag{10.1}$$

In this case, $\Pi_1(S, E, C) = \{\{\neg \mathit{Alive}, \neg \mathit{Walking}\}\}$ i.e. the change of *Alive* brought about a change of *Walking*. However, if we had had

$$S' = \{\neg \mathit{Alive}, \neg \mathit{Walking}\}, E' = \{\mathit{Walking}\}, C' = C,$$

then $\Pi_1(S', E', C')$ would have been empty: '$\Rightarrow$' has a function similar to our '*Do*', enabling changes of right-hand side fluent values given changes of left-hand side values, but *not* the other way around (compare this to Example 9.8 of Chapter 9). There is one sort of domain constraint that can be expressed in MT's approach but not in ours: MT allow effects to be *any* propositional combination of fluents, and thus an effect may be a disjunction of two fluents. This cannot be expressed by our 2-point causal theories (we have no construct of the form $Do(X \vee Y, b)$). But it is exactly here that MT can give counterintuitive results; to see this, consider the general case where there is an effect X that further causes $Y \vee Z$, and an initial state with $\neg X, \neg Y$ and $\neg Z$:

$$\begin{aligned} S &= \{\neg X, \neg Y, \neg Z\} \\ E &= \{X\} \\ C &= \{X \Rightarrow (Y \vee Z)\} \end{aligned}$$

Proposition 10.1 $\Pi_1(S, E, C) = \{\{X, Y, \neg Z\}, \{X, \neg Y, Z\}\}$

The proof of this proposition can be found in Appendix 10.7.2. We have seen in Example 9.11 (Chapter 9, Section 9.5.4) that this seems too strong in general.

Another difference between MT's approach and ours has already been pointed out in Section 9.5.2 of the previous chapter: there are domain constraints that would be modeled as a single axiom $A \Rightarrow B$ in MT's approach, while we prefer to model them using two separate axioms:

$$\begin{aligned} Do(A(t), \text{TRUE}) &\supset Do(B(t), \text{TRUE}) \\ A(t) &\supset B(t) \end{aligned} \tag{10.2}$$

[1] $\Pi_1(S, E, C)$ is the function McCain & Turner use in their recent paper [105]; it stands at the basis of their 'Causal Theory of Action and Change' [104]. It is a slight modification of their earlier next-state function $Res^4_C(E, S)$ [103]. For the precise relation between $\Pi_1(S, E, C)$ and $Res^4_C(E, S)$, see [105].

McCain & Turner's semantics treats '$A \Rightarrow B$' just as we would treat the single axiom:

$$A(t) \supset Do(B(t), \text{TRUE}) \tag{10.3}$$

We have seen in Chapter 9, Section 9.5.3 that this is in general *not* the same as (10.2).

But *if* we translate constraints like $A \Rightarrow B$ indeed as (10.3), and we restrict ourselves to domains without disjunctive effects, then it turns out that MT and our approach agree on all problem domains that can be represented in the languages of both approaches. In Definitions 10.2 and 10.3 (page 243) syntactic ($\sim$) and semantical ($\cong$) correspondence for 2-point causal theories and MT's theories are defined. $\sim$ has been defined such that disjunctive effects cannot occur in corresponding theories and such that rules of the form '$A \Rightarrow B$' are translated into sentences of the form (10.3). $\cong$ is defined such that $\mathcal{M} \cong (S, S')$ iff the fluents that hold in S hold in $\mathcal{M}$ at time 0 and the fluents that hold in S' hold in $\mathcal{M}$ at time 1. As an example of how Definition 10.2 works, we consider the causal theory (S, E, C) defined by (10.1) at the beginning of this section. The theory described there corresponds to a 2-point causal theory T_{C} (the subscript C stands for 'Causal') such that CONS contains the axioms:

$$Alive(0) \wedge Walking(0)$$
$$Do(Alive(1), \text{FALSE})$$
$$\neg Alive(1) \supset Do(Walking(1), \text{FALSE})$$

which can be seen to stem from items 3,4 and 5 of Definition 10.9, respectively. EQ contains

$$Alive(1) \equiv Alive(0) \ ; \ Walking(1) \equiv Walking(0)$$

which can be seen from item 1 of the definition.

We are now ready to state our equivalence theorem, the proof of which can be found in Appendix 10.7.3. In the theorem, $\models_{\mathbf{c}}$ is defined as in Definition 9.7.

Theorem 10.4 *For any 2-point causal theory T_{C} and any domain description $T_{\text{MT}} = (S, E, C)$ such that $T_{\text{C}} \sim T_{\text{MT}}$, we have:*

$$S' \in \Pi_1(S, E, C) \Rightarrow \text{ there exists an } \mathcal{M} \models_{\mathbf{c}} T_{\text{C}} \text{ with } \mathcal{M} \cong (S, S')$$

$$\mathcal{M} \models_{\mathbf{c}} T_{\text{C}} \Rightarrow \Pi_1(S, E, C) \text{ contains an } S' \text{ with } \mathcal{M} \cong (S, S')$$

10.3 Lin's Embrace of Causality

Lin [99, 100] has recently introduced a new method for reasoning about action that is based on a version of the situation calculus [107]. In the situation calculus, time is modeled by *situations s* rather than time points. The sort *actions* corresponds to our *events*; however, in Lin's approach symbols of the sort are not necessarily constants (they can be n-ary function symbols rather than only 0-ary). Similarly, fluents in situation calculus correspond to our fluents, and again, Lin allows fluent symbols to

For $F_i \in \mathbf{F}$ we will sometimes write F_i^{TRUE} to denote F_i and F_i^{FALSE} to denote $\neg F_i$.

Definition 10.2 *For any 2-point causal theory* $T_{\mathrm{C}} = \langle \mathbf{V}, \mathbf{U}, \mathrm{EQ}, \mathrm{CONS}\rangle$ *for sets* $\mathbf{E}$ *and* $\mathbf{F}$ *and any* $T_{\mathrm{MT}} = (S, E, C)$ *of MT's approach we define* $T_{\mathrm{C}} \sim T_{\mathrm{MT}}$ *('T_{C} corresponds to T_{MT}') to be true iff all of the following hold:*

1. $\mathbf{E} = \emptyset$; $\mathbf{F} = \{F_1, \ldots, F_n\}$; $F_i \in \mathbf{F}$ *iff* F_i *is an atom in the propositional language for which* T_{MT} *is defined;* $\mathbf{U} = \emptyset$; EQ *is defined as required by Definition 9.6, page 214.*
2. *Each sentence in* CONS *corresponds either to* S *or to a sentence in* E *or to a sentence in* C. *The set* S *and the sentences in* E *and* C *all correspond to a sentence in* CONS. *Here 'correspondence' is defined as in items 3-5 below:*
3. *the sentence* Γ_{C} *in* CONS *corresponds to* S *(and vice versa) iff* Γ_{C} *is of the form* $F_1^{b_1}(0) \wedge \ldots F_n^{b_n}(0)$ *and* $S = (F_1^{b_1}, \ldots, F_n^{b_n})$.
   Here $b_i \in \mathbf{B}$ *for all* $1 \leq i \leq n$. .
4. *A sentence* Γ_{C} *in* CONS *corresponds to a sentence* Γ_{MT} *in* E *iff* Γ_{C} *is of the form* $Do(F_i(1), b)$ *and* Γ_{MT} *is of the form* F_i^b. *Here* $b \in \mathbf{B}$.
5. *A sentence* Γ_{C} *in* CONS *corresponds to a sentence* Γ_{MT} *in* C *iff* Γ_{C} *is of the form* $(m \geq 1)$

$$\Phi \supset Do(F_{i_1}(1), b_1) \wedge \ldots \wedge Do(F_{i_m}(1), b_m) \tag{10.4}$$

   and Γ_{MT} *is of the form*

$$\Phi' \Rightarrow F_{i_1}^{b_1} \wedge \ldots \wedge F_{i_m}^{b_m} \tag{10.5}$$

   Here Φ *is a propositional combination of atoms of the form* $F(1)$ *where* F *is any element of* $\mathbf{F}$. Φ' *is the result of replacing all occurrences of* $F(1)$ *by* F. *For all* j, $1 \leq i_j \leq n$ *and* $b_j \in \mathbf{B}$.

Definition 10.3 *Given a model* $\mathcal{M}_C$ *for a causal theory* T_{C} *and a pair of states* (S, S') *for a* T_{MT} *with* $T_{\mathrm{C}} \sim T_{\mathrm{MT}}$ *we say that* $\mathcal{M}_C$ *corresponds to* (S, S') *iff for all* $F_i \in \mathbf{F}$ *we have*

$$\mathcal{M}_C \models F_i(0) \Leftrightarrow F_i \in S \quad \text{and } \mathcal{M}_C \models F_i(1) \Leftrightarrow F_i \in S'$$

Lin's Procedure Start with a theory T_{LIN} that consists of (1) unique names axioms for all actions and fluents (2) unique names and domain closure-axioms for truth-values and (3) all domain-specific axioms. Then:

1. Add the following basic axioms for the predicate *Caused* to T_{LIN} :

$$Caused(f, \text{TRUE}, s) \supset Ho(f, s) \tag{10.6}$$
$$Caused(f, \text{FALSE}, s) \supset \neg Ho(f, s) \tag{10.7}$$

2. Circumscribe *Caused* in T_{LIN} with all other predicates fixed. Let T'_{LIN} be the resulting theory.
3. Add to T'_{LIN} the following 'frame axiom' and let T''_{LIN} be the resulting theory.

$$\begin{aligned} &Poss(a, s) \supset \\ &\quad \{(\neg Caused(f, \text{TRUE}, Result(a, s)) \wedge \neg Caused(f, \text{FALSE}, Result(a, s))) \\ &\qquad\qquad \supset [Ho(f, Result(a, s)) \equiv Ho(f, s)]\} \end{aligned} \tag{10.8}$$

4. Maximize *Poss* in T''_{LIN} to obtain the final theory T'''_{LIN} .

be n-ary. For any action e and situation s, the function[2] $Result(e, s)$ stands for the situation that results when performing e in s. Ordinary situation calculus contains only the functions described above and the predicate *Ho*, defined on fluent-situation pairs. In addition to this, Lin uses two additional predicates *Caused* and *Poss*.

It turns out that Lin's method is *very* similar to ours. We will compare it to the instantiation of causal theories in which only one event is allowed to happen at a time (Chapter 9, Definition 9.18), since this instantiation is conceptually closest to the situation calculus. Like us, Lin uses an additional sort *truth values*; a variable of the sort can be either TRUE or FALSE. Lin's *Caused*-predicate is ternary: $Caused(f, v, s)$ is true if the fluent f is caused to have the truth value v in situation s. On page 244 we give a (somewhat extended) quote of [99] which describes the method. The last step of Lin's method is meant to deal with the qualification problem which we do not address in our work, so it will be of no concern to us. We first need the following:

Proposition 10.5 *If we exchange step 1 and step 2 in Lin's procedure we will arrive at an equivalent final theory* T'''_{LIN} .

Proof: If we rewrite (10.6) and (10.7) as disjunctions, then *Caused* appears only in negated form in them. It then easily follows from the model-theoretic characterization of circumscription (see Appendix 9.8) that the theory obtained by adding (10.6) and (10.7) before the circumscription has the same models as the theory obtained by adding them after the circumscription. □

[2]The actual name for the function used by Lin is '*Do*' in stead of *Result*; we use the name *Result* in order to avoid confusion with our own *Do-predicate*.

Let us write $\text{EQ}_{\text{LIN}} = \{(10.6) \cup (10.7) \cup (10.8)\}$. With this notation, step 1 and step 2 interchanged and step 4 left out, Lin's approach can be rephrased as follows:

$$T''_{\text{LIN}} = \text{EQ}_{\text{LIN}} \cup \mathit{Circum}(T_{\text{LIN}}\ ; \mathit{Caused}) \tag{10.9}$$

Now if we compare EQ_{LIN} to the set of axioms EQ in our definition of causal theories (Definition 9.18, page 227), we see that they are strikingly similar: Once we rename *Caused* to *Do*, (10.6) and (10.7) actually become equivalent to our intervention axioms (9.33) and (9.34)! If we furthermore realize that $\mathit{Result}(a, s)$ refers to the first time point considered after the time point corresponding to situation s, we see that also (10.8) is almost the same as our persistence axiom (9.36). And most importantly, if we compare the characterization of Lin's approach (10.9) to the definition of models for causal theories (Definition 9.16, page 226), we see that, if EQ is as in Definition 9.18, then (10.9) above is nearly equivalent to Definition 9.16, item 1. The differences are that our approach uses integer time while Lin's uses situations and that our approach does not handle the qualification problem and infinite numbers of fluents and events (CONS contains a domain closure axiom while T_{LIN} does not). On the other hand, Lin's approach does not handle the extensions to our approach introduced in sections 9.6.4-9.6.7 of the previous chapter. Otherwise, as is hopefully clear from just looking at the respective definitions, Lin's approach is almost identical to ours. Though we have not proven it formally, we conjecture the two approaches to be equivalent on the set of reasoning domains for which both are defined.

Interestingly, in Lin's papers neither the choice to keep *Ho* fixed during circumscription of *Caused*, nor the choice to add the persistence axiom only after this first circumscription is motivated in terms much other than 'if you do not do it, you get counterintuitive results'. Our work thus provides an external motivation for these choices, cf. the remarks in Chapter 9, Section 9.4: the axioms (10.6)-(10.8) can be interpreted as mimicking the replacement of structural equations. This replacement can only be done properly if the right interpretations of *Do* have been obtained already for each particular interpretation of *Ho*. Note that, for the 'replacement' to happen correctly, it is *not* crucial that the intervention axioms are added after the circumscription of CONS (which is why Lin can add his version of them to T_{LIN} before circumscribing). It *is* however crucial that the persistence axiom is only added after the circumscription. Otherwise, as can be easily seen from the persistence axiom (9.36), we would infer that in each model with $\neg[\mathit{Ho}(F, t-1) \equiv \mathit{Ho}(F, t)]$ for some fluent F, we would have $\mathit{Do}(F, b, t)$ for some b. Thus any change would automatically be accompanied by an intervention, and we would select models with spurious changes.

10.4 Baral and Gelfond

We now undertake the most difficult and extensive comparison: we compare our approach to Baral and Gelfond's (BG) approach based on the *action description language* $\mathcal{L}_3$ [11]. $\mathcal{L}_3$ is an extension of Baral, Gelfond and Provetti's language $\mathcal{L}_1$ [12] which in turn is an extension of Lifschitz' language $\mathcal{A}$ [54, 135]. $\mathcal{L}_3$ extends $\mathcal{A}$ to deal with concurrent actions, actions with non-deterministic effects and observations of the actions that take place and the fluents that hold in arbitrary situations. On the other hand, it cannot at all deal with ramifications. The way it treats concurrent actions is similar to the way we treat events: BG always prefer the models of a domain in which as few actions as possible take place. Otherwise, at least on the surface, BG's approach looks quite different from ours. It is definitely not based on Pearl's ideas. Still, it turns out to be equivalent to our approach on most reasoning domains for which both approaches are defined.

We will compare BG's approach to our instantiation of causal theories in which concurrent events are allowed to happen but surprises are not, that is, we use Definition 9.24 (Chapter 9, page 230). The comparison will follow the same pattern as the comparison to MT's approach did: we first give an example of a reasoning domain where our approach gives better results than BG's does. We will then define syntactical and semantical correspondence relations and provide a theorem stating that, for the subset of possible reasoning domains for which the syntactical correspondence relation is defined, corresponding theories have corresponding models. We first have to explain the basics of BG's approach though. For more details we refer to [11] and to the much more extended [12]. Strictly speaking, the latter reference is about $\mathcal{L}_1$ and not $\mathcal{L}_3$, but $\mathcal{L}_3$ is only a minor extension of $\mathcal{L}_1$.

Review of $\mathcal{L}_3$

BG's approach consists of *domain descriptions* D written in a language $\mathcal{L}_3$. If a domain description D is *consistent*, then it has *models* M which determine what fluents hold at what time and what actions happen when. $\mathcal{L}_3$ consists of the sets of symbols $\mathcal{F}$ (corresponding to our fluents), $\mathcal{A}$ ('unit actions', corresponding to our events) and $\mathcal{S}$ ('situations', corresponding to states of the world at specific points in time). $\mathcal{S}$ contains two special situations s_0 and s_N: the *initial* and *current* situation. A *fluent literal* is a fluent possibly preceded by $\neg$. By a *generalized action* a, BG mean a disjunction of arbitrary sets of unit actions: $a = a_1|\ldots|a_m$ $(m \geq 1)$; $a_i = \{a_{i1}, \ldots, a_{in}\}$ for $1 \leq i \leq n$. Each a_i is called a *compound* action and interpreted as a set of actions which are performed concurrently and which start and stop contemporaneously. If a generalized action is performed, this means that one of the constituent compound actions is performed and it is not known which.

Domain descriptions D in $\mathcal{L}_3$ may contain several kinds of rules. First, there are *effect laws* of the form

$$a \textbf{ causes } f \textbf{ if } p_1, \ldots, p_n$$

where a is a compound action and $f, p_1, \ldots, p_n$ $(n \geq 0)$ are fluent literals. This should be read as 'f is guaranteed to be true after the execution of an action a in any state of the world in which $p_1, \ldots, p_n$ are true'.

Second, there are *fluent facts* of the form f **at** s where f is a fluent literal and s is a situation. This should be read as 'f is observed to be true in situation s'.

Third, there are *occurrence facts* which are expressions of the form

$$\alpha \textbf{ occurs\_at } s$$

where α is a sequence of generalized actions and s is a situation. This says that 'the sequence α of actions was observed to have occurred in situation s'.

Fourth, there are *precedence facts* of the form s_1 **precedes** s_2. This states that situation s_1 occurred before s_2.

The three kinds of facts introduced above are called *atomic*. A *fact* is a propositional combination of atomic facts[3].

An *interpretation* $M = (\Psi, \Sigma)$ for a domain description D contains a 'situation assignment' Σ and a 'causal interpretation' Ψ.

A *situation assignment* is a mapping from $\mathcal{S}$ to sequences of actions, such that (1) $\Sigma(s_0) = []$ ($[]$ denotes the empty sequence) and (2) for every $s_i \in \mathcal{S}$, $\Sigma(s_i)$ is a prefix of $\Sigma(s_N)$. Intuitively, Σ defines an 'action schedule': it says which actions happen in between which situations. A *causal interpretation* Ψ maps sequences of actions to 'states'. A *state* σ is an interpretation of all fluents, denoted by the set of fluents that are interpreted to be true. Hence $\Sigma(s)$ denotes the sequence of actions that have led to situation s, and $\Psi(\Sigma(s))$ denotes the set of fluents that hold in situation s.

If all the rules of a domain description D are true in an interpretation M, we call M a *model* of D. The definition of a rule 'being true in interpretation M' is relatively straightforward. The precise definition of 'truth in a model' and of modelhood can be found in Appendix 10.8.1.

A Distinguishing Example We will see that for most of the domains expressible in both formalisms, BG and our causal theories give the same results. However, in domains involving *specificity*, BG and our approach give different results, and we claim that for these domains, BG gives the less intuitive ones. The default assumption of specificity says that 'more specific information about actions overrides less specific information'. We illustrate this using a standard example [150]: suppose that if you lift a bowl of soup with either your left or your right hand, but not both, then you will spill the soup and the table will get wet. If you lift the bowl with both hands however, then you will not spill the soup. BG formalize this using the following domain description D, containing only two situations s_0 and s_N:

$$\{\mathit{Lift\_left}\} \quad \textbf{causes} \quad \mathit{Wet} \tag{10.10}$$

$$\{\mathit{Lift\_right}\} \quad \textbf{causes} \quad \mathit{Wet} \tag{10.11}$$

$$\{\mathit{Lift\_left}, \mathit{Lift\_right}\} \quad \textbf{causes} \quad \neg\,\mathit{Wet}\ \textbf{if}\ \neg\,\mathit{Wet} \tag{10.12}$$

BG now formalize the assumption of specificity as follows: if the preconditions of (10.12) hold, then rules (10.10) and (10.11) are ignored when determining the effects of *Lift_left* and *Lift_right* – see the paragraph on 'causal models' in Appendix 10.8.1 for details on how this is achieved.

Now in this simple example BG's approach works well. However, suppose your table is placed in your garden, where there is also a sprinkler very near to your table, obeying the following additional rule:

$$\mathit{Turn\_on\_sprinkler}\ \textbf{causes} \quad \mathit{Wet} \tag{10.13}$$

[3]There is one extra kind of rule in $\mathcal{L}_3$, the *hypothesis*. Hypotheses however cannot occur in domain descriptions D and will therefore be of no concern to us.

Now consider a situation in which your table is dry. Suppose in this situation you lift the soup bowl with both hands while somebody else turns on the sprinkler. While you would intuitively expect the table to get wet, according to BG's approach, this is an inconsistent domain description! More formally, suppose we have the additional facts

$$s_0 \textbf{ precedes } s_N \ \wedge \neg \mathit{Wet} \textbf{ at } s_0 \wedge \\ [\{\mathit{Turn\_on\_sprinkler}, \mathit{Lift\_left}, \mathit{Lift\_right}\}] \textbf{ occurs\_at } s_0 \tag{10.14}$$

Proposition 10.6 *There exists no model M for domain description* $D = \{(10.10) - (10.14)\}$.

The proof of this proposition can be found in Appendix 10.8.2.

One should stress that there is nothing wrong with 'specificity' in itself! From our point of view, specificity just says that, just as you assume that 'no events happen in general' when determining your set of models, you may already assume it in your specification of effect axioms; in the present example, 'specificity' amounts to the default assumption that if only *Lift_left* is known to take place, then *Lift_right* is assumed not to take place (and vice versa). We did not build in such a feature in our causal theories. While this somewhat restricts the applicability of our approach, it is unrelated to the problem with BG's approach mentioned above. That problem is caused by the inappropriate use of **causes** in (10.12): there is definitely no *sufficient cause* for not getting wet if you lift the soup bowl with two hands: no intervention that sets the value of *Wet* is performed. However, **causes** does receive a semantics in BG's approach as if it would always represent an intervention. From a Pearlian point of view, it comes as no surprise that this may lead to counterintuitive results.

Because we do not feature specificity, we would have to formalize (10.10) as

$$\mathit{Ho}(\mathit{Lift\_left}, t) \wedge \neg \mathit{Ho}(\mathit{Lift\_right}, t) \supset \mathit{Do}(\mathit{Wet}, \text{TRUE}, t+1) \tag{10.15}$$

and (10.11) accordingly. (10.12) would then simply disappear. If the sprinkler were turned on, then according to the definition of models for causal theories (Definition 9.16, page 226), the table would definitely get *Wet*.

Correspondence and Equivalence

We will see that BG's approach and ours are equivalent on all domains for which both are defined except those involving specificity. Below we define syntactic ($\sim$) and semantic ($\cong$) correspondence, and we provide a theorem stating that syntactically corresponding theorems yield semantically corresponding models. However, before we can do all this, we have to take care of three technical problems that arise. We consider each of these in turn:

- First, we have to rule out domain descriptions D such as the above which may involve specificity and which thus may be handled differently by the two approaches. These are the *ambiguous* D:

  Definition 10.7 *If a domain description D in the language $\mathcal{L}_3$ contains two effect laws*

$$a \textbf{ causes } f \textbf{ if } p_1, \ldots, p_n \quad \text{and} \quad a' \textbf{ causes } \neg f \textbf{ if } p'_1, \ldots, p'_m \tag{10.16}$$

where (1) $a \cap a' \neq \emptyset$ *or* $a = \emptyset$ *or* $a' = \emptyset$ *and (2)* $\{p_1, \ldots, p_n\} \cap \{\neg p'_1, \ldots, \neg p'_m\} = \emptyset$ *then* D *is said to be* **ambiguous**.

The exclusion of ambiguous domain descriptions will be echoed in condition (b) of Theorem 10.11 below.

- The second problem is that the definition of modelhood comes in two versions in BG's papers [12]. In one of the two versions, logically consistent initial states are *never* ruled out, even if this would lead to models in which more actions occur than is strictly necessary. In the other version only the models with a minimal number of actions are selected. Since in our approach we always prefer the models with the least number of events (i.e. the smallest, in the subset sense, interpretations of Ab_1), our approach should clearly be compared to the second version. We therefore assume, in the following, that whenever we speak of a model of BG's approach, we mean a model according to the second version of the definition of modelhood – this second version is the definition given in Appendix 10.8.1.

- The third problem is that BG use *names* for situations. In order to create corresponding theories, we need names for our time points, too. The way we define of our language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F})$ does not allow for this. We therefore have to extend Definition 9.24 of page 230 in the following straightforward way:

  Definition 10.8 *A first-order causal theory* $T_{\mathrm{C}} = \langle \textsc{eq}, \textsc{cons} \rangle$ *for a language* $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{T})$ *(where* $\mathbf{T}$ *is a finite set of constants) is a first-order causal theory with persistence, dependent fluents and concurrent events (i.e. a theory according to Definition 9.24, page 230) for the tuple* $\langle \mathbf{D}, \mathbf{E}, \mathbf{F} \rangle$ *extended with the constants in* $\mathbf{T}$*). These constants are all of sort time points.*

  The definition of models for causal theories (Definition 9.16, page 226) ensures that the constants in $\mathbf{T}$, which we will call *time names*, will always be interpreted as nonnegative integers. Whenever in the following we speak of T_{C}, we mean a theory T_{C} defined according to Definition 10.8 above.

Having taken care of these problems, we are ready to define syntactic correspondence. This is done in Definition 10.9 on page 250. As an example of how that definition works, we will give a simple domain description and a causal theory that corresponds to it. For this, let D be the domain description consisting of (10.10), (10.11),(10.13) and (10.14). This is just the soup-bowl domain we considered before but without the malfunctioning specificity axiom (10.12). According to Definition 10.9, this domain is equivalent to a causal theory T_{C} with an EQ containing the standard two intervention axioms and our persistence and no-events axiom (9.47), and a CONS containing UNA-

For fluents $F_i \in \mathbf{F}$ we use the following notation: when occurring in a formula of $\mathcal{L}_3$, F_i^{TRUE} should be read as F_i; F_i^{FALSE} should be read as $\neg F_i$. When occurring in a sentence of CONS, $Ho(F_i^{\text{TRUE}}, t)$ should be read as $Ho(F_i, t)$; $Ho(F_i^{\text{FALSE}})$ should be read as $\neg Ho(F_i, t)$.

Definition 10.9 *For any theory T_{C} for a language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{T})$ and domain description D for a language $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$, we define $T_{\text{C}} \sim D$ ('T_{C} corresponds to D') to hold iff all of the following hold:*

1. **constants** *1)* $\mathbf{D} = \emptyset$, *2)* $\mathbf{E} = \mathcal{A}$; *3)* $\mathbf{F} = \mathcal{F} = \{F_1, \ldots, F_{n_F}\}$; *4)* $\mathbf{T} = \mathcal{S}$; *and 4)* $s_0, s_N \in \mathcal{S}$.

2. **general axioms** EQ *is as required by Definition 10.8.* CONS *contains UNA- and DC- axioms for* $\mathbf{E}, \mathbf{F}$ *and* $\mathbf{B}$ *and the additional axiom* $s_0 = 0$.

3. **sentences** *Each sentence in* CONS *that is not equal to one of the sentences mentioned under item 2 above, corresponds to either an effect law or a fact in D. Each effect law and each fact in D corresponds to a sentence in* CONS. *Here 'correspondence' is defined as follows:*

4. **effect laws** *A sentence Γ_{C} corresponds to an effect law L in D iff Γ_{C} is of the form*
$$\forall t.[\, Ho(F_{j_1}^{b_1}, t) \wedge \ldots \wedge Ho(F_{j_m}^{b_m}, t) \;\wedge Ho(a^1, t) \wedge \ldots \wedge Ho(a^n, t)\,] \supset Do(F_i, b, t+1) \quad (10.17)$$
while L is of the form: $$\{a^1, \ldots, a^n\} \textbf{ causes } F_i^b \textbf{ if } F_{j_1}^{b_1}, \ldots, F_{j_m}^{b_m} \quad (10.18)$$

5. **fluent facts** *A sentence Γ_{C} corresponds to a fluent fact F in D iff Γ_{C} is of the form '$Ho(f, s)$' while F is of the form 'f* **at** *s'.*

6. **occurrence facts** *A sentence Γ_{C} corresponds to an occurrence fact O in D iff O is of the form '$[a_1, \ldots, a_n]$* **occurs_at** *s', where $n > 0$, $a_i = a_{i1} \mid \ldots \mid a_{im_i}$, $a_{ij} = \{a_{ij}^1, \ldots, a_{ij}^{k(i,j)}\}$, $a_{ij}^{k(i,j)} \in \mathcal{A}$, while Γ_{C} is of the form:*
$$\begin{array}{r}
[\, \mathbf{H}(a_{11}, s) \vee \;\ldots\; \vee \mathbf{H}(a_{1m_1}, s)\,] \;\wedge \\
[\, \mathbf{H}(a_{21}, s+1) \vee \;\ldots\; \vee \mathbf{H}(a_{2m_2}, s+1)\,] \;\wedge \\
\vdots \\
\wedge\, [\, \mathbf{H}(a_{n1}, s+n-1) \vee \;\ldots\; \vee \mathbf{H}(a_{nm_n}, s+n-1)\,]
\end{array} \quad (10.19)$$
where $\mathbf{H}(a_{ij}, s)$ is short for $Ho(a_{ij}^1, s) \wedge \ldots \wedge Ho(a_{ij}^{k(i,j)}, s)$.

7. **precedence facts** *A sentence Γ_{C} corresponds to a precedence fact P in D iff Γ_{C} is of the form '$s_1 < s_2$' while P is of the form 's_1* **precedes** *s_2'.*

8. **non-atomic facts** *A sentence Γ_{C} corresponds to a fact F iff Γ_{C} is a propositional combination of constituents that each correspond to an atomic fact and F is the same propositional combination of the corresponding atomic facts.*

Consider an interpretation $M = (\Psi, \Sigma)$ of a domain description D in $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$. $\Sigma(s_N)$ can be written as a (possibly empty) sequence of compound actions (here $last \in \mathbf{N}_0$):

$$\Sigma(s_N) = [a_0, a_1, \dots, a_{last}]$$

Similarly, we can write for all $s \in \mathcal{S}$: $\Sigma(s) = [a_0, \dots, a_t]$ with $t \leq last$. For $t > last$ we define a_t to be the empty set. We define $[a_{-1}]$ to be equal to the empty sequence []. Using this convention, for $0 \leq t \leq last + 1$, we define σ_t as an abbreviation for $\Psi([a_0, \dots, a_{t-1}])$. For $t > last + 1$, σ_t is defined to be equal to σ_{last+1}.

Definition 10.10 *For any T_{C} for a language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{T})$ and any non-ambiguous domain description D for a language $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$ such that $T_{\text{C}} \sim D$, let $M = (\Psi, \Sigma)$ be any interpretation of $(\mathcal{F}, \mathcal{A}, \mathcal{S})$ and $\mathcal{M}$ be any interpretation of our language. We say that M corresponds to $\mathcal{M}$ (written as '$M \cong \mathcal{M}$') iff (a) $\mathcal{M}$ interprets all time points as integers and '$+$' and '$<$' accordingly; and (b) for all $f \in \mathcal{F}, a \in \mathcal{A}$ and $s \in \mathcal{S}$ and all $t \in \mathbf{N}_0$:*

1. $a \in a_t \Leftrightarrow \mathcal{M} \models Ho(a, t)$
2. $f \in \sigma_t \Leftrightarrow \mathcal{M} \models Ho(f, t)$
3. *$\Sigma(s)$ contains exactly t elements* $\Leftrightarrow \mathcal{M} \models s = t$

and DC-axioms and on top of that the axioms:

$$\begin{gathered} s_0 = 0 \\ Ho(Lift\_left, t) \supset Do(Wet, \text{TRUE}, t+1) \\ Ho(Lift\_right, t) \supset Do(Wet, \text{TRUE}, t+1) \\ Ho(Turn\_on\_sprinkler, t) \supset Do(Wet, \text{TRUE}, t+1) \\ (s_0 < s_N) \wedge \neg Ho(Wet, s_0) \wedge \\ [\ Ho(Turn\_on\_sprinkler, s_0) \wedge Ho(Lift\_left, s_0) \wedge Ho(Lift\_right, s_0)\ ] \end{gathered} \tag{10.20}$$

Here the first axiom is introduced by item 2 of the definition; the second, third and fourth axioms are introduced by items 3 and 4 of the definition and the fifth axiom is introduced by item 8 of the definition. Item 6 of definition 10.9 deserves special attention: it translates any rule in D of the form '$[a_1, \dots, a_n]$ **occurs_at** s' into an axiom in CONS which expresses that a_1 should hold at time point s, a_2 at time point $s+1$ etc. In other words, if, in BG's formalism, several actions $[a_1, \dots, a_n]$ take place sequentially in a situation s, we interpret this as saying, in our formalism, that the first of them happens at the point in time t corresponding to s, the second to the point in time directly thereafter etc. Any situation s' such that s precedes s' in BG's formalism will therefore be mapped to a point in time t' that is sufficiently larger than t so as to allow for all the actions $[a_1, \dots, a_n]$ to happen in between t and t'.

The semantical correspondence relation '$\cong$' is introduced in Definition 10.10 on page 251. Just above that definition, we introduce the notation a_t to stand for the $(t+1)$-th compound action taking place in an interpretation M. Similarly, σ_t stands for the state of the world (i.e. the set of all fluents that hold) just before the $(t+1)$-th compound action takes place. In the definition itself, it is checked whether for any set of actions denoted by a_t, the same actions take place in $\mathcal{M}$ *at time* t; σ_t is treated similarly.

Having discussed both syntactic and semantical correspondence, we are now ready to state our theorem. The proof can be found in Appendix 10.8.3.

Theorem 10.11 *For any theory T_{C} for a language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{T})$ and any domain description D for a language $\mathcal{L}_3$ of Baral and Gelfond's such that a) $T_{\mathrm{C}} \sim D$ and b) D is not* ambiguous*, we have:*

$$M = (\Psi, \Sigma) \text{ is a model for } D \Rightarrow$$
$$\text{there exists an } \mathcal{M} \text{ with } M \cong \mathcal{M} \text{ such that } \mathcal{M} \text{ is a preferred model of } T_{\mathrm{C}}$$

and we further have

$$\mathcal{M} \text{ is a preferred model of } T_{\mathrm{C}} \Rightarrow \text{there exists a model } M \text{ for } D \text{ such that } M \cong \mathcal{M}$$

10.5 A Brief Note on Other Approaches based on Pearl's ideas

There have been several earlier papers on common-sense reasoning about action that were (partially) based on Pearl's ideas; we mention the papers of Darwiche and Pearl [33, 34], Boutilier and Goldszmidt [19] and Geffner [52]. However, in the works of Darwiche, Pearl, Boutilier and Goldszmidt actions are not directly treated as interventions, but rather compiled into nodes in a causal graph (in Darwiche and Pearl's case) or a Bayesian Network (in Boutilier and Goldszmidt's case). The resulting theories are quite different from ours in that they lack the *Do*-operator, which is fundamental to our solution of the ramification problem. Only the approach of Geffner is able to handle comparable instances of the ramification problem. We plan a more in-depth comparison to Geffner's approach in future work.

Earlier, the present author has introduced the two model selection criteria $\mathbf{S}_0$ and $\mathbf{I}_0$ [62, 63] which were claimed to be based on Pearl's causal graphs. However, the connection was not investigated in detail. The work reported here can be seen as an extension of this original work, but now with the connection worked out in full detail (see the appendix to Chapter 9). Putting the theory in a form that makes as explicit as possible the connection to Pearl's work has caused the causal theories discussed in this thesis to look – superficially – quite different from $\mathbf{S}_0$ and $\mathbf{I}_0$.

10.6 Conclusion

In this chapter we have compared our causal theories to existing (causal) approaches to NMTR. We have seen that our theories permit the same inferences as these existing approaches for large classes of reasoning domains, while showing (in the case of McCain and Turner and Baral, Gelfond and Provetti) that our approach sometimes gives better results on domains falling outside these classes. In this way, the two goals we set ourselves at the beginning of the chapter have been achieved: we provided evidence that our approach 'truely works well', and we gained insight in how the treatment of causality in existing approaches is related to Pearl's 'sufficient cause principle': both the counterexamples we referred to above (given by proposition 10.1 and 10.6, respectively) can be interpreted as stemming from not following this principle. In contrast, Lin's persistence and causation axioms (axioms (10.6)-(10.8)) are almost equivalent to our persistence and intervention axioms, which (as explained in the previous chapter, Section 9.6.1) are directly based on the sufficient cause principle.

Future Work In future work, we would like to make a more extensive comparison between our approach and that of Thielscher [151]. Apart from Pearl, Thielscher is one of the few authors in the field who tries to make completely clear what he means by a causal law, i.e. he attempts to give a semantics to statements of the form 'A causes B' in non-causal terms. Interestingly, Thielscher's is also the only approach we are aware of that can correctly handle reasoning domains for which our approach fails ([151], Example 18). On the other hand, our use of Do is much less restrictive than Thielscher's use of `causes`. We plan to study the exact differences between Thielscher's and our approach in the near future.

Another approach that deserves further attention is the one based on Sandewall's and Doherty's *occlusion* concept [135, 134, 66]. The underlying idea behind occlusion is similar to Pearl's ideas about interventions: in Sandewall, Gustafsson and Doherty's work, if a fluent is *affected* by some action, then it becomes *occluded* for the duration of the action, which means that its value becomes independent of the value it had directly before the action started. Compare this to the sufficient cause principle: if an action takes place that *sets* the value of some fluent, then the fluent becomes independent of the values of any variables in the domain which normally influence it. We see that the two concepts are closer than their names suggest and we think that they should be compared in detail.

This brings us once again to what may be the most interesting aspect of our approach: it partially bridges the conceptual gaps that exist between different paradigms in the field of reasoning about action. First, within the logical (nonmonotonic reasoning) paradigm, it relates the causal approaches to the non-causal approaches. Second and more generally, it establishes a connection between formalizations of causal knowledge that have been developed within this logical paradigm (such as Lin's and McCain and Turner's) and those that have their roots in the probabilistic/ Bayesian network tradition (i.e. Pearl's causal networks).

10.7 Appendix: McCain & Turner vs. 2-point causal theories

10.7.1 Formal Definition of MT's Next-State Function

The definitions in this section have all been copied from [105, 104]. MT start with a propositional language which includes the 0-ary logical connectives TRUE and FALSE. TRUE and ¬FALSE are tautologies in which no atoms occur. Each interpretation is identified with the set of literals true in it. A *causal law* is defined to be an expression of the from $\phi \Rightarrow \psi$ where ϕ and ψ are formulas of the propositional language. A set of causal laws is called a *causal theory.* Now for every causal theory D and interpretation I, we let

$$D^I = \{\psi : \text{for some } \phi,\ \phi \Rightarrow \psi \in D \text{ and } I \models \phi\}\ .$$

That is, D^I is the set of consequents of all causal laws in D whose antecedents are true in I.

Definition 10.12 *Let D be a causal theory and I be an interpretation. We say that I is* causally explained *according to D if I is the unique model of D^I.*

We now define $\Pi_1(S, E, C)$. It is a function on *initial states* S, *explicit effects* E and *background knowledge* C. A state is just an interpretation; an explicit effect is a set of formulas; the background knowledge is a set of causal laws.

Definition 10.13 *For any interpretation S, set of propositional formulas E and set of causal laws C, $\Pi_1(S, E, C)$ is defined to be the set consisting of all states that are causally explained according to the causal theory*

$$\{L \Rightarrow L : L \in S\} \cup \{\text{TRUE} \Rightarrow \phi : \phi \in E\} \cup C$$

We will repeatedly use the following lemma which gives a precise characterization of $\Pi_1(S, E, C)$. It was first presented by McCain & Turner [105] but without proof. The proof however is not difficult; we provide one elsewhere.

Lemma 10.14 (McCain & Turner) *For any interpretation S, set of propositional formulas E and set of causal laws C, a state S' belongs to $\Pi_1(S, E, C)$ if and only if*

- $S' \models E \cup C^{S'}$
- $S' \setminus S \subseteq \{\phi : \phi \text{ is a literal and } (S \cap S') \cup E \cup C^{S'} \models \phi\}$

The proof can be found in [64].

10.7.2 Proof of Proposition 10.1

Take S, E and C as in the statement of the proposition. Let S' be any member of $\Pi_1(S, E, C)$. By Lemma 10.14 in Appendix 10.7.1 it follows that $S' \models E$ and hence $X \in S'$. This means $C^{S'}$ must contain $Y \vee Z$. Applying Lemma 10.14 again, we find $S' \models Y \vee Z$. So there are three possibilities left for S': $S' \in \{S_1, S_2, S_3\}$ where $S_1 = \{X, Y, Z\}$, $S_2 = \{X, Y, \neg Z\}$, $S_1 = \{X, \neg Y, Z\}$. Now the second part of Lemma 10.14 does not hold for $S' = S_1$ while it does hold for $S' = S_2$ and $S' = S_3$. The first part of the lemma also holds for $S' = S_2$ and $S' = S_3$, so $\Pi_1(S, E, C) = \{S_2, S_3\}$.

10.7.3 Proof of Theorem 10.4

Our proof makes repeated use of Lemma 10.14 of Appendix 10.7.1. We first prove the first part of Theorem 10.4, which is restated below. We use the notation introduced in Definition 10.2, i.e. we will sometimes write F_i^{TRUE} to denote F_i and F_i^{FALSE} to denote $\neg F_i$.

Theorem 10.4 (part I) *For any theory* T_{C} *and any domain description* $T_{\mathrm{MT}} = (S, E, C)$ *such that* $T_{\mathrm{C}} \sim T_{\mathrm{MT}}$ *, we have:*

$$S' \in \Pi_1(S, E, C) \Rightarrow \textit{ there exists an } \mathcal{M} \textit{ with } \mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}} \textit{ and } \mathcal{M} \cong (S, S')$$

Proof: For any given $S, E, C; S' \in \Pi_1(S, E, C)$ and T_{C} we will first construct a model $\mathcal{M}$ with $\mathcal{M} \cong (S, S')$. We then show $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$.

Construction of $\mathcal{M}$ We construct a causal model $\mathcal{M}$ according to the following definition, in which we identify sets containing conjunctions of literals with the sets containing just the constituting literals:

1. The initial state of $\mathcal{M}$ is S; the final state of $\mathcal{M}$ is S'. In other words: $F_i \in S \Leftrightarrow \mathcal{M} \models F_i(0)$; $F_i \in S' \Leftrightarrow \mathcal{M} \models F_i(1)$.
2. $F_i^b \in E \cup C^{S'} \Leftrightarrow \mathcal{M} \models Do(F_i(1), b)$.

It is clear that a model $\mathcal{M}$ satisfying all the conditions above can indeed be constructed. It follows immediately from the definition of '$\cong$' (Definition 10.3) and item 1 in the construction of $\mathcal{M}$ that $\mathcal{M} \cong (S, S')$. We now prove that $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ in three stages. In stage 1, we show $\mathcal{M} \models$ CONS. We use this to show in stage 2 that $\mathcal{M}$ is a minimal model for CONS of $\mathcal{A}(\mathbf{V})$ within context $\mathcal{M}\rfloor_{\mathbf{V} \cup \mathbf{U}}$. This shows that $\mathcal{M}$ satisfies condition (1) of Definition 9.7. In stage 3, we show that $\mathcal{M}$ also satisfies condition (2) of that definition.

Stage 1 We have to show that $\mathcal{M} \models \phi$ for all axioms ϕ contained in CONS. From Definition 10.2, item 2, it follows that it is sufficient to show that $\mathcal{M} \models \phi$ for all sentences in CONS that correspond to either S or to the sentences in E and C. By items 3 and 4 of the same definition and the construction of $\mathcal{M}$ above, it follows immediately that indeed $\mathcal{M} \models \phi$ for the sentences ϕ that correspond to S and E.

Concerning C, let Γ be any sentence in CONS of the form (10.4). We will show $\mathcal{M} \models \Gamma$. Let us write $\Gamma \equiv \Phi \supset \Psi$ with Ψ of the form $Do(F_{i_1}(1), b_1) \wedge \ldots \wedge Do(F_{i_m}(1), b_m)$. If $\mathcal{M} \not\models \Phi$ we are done, so let us suppose $\mathcal{M} \models \Phi$. By construction of $\mathcal{M}$, we have $S' \models \Phi'$ where Φ' is obtained from Φ by replacing all occurrences of $F_i(1)$ in Φ by F_i. By item 5 of Definition 10.2, C must contain a rule of form $\Phi' \Rightarrow F_{i_1}^{b_1} \wedge \ldots \wedge F_{i_m}^{b_m}$ and, since $S' \models \Phi'$, we conclude that $F_{i_j}^{b_j} \in C^{S'}$ for all $1 \leq j \leq m$. Hence by item 2 of the construction of $\mathcal{M}$, $\mathcal{M} \models \Psi$ and hence $\mathcal{M} \models \Gamma$.

Stage 2 We have to show that $\mathcal{M}$ is a minimal element for CONS of $\mathcal{A}(\mathbf{V})$ within the set

$$\mathbf{M}(\mathcal{M}) = \{\mathcal{M}' \mid \mathcal{M}' \text{ and } \mathcal{M} \text{ have the same interpretation of } \mathbf{V} \cup \mathbf{U}\}$$

For this, let $\mathcal{M}' \in \mathbf{M}(\mathcal{M})$ be a model for CONS. Suppose, by way of contradiction, that $\{\phi \in \mathcal{A}(\mathbf{V}) \mid \mathcal{M}' \models \phi\}$ is a proper subset of $\{\phi \in \mathcal{A}(\mathbf{V}) \mid \mathcal{M} \models \phi\}$. Then there is a pair (F_i, b) such that $\mathcal{M}' \not\models Do(F_i, b)$ while $\mathcal{M} \models Do(F_i, b)$. $\mathcal{M}$ is constructed such that $F_i^b \in E \cup C^{S'}$. Now if $F_i^b \in E$, then, by Definition 10.2, item 4, $\mathcal{M}' \not\models$ CONS and we arrive at a contradiction. So suppose $F_i^b \in C^{S'}$. It follows from the definition of $C^{S'}$ that $S' \models \Phi'$ for a Φ' such that C contains some sentence of form $\Phi' \Rightarrow \Psi'$ and $F_i^b \in \Psi'$. Hence by Definition 10.2, item 5, CONS contains the sentence $\Phi \supset \Psi$ where Φ, Ψ stand to Φ', Ψ' as required in that item of the definition. Since $S' \models \Phi'$, by the construction of $\mathcal{M}$, we have $\mathcal{M} \models \Phi$. Since $\mathcal{M}'$ and $\mathcal{M}$ share the same interpretations of the variables in $\mathbf{V}$, we also have $\mathcal{M}' \models \Phi$. We assumed $\mathcal{M}' \models$ CONS so also $\mathcal{M}' \models \Psi$ and in particular $\mathcal{M}' \models Do(F_i, b)$. But we assumed the contrary so we have arrived at a contradiction.

Stage 3 We have to show that $\mathcal{M} \models$ EQ$'$ where EQ$'$ is the updated set of structural equations referred to in condition (2) of Definition 9.7. By construction of $\mathcal{M}$, there are two cases for each F_i: either EQ$'$ contains $F_i(1) \equiv b$ for some b or EQ$'$ contains $F_i(1) \equiv F_i(0)$.

The first case happens if $\mathcal{M} \models Do(F_i(1), b)$. By construction of $\mathcal{M}$, it follows that in this case $F_i^b \in E \cup C^{S'}$. By Lemma 10.14 and the fact that $S' \in \Pi_1(S, E, C)$ we have $S' \models E \cup C^{S'}$ and thus $S' \models F_i^b$. Then by construction of $\mathcal{M}$, indeed $\mathcal{M} \models F_i(1) \equiv b$ and we are done.

The second case happens if $\mathcal{M} \not\models Do(F_i, b)$ for any $b \in \mathbf{B}$. It follows from the construction of $\mathcal{M}$ that in this case there is no b such that $F_i^b \in E \cup C^{S'}$. Suppose, by way of contradiction, that $\mathcal{M} \not\models F_i(1) \equiv F_i(0)$. Then (by construction of $\mathcal{M}$), $F_i^b \in S' \setminus S$ for some b. By Lemma 10.14, $(S \cap S') \cup E \cup C^{S'} \models F_i^b$. Clearly, $F_i^b \notin S \cap S'$. We have already shown that F_i^b cannot be in $E \cup C^{S'}$ either. Since both $S \cap S'$ and $E \cup C^{S'}$ only contain conjunctions of literals, it follows that $(S \cap S') \cup E \cup C^{S'} \not\models F_i^b$ and we have reached a contradiction. $\Box$

We now restate and prove the second part of Theorem 10.4.

Theorem 10.4 (part II) *If we have a theory* T_{C} *and a* $T_{\mathrm{MT}} = (S, E, C)$ *such that* $T_{\mathrm{C}} \sim T_{\mathrm{MT}}$, *then*

$$\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}} \;\Rightarrow\; \Pi_1(S, E, C) \textit{ contains an } S' \textit{ with } \mathcal{M} \cong (S, S')$$

Proof: We will define S' in terms of a given model $\mathcal{M}$ with $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$, as follows: for all F_i: $F_i \in S' \Leftrightarrow \mathcal{M} \models F_i(1)$. We will show that this S' is contained in $\Pi_1(S, E, C)$ and that $\mathcal{M} \cong (S, S')$.

Since $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ and thus $\mathcal{M} \models$ CONS we know by items 2 and 3 of Definition 10.2 that for all F_i: $F_i \in S \Leftrightarrow \mathcal{M} \models F_i(0)$. From this and the definition of S' above it follows $\mathcal{M} \cong (S, S')$. It remains to be shown that $S' \in \Pi_1(S, E, C)$. By Lemma 10.14 it is sufficient to prove claims 1 and 2 below.

Claim 1 $S' \models E \cup C^{S'}$.
To prove this, we show that for all F_i^b with $F_i^b \in E$ and for all F_i^b with $F_i^b \in C^{S'}$ we have $S' \models F_i^b$. In the first case ($F_i^b \in E$), as $\mathcal{M} \models$ CONS and by Definition 10.2, item 4, we have $\mathcal{M} \models Do(F_i(1), b)$. Since $\mathcal{M} \models$ EQ$'$ this means $\mathcal{M} \models F_i^b(1)$. From the definition of S', we now have $S' \models F_i^b$ and we are done. In the second case, i.e. $F_i^b \in C^{S'}$, C must contain a law of form $\Phi' \Rightarrow \Psi'$ such that $S' \models \Phi'$ and one of the conjuncts in Ψ' is F_i^b. By definition of S' and Definition 10.2, item 5, we also have $\mathcal{M} \models \Phi$ where $\Phi \supset \Psi$ is the sentence in CONS that corresponds to $\Phi' \supset \Psi'$. As $\mathcal{M} \models$ CONS, we have $\mathcal{M} \models \Psi$ and in particular $\mathcal{M} \models Do(F_i(1), b)$. As $\mathcal{M} \models$ EQ$'$, it follows $\mathcal{M} \models F_i^b(1)$ and hence, by definition of S', $S' \models F_i^b$.

Claim 2 $S' \setminus S \subseteq \{\phi : (S \cap S') \cup E \cup C^{S'} \models \phi\}$.
To prove this, we show that for any F_i^b, if $F_i^b \in S' \setminus S$, then we must also have $F_i^b \in (S \cap S') \cup E \cup C^{S'}$. So suppose $F_i^b \in S' \setminus S$. By definition of S', we have for such an F_i^b that $\mathcal{M} \models F_i^b(1) \wedge \neg F_i^b(0)$. Since $\mathcal{M} \models$ EQ$'$, the structural equation $F_i(1) \equiv F_i(0)$ cannot be in EQ$'$ and thus there must have been an intervention: we must have $\mathcal{M} \models Do(F_i(1), b)$. Now as $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$, there can be no other $\mathcal{M}' \models$ CONS with the same interpretations of $\mathbf{V}$ and $\mathbf{U}$ but with a smaller (in the subset sense) interpretation of the '*Do*' variables. This means that CONS must contain an axiom somehow mentioning $Do(F_i(1), b)$. From Definition 10.2 we see there are two possibilities: The first possibility is that CONS contains the axiom $Do(F_i(1), b)$, i.e. an axiom of the form given in item 4 of Definition 10.2. In this case, it follows $F_i^b \in E$ so $F_i^b \in (S \cap S') \cup E \cup C^{S'}$ and we are done.

The second possibility is that CONS contains an axiom of form (10.4) while $\mathcal{M} \models \Phi$ where Φ is as in (10.4) and one of the conjuncts on the right-hand side of the axiom is $Do(F_i(1), b)$. In that case, C contains the corresponding rule $\Phi' \Rightarrow \Psi'$ with $F_i^b \in \Psi'$ and, by definition of S', $\Phi' \in S'$. Therefore, $F_i^b \in C^{S'}$ and hence $F_i^b \in (S \cap S') \cup E \cup C^{S'}$ and we are done. □

10.8 Appendix: $\mathcal{L}_3$ and First-Order Causal Theories

10.8.1 Formal Definition of Modelhood

The definitions in this section have all been copied from [11]. In order to formally define the notion of 'model', we first need to introduce a menagerie of other concepts. For detailed explanation of these concepts and clarifying examples, we refer to [11]. Roughly, the definition of 'model' for BG's domain descriptions D (Definition 10.19) makes use of two other, more basic notions: the notion of a *causal model* and the notion of 'a fact being true in an interpretation M'. Both concepts will be introduced below.

In all that follows, $\circ$ stands for *concatenation*: For a given sequence of objects α and an object a, the 'concatenation of α and a', written as $\alpha \circ a$, stands for the sequence of α followed by a.

Causal Models A *state* is a set of fluent names. A *causal interpretation* is a partial function Ψ from sequences of actions to states such that (1) the empty sequence [] belongs to the domain of Ψ and (2) Ψ is prefix-closed, i.e. for any sequence of actions α and any action a, if $\alpha \circ a$ is in the domain of Ψ then so is α.

Given a fluent f and a state σ, we say that f is *true* in σ if $f \in \sigma$; $\neg f$ is *true* in σ if $f \notin \sigma$. The truth of a propositional combination of fluents with respect to σ is defined as usual.

A fluent f is an *immediate effect* of (executing) a in σ if there is an effect law 'a **causes** f **if** $p_1, \ldots, p_n$' in D whose preconditions $p_1, \ldots p_n$ hold in σ.

A fluent f is an *inherited effect* of (executing) a in σ if there is $b \subset a$ such that (1) f is an immediate effect of b in σ; (2) there is no action c such that $b \subset c \subseteq a$ and $\neg f$ is an immediate effect of c in σ.

f is an *effect* of (executing) a in σ if either f is an immediate effect of a in σ or f is an inherited effect of a in σ.

Now let

$$E_a^+(\sigma) = \{f : f \text{ is an effect of } a \text{ in } \sigma\} \;;\; E_a^-(\sigma) = \{f : \neg f \text{ is an effect of } a \text{ in } \sigma\}$$

and $Res(a, \sigma) = \sigma \cup E_a^+(\sigma) \setminus E_a^-(\sigma)$.

Definition 10.15 *A causal interpretation Ψ* satisfies *effect laws of D if for any sequence $\alpha \circ a$ from the language of D:*

$$\Psi(\alpha \circ a) = Res(a, \Psi(a)) \text{ if } E_a^+(\Psi(\alpha)) \cap E_a^-(\Psi(\alpha)) = \emptyset$$

and undefined otherwise.
We say that Ψ is a causal model *of D if it satisfies all the effect laws of D.*

Truth of Facts We already defined a *situation assignment* on page 246. To introduce the notion of 'truth of facts in an interpretation', we first need to make our definition of 'interpretation' (also page 246) more precise:

Definition 10.16 *For any domain description D and any causal interpretation Ψ that is a causal model of D, an* interpretation *M of $\mathcal{L}_3$ is a pair (Ψ, Σ) where Ψ is a causal model of D, Σ is a situation assignment of $\mathcal{S}$ and $\Sigma(s_N)$ belongs to the domain of Ψ.*

Definition 10.17 *For any interpretation $M = (\Psi, \Sigma)$,*

1. *(f **at** s) is* true *in M (or satisfied by M) if f is true in $\Psi(\Sigma(s))$.*
2. *Atomic fact ($[a_1, \ldots, a_n]$ **occurs_at** s) is true in M if there is a sequence $\beta = [b_1, \ldots, b_n]$ of actions such that (1) The sequence $\Sigma(s) \circ \beta$ is a prefix of the actual path of M, and (2) for any $i, 1 \leq i \leq n$, $a_i^* \subseteq b_i$. Here a_i^* is an instance[4] of a_i.*
3. *(s_1 **precedes** s_2) is true in M if $\Sigma(s_1)$ is a proper prefix of $\Sigma(s_2)$.*
4. *Truth of non-atomic facts in M is defined as usual.*

A set of facts is true in interpretation M if all its members are true in M.

[4]For any compound action a^* and any generalized action a, we say a^* is an instance of a if $a = a_1 | \ldots | a_m$ and for some $1 \leq i \leq m$, $a_i = a^*$.

Models of domain descriptions D First, we need to introduce one more concept:

Definition 10.18 *Let* $\alpha = [a_1, \ldots, a_n]$ *and* $\beta = [b_1, \ldots, b_m]$ *be sequences of actions. We say that* $\alpha \leq \beta$ *if there exists a subsequence*[5] $[b_{i_1}, \ldots, b_{i_n}]$ *of* β *such that for every* $a_k \in \alpha, a_k \subseteq b_{i_k}$.

We can now finally define what it means to be a model:

Definition 10.19 [11] *An interpretation* $M = (\Psi, \Sigma)$ *is called a* model *of a domain description* D *in* $\mathcal{L}_3$ *if the following conditions are satisfied:*

C1. Ψ *is a causal model of* D

C2. *facts of* D *are* true *in* M

C3. *there is no other interpretation* $N = (\Psi', \Sigma')$ *such that* N *satisfies the conditions* **C1.** *and* **C2.** *and* $\Sigma'(s_N) \leq \Sigma(s_N)$ *and* $\Sigma'(s_N) \neq \Sigma(s_N)$.

The definition given above is slightly different from the one given in [11, 12]. This is because we opt for the version of BG's approach in which the models with the least number of actions are always preferred (see page 249). At page 10 of [12], it is claimed that in this case, the definition should be with **C1** and **C2** as above, but **C3** replaced by

C3'. there is no other interpretation $N = (\Psi', \Sigma')$ such that N satisfies the conditions **C1.** and **C2.** and $\Sigma'(s_N) \leq \Sigma(s_N)$.

However, we must assume that BG have made a mistake here: it is easy to show that if one used **C3'** instead of **C3**, then there will be no models for *any* domain description D whatsoever. We therefore opt for the unproblematic version of **C3** given above.

10.8.2 Proof of Proposition 10.6

We abbreviate all fluent and action names in the obvious way.

We have $\mathcal{F} = \{W\}, \mathcal{A} = \{LL, LR, T\}, \mathcal{S} = \{s_0, s_N\}$. Suppose, by means of contradiction, that a model $M = (\Psi, \Sigma)$ of D exists. We then have by Definition 10.19, **C2** and (10.14) that $\neg W$ **at** s_0 must hold in M. By Definition 10.17, item 2 it follows that $\Psi([]) = \emptyset$ (i.e. $\neg W$ is true in $\Psi([])$). Also, Ψ must be a causal model of D. It is straightforward to verify that for $a = \{T, LL, LR\}$ we have $E_a^+(\Psi(\alpha)) \cap E_a^-(\Psi(\alpha)) = \{W\}$. Definition 10.15 now tells us that $\Psi([\{T, LL, LR\}])$ is undefined. Since by Definition 10.17, item 2, we have for any model $M = (\Psi, \Sigma)$ of D that $[\{T, LL, LR\}]$ is a prefix of $\Sigma(s_N)$, it follows that $\Psi(\Sigma(s_N))$ is undefined too. But then, by Definition 10.16, M is not an interpretation of D. Therefore it is not a model of D either and we have reached the desired contradiction. □

[5]Given a sequence $X = x_1, \ldots, x_m$, another sequence $Z = z_1, \ldots, z_n$ is a subsequence of X if there exists a strictly increasing sequence $i_1, \ldots, i_n$ of indices of X such that for all $j = 1, 2, \ldots, n$, we have $x_{i_j} = z_j$.

10.8.3 Proof of Theorem 10.11

The proof of Theorem 10.11 is necessarily quite involved, the reason being that both our definition of 'preferred models' and BG's definition of models contain a self-reference: a model $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ is preferred if there is *no other* model $\mathcal{M}' \models_{\mathbf{c}} T_{\mathrm{C}}$ with a smaller interpretation of Ab_1. Similarly, M is a model of D if there is *no other* model satisfying conditions **C1** and **C2** of Definition 10.19 in which fewer actions happen. For this reason, we will first show that for corresponding M and $\mathcal{M}$, we have that the same fluents and events hold at the same time (Lemma 10.20). We use this result to show that for corresponding M and $\mathcal{M}$, we have $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ iff conditions **C1** and **C2** hold for M (this is essentially what happens in lemmas 10.22, 10.23 and 10.24). Only then will we be in a position to prove the theorem itself. We start by proving the four lemmas mentioned.

Correspondence of Facts

In the following, whenever we mention T_{C}, we mean a causal theory T_{C} defined for a language $\mathcal{L}(\mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{T})$ according to Definition 10.8.

Lemma 10.20 *For any T_{C} for an instance of our language and any non-ambiguous domain description D for a language $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$ such that $T_{\mathrm{C}} \sim D$, let $M = (\Psi, \Sigma)$ be any interpretation of $(\mathcal{F}, \mathcal{A}, \mathcal{S})$ and $\mathcal{M}$ be any interpretation of our language such that $M \cong \mathcal{M}$. Let F be any fact in D and ϕ be any axiom in* CONS *such that F and ϕ are corresponding facts (where correspondence is defined as in Definition 10.9, items 5-8). We have:*

$$F \text{ is true in } M \quad \Leftrightarrow \mathcal{M} \models \phi \tag{10.21}$$

Proof of Lemma 10.20: The fact F mentioned in the lemma can either be atomic or non-atomic. We first consider atomic facts. They can be of three kinds: fluent facts, occurrence facts and precedence facts. We only give the proof for fluent facts. The proofs for the other two cases follow the same scheme as the one for fluent facts so we omit them.

So suppose that F is a fluent fact 'f **at** s'. We show that if F is true in M, then the corresponding fluent fact ϕ holds in $\mathcal{M}$. Notice ϕ equals '$Ho(f, s)$'.

So suppose F is true in M. It follows that f is true in $\Psi(\Sigma(s))$, so $f \in \sigma_t$ for the t that is equal to the length of the sequence $\Sigma(s)$. Now since $M \cong \mathcal{M}$, it follows from Definition 10.10 that $\mathcal{M} \models Ho(f, t)$ and $\mathcal{M} \models s = t$, so $\mathcal{M} \models Ho(f, s)$, so $\mathcal{M} \models \phi$.

If ϕ holds in $\mathcal{M}$, then it follows that F is true in D by an analogous line of reasoning, but in the other direction. We omit the details.

We now consider non-atomic facts. The truth of non-atomic facts F in an interpretation M is defined in terms of its constituent atomic facts in exactly the same manner as the truth of non-atomic sentences for a model $\mathcal{M}$ is defined in terms of its constituent atoms. Since we have already proven that the lemma holds for all atomic facts, it follows that it still holds for non-atomic facts. $\square$

Causal Consistency and its Lemma's

Definition 10.21 (Causally Consistent) *For any T_{C} for an instance of our language and any non-ambiguous domain description D for a language $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$ such that $T_{\mathrm{C}} \sim D$, let $M = (\Psi, \Sigma)$ be any interpretation of $(\mathcal{F}, \mathcal{A}, \mathcal{S})$ and $\mathcal{M}$ be any interpretation of our language. We say that M and $\mathcal{M}$ are* causally consistent *iff*

1. $M \cong \mathcal{M}$.
2. $\mathcal{M} \models \forall e, b, t.\ \neg Do(e, b, t)$
3. *For all F_i^b: a) $\mathcal{M} \models \neg Do(F_i, b, 0)$. b) for all $t \in \{1, 2, \ldots\}$: $\mathcal{M} \models Do(F_i, b, t)$ iff* CONS *contains an axiom ϕ of form (10.17) such that (1) the F_i and b mentioned in the right-hand side of ϕ are equal to the F_i and b mentioned here and (2) the left-hand side of ϕ holds in $\mathcal{M}$.*
4. $\mathcal{M} \models \forall e, t\ .\ Ab_1(e, t) \equiv Ho(e, t)$.
5. $|\mathcal{M}|_b = \mathbf{B}; |\mathcal{M}|_e = \mathbf{E}; |\mathcal{M}|_f = \mathbf{F}$*; $\mathcal{M}$ interprets all constants in* $\mathbf{B}$, $\mathbf{E}$ *and* $\mathbf{F}$ *as themselves.*
6. *Ψ is a causal model of D.*

Lemma 10.22 *For any T_{C} for an instance of our language and any non-ambiguous domain description D for a language $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$ such that $T_{\mathrm{C}} \sim D$, let $M = (\Psi, \Sigma)$ be any interpretation of $(\mathcal{F}, \mathcal{A}, \mathcal{S})$ and $\mathcal{M}$ be any interpretation of our language such that $M \cong \mathcal{M}$. We have:*

$$\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}} \Rightarrow \textit{All facts of } D \textit{ are true in } M \tag{10.22}$$

and furthermore

$$\begin{gathered}\textit{All facts of } D \textit{ are true in } M \ \&\ M \textit{ and } \mathcal{M} \textit{ are causally consistent} \\ \Rightarrow \\ \mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}\end{gathered} \tag{10.23}$$

Lemma 10.23 *For any T_{C} for an instance of our language and any domain description D for a language $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$ such that $T_{\mathrm{C}} \sim D$, let $M = (\Psi, \Sigma)$ be any interpretation of D such that M satisfies conditions* **C1** *and* **C2** *of Definition 10.19. Then there exists an interpretation $\mathcal{M}$ of our language such that M and $\mathcal{M}$ are causally consistent.*

Lemma 10.24 *For any T_{C} for an instance of our language and any non-ambiguous domain description D for a language $\mathcal{L}_3 = (\mathcal{F}, \mathcal{A}, \mathcal{S})$ such that $T_{\mathrm{C}} \sim D$, let $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$. Then there exists an interpretation $M = (\Psi, \Sigma)$ of $\mathcal{L}_3$ such that M and $\mathcal{M}$ are causally consistent.*

Proofs of lemmas 10.22-10.24

Proof of Lemma 10.22 (first part) We prove the contrapositive of (10.22). Suppose not all facts of D hold in M. By Lemma 10.20, the fact that $M \cong \mathcal{M}$ and the definition of corresponding theories (Definition 10.9) it follows that $\mathcal{M} \not\models$ CONS and therefore $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$. $\Box$

Proof of Lemma 10.22 (second part) Suppose we have an M and $\mathcal{M}$ such that $M \cong \mathcal{M}$ and (10.23) holds. We show in three stages that $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$: in stage 1, we show $\mathcal{M} \models$ CONS. We use this to show in stage 2 that $\mathcal{M} \models$ *Circum*(CONS; *Do*). In stage 3, we show that $\mathcal{M} \models$ EQ. Stage 1–3 together show that $\mathcal{M}$ satisfies condition (1) of the definition of models for causal theories (Definition 9.16, page 226). That $\mathcal{M}$ satisfies condition (2) of that definition follows already from the definition of '$\cong$' (Definition 10.10, item (a)).

Stage 1 We show that $\mathcal{M} \models \phi$ for all axioms ϕ contained in CONS. From Definition 10.9, item 2, it follows that CONS contains UNA- and DC-axioms for $\mathbf{B}$, $\mathbf{E}$ and $\mathbf{F}$. It follows directly from the fact that M and $\mathcal{M}$ are causally consistent that these axioms hold for $\mathcal{M}$ too. We now consider the axioms of form (10.17) in CONS (Definition 10.9 item 4). Let us denote by ϕ any of these axioms. ϕ can be written as $\forall t.\mathbf{H}(t) \supset Do(F_i^b, t+1)$ where $\mathbf{H}$ is some conjunction of instances of *Ho*. We now show that for all $t \geq 0$, $\mathcal{M} \models \mathbf{H}(t) \supset Do(F_i, b, t+1)$. Since we assume that *time points* are interpreted as the nonnegative integers, this is sufficient to show $\mathcal{M} \models \phi$. So take any $t \geq 0$. For this t, either $\mathcal{M} \not\models \mathbf{H}(t)$ and we are done. If $\mathcal{M} \models \mathbf{H}(t)$ then by the definition of causal consistency, we have $\mathcal{M} \models Do(F_i, b, t+1)$ and we are done again.

It follows from Lemma 10.20 and the fact that all facts of D are true in M that $\mathcal{M} \models \phi$ for all axioms ϕ in T_{C} mentioned in items 5–8 of Definition 10.9 (all these axioms correspond to some fact in D). From item 3 of Definition 10.9, we conclude that CONS contains no more axioms than those for which we already checked that they hold in $\mathcal{M}$. This completes stage 1.

Stage 2 We show that $\mathcal{M} \models$ *Circum*(CONS; *Do*). Notice first that by Proposition 9.17 all models share the same universes for CONS. From this and the characterization of circumscription given in Proposition 9.29 it follows that it is sufficient to show that among all the models for CONS with the same interpretation of *Ho*, there is no $\mathcal{M}'$ whose interpretation of *Do* is a proper subset of that of $\mathcal{M}$. Suppose, by means of contradiction, that such an $\mathcal{M}'$ exists, i.e. $\mathcal{M}' \models$ CONS, $\mathcal{M}[Ho] = \mathcal{M}'[Ho]$ but $\mathcal{M}'[Do] \subsetneq \mathcal{M}[Do]$. By the definition of causal consistency, item 2, $\mathcal{M}$ and therefore also $\mathcal{M}'$ have $\neg Do(e, b, t)$ for any e, b, t. It follows there is a pair $(F_{i^*}^{b^*}, t^*)$ with $\mathcal{M}' \models \neg Do(F_{i^*}, b^*, t^* + 1)$ while $\mathcal{M} \models Do(F_{i^*}, b^*, t^* + 1)$. It follows from the definition of causal consistency, item 3 that T_{C} contains an axiom ϕ of form (10.17) where (1) the F_i,b and t mentioned in the right-hand side of ϕ are equal to F_{i^*},b^* and t^* and (2) the left-hand side of ϕ holds in $\mathcal{M}$, and therefore also in $\mathcal{M}'$. But also $\mathcal{M}' \models$ CONS so $\mathcal{M}' \models Do(F_{i^*}, b^*, t^*)$ and we arrive at a contradiction.

Stage 3 We show that $\mathcal{M} \models$ EQ by showing that axioms (9.33) and (9.34), (9.36) and (9.47) hold in $\mathcal{M}$. By items 2 and 4 of the definition of causal consistency and the fact that M and $\mathcal{M}$ are causally consistent it follows that (9.47) holds in $\mathcal{M}$. Concerning (9.33) and (9.34), we must show that for all $x \in \mathbf{E} \cup \mathbf{F}, t \in \mathbf{N}_0$, if $\mathcal{M} \models Do(x, \text{TRUE}, t)$ then we also have $\mathcal{M} \models Ho(x, t)$ and similarly, if $\mathcal{M} \models Do(x, \text{FALSE}, t)$

then we also have $\mathcal{M} \models \neg Ho(x,t)$. Let us thus assume $\mathcal{M} \models Do(x,b,t)$ for some $x \in \mathbf{E} \cup \mathbf{F}$, $t \in \mathbf{N}_0$ and $b \in \mathbf{B}$. Since $\mathcal{M} \models \neg Do(e,b,t)$ for any e, b and t (see the definition of causal consistency), we may assume $x = F_i^b$ for some $F_i \in \mathbf{F}$. From the same definition, it follows $t \geq 1$ and that CONS must contain an axiom ϕ of form (10.17) such that the right-hand side of ϕ is equal to $Do(F_i, b, t)$ and the left-hand side holds in $\mathcal{M}$. It follows D contains an effect law of form (10.18), i.e. of form $\{a^1, \ldots, a^n\}$ **causes** F_i^b **if** $F_{j_1}^{b_1}, \ldots, F_{j_m}^{b_m}$. Since $M = (\Psi, \Sigma)$ corresponds to $\mathcal{M}$ we have that $\{a^1, \ldots, a^n\} \subseteq a_{t-1}$ and $F_{j_1}^{b_1}, \ldots, F_{j_m}^{b_m}$ all hold in σ_{t-1}. By the following claim we have that F_i^b is an effect of a_{t-1} in σ_{t-1}.

Sublemma 10.25 *Let σ be a state for a non-ambiguous domain description D. Let f be a fluent literal and a be a compound action. Now suppose there is $b \subseteq a$ such that f is an* immediate *effect of b in σ. Then f is an (ordinary) effect of a in σ.*

Proof: follows almost directly from the definitions of immediate effect, inherited effect (Section 10.8.1 and ambiguity (Definition 10.7) . We omit the details. □

By the definition of causal consistency Ψ is a causal model of D and $M \cong \mathcal{M}$. Since $M \cong \mathcal{M}$, $\Psi(\Sigma(s_N))$ is defined. We now have by the claim below that σ_t is defined.

Claim Let Ψ be a causal model and Σ be a situation assignment for some domain description D. If $\Psi(\Sigma(s_N))$ is defined, then for all $t \geq 0$, σ_t is defined.
Proof: is immediate from the definition of σ_t in Definition 10.10. □

Since σ_t is defined, it follows by Definition 10.15 that $\sigma_t = Res(a_{t-1}, \sigma_{t-1})$. Since F_i^b is an effect of a_{t-1} in σ_{t-1}, this means F_i^b is contained in σ_t. Since $M \cong \mathcal{M}$, we have $\mathcal{M} \models Ho(F_i^b, t)$, which is what we had to prove.
We will now show that $\mathcal{M} \models$ (9.36). For this, suppose $\mathcal{M} \models \neg Ho(F_i^b, t-1) \wedge Ho(F_i^b, t)$ for some F_i^b and t. One can show $\mathcal{M} \models Do(F_i, b, t)$ in a manner analogous to (but simpler than) the reasoning in the first part of stage 3, above. We omit the details of this part of the proof. Since we can show this for *any* $t > 0$, *any* b and *any* F_i, it follows by Proposition 9.17 that the following axiom holds for $\mathcal{M}$:

$$\begin{aligned}&\forall f, t\ .(t > 0) \supset \\ &\quad \neg[\ Ho(f,t) \equiv\ Ho(f, t-1)\ ] \supset [\ Do(f, \text{TRUE}, t) \quad \vee \quad Do(f, \text{FALSE}, t)\ ]\end{aligned}$$

which is clearly equivalent to (9.36). This ends the proof of Lemma 10.22. □

Proof of Lemma 10.23 M is an interpretation of (Ψ, Σ) so, by Definition 10.16, $\Psi(\Sigma(s_N))$ is defined; this means Definition 10.10 applies. Items 1 and 2 of that definition determine the interpretation of $Ho(x,t)$ for all $x \in \mathbf{E} \cup \mathbf{F}$ and all $t \geq 0$ in a way that clearly cannot lead to any contradiction. Also, it is clear that item 3 in the definition cannot lead to any contradictory assignment in $\mathcal{M}$ of time name *symbols* to time *points*. Hence an interpretation $\mathcal{M} \cong M$ exists. Now items 2 to 5 of the definition of causal consistency determine the interpretations of all atoms in $\mathcal{M}$ other than those of the form $Ho(x,t)$, again in a way that cannot give rise to any contradiction. □

Proof of Lemma 10.24 It follows immediately from the fact that $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ and the form that CONS must have according to Definition 10.9, that conditions 2 and 3 of the definition of causal consistency hold for $\mathcal{M}$. Since condition 2 holds and $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ and so, in particular, $\mathcal{M} \models \mathit{Circum}(\text{CONS}; \mathit{Do})$, condition 4 holds too. By Proposition 9.17 condition 5 holds too. Hence it only remains to prove that an interpretation $M = (\Psi, \Sigma)$ exists with $M \cong \mathcal{M}$ such that Ψ is a causal model for D. Using Definition 10.10 we can construct such an M as follows:

1. We set $\Sigma(s_N) = [a_0, \ldots, a_{last}]$ where a_i is defined as in Definition 10.10, item 1. We define, for any $s \in \mathcal{S}$, $\Sigma(s)$ to be the prefix of $\Sigma(s_N)$ of length t for the t for which $\mathcal{M} \models s = t$. This completes the definition of Σ.
2. We now define Ψ: we set for all $t \geq 0$, $\Psi([a_0, \ldots, a_{t-1}]) = \sigma_t$ (we use the convention that $[a_{-1}] = []$, so $\Psi([]) = \sigma_0$), where the a_i's are defined as above and σ_t is defined as in Definition 10.10, item 2. Let us denote by the Σ-*sequences* those sequences that start with a prefix of $\Sigma(s_N)$ (note this includes the empty sequence). Now for *any* sequence of compound actions α that is not a Σ-sequence we define $\Psi(\alpha)$ to be such that it satisfies Definition 10.15. Note that this is always possible, since according to that definition $\Psi(\alpha)$ is allowed to be undefined for some α. It can be seen from the definition that, since $\Psi([])$ has already been defined above, this uniquely defines the function Ψ for any sequence that does not start with a prefix of $\Sigma(s_N)$. This completes the definition of Ψ.

For any sequence of compound actions except possibly the Σ-sequences, we know (from the way we defined Ψ) that Ψ satisfies all the effect laws of D. If we can prove that the Σ-sequences, too, satisfy all the effect laws of D, we have by Definition 10.15 that Ψ is a causal model of D and we are done.

We now show that indeed for any Σ-sequence α, $\Psi(\alpha)$ satisfies all effect laws of D. We start with the following claim:

Claim For any Σ-sequence $\alpha \circ a_t = [a_0, \ldots a_{t-1}, a_t]$, we have $E^+_{a_t}(\Psi(\alpha)) \cap E^-_{a_t}(\Psi(\alpha)) \neq \emptyset$.
Proof: Easy and omitted; see [64]. □

From the claim, it follows that we must prove that for any Σ-sequence $\alpha \circ a_t = [a_0, \ldots a_{t-1}, a_t]$, for any $t \geq 0$, we have $\Psi(\alpha \circ a_t) = \mathit{Res}(a_t, \Psi(\alpha))$. From the definition of σ_t and Definition 10.15, we know that this is equivalent to showing that for all $F_i \in \mathcal{F}$, for all $t \geq 0$,

$$F_i \in \sigma_{t+1} \Leftrightarrow F_i \in [\sigma_t \cup E^+_{a_t}(\sigma_t)] \setminus E^-_{a_t}(\sigma_t)] \tag{10.24}$$

We will prove (10.24) using the following

Claim Let either $b = \text{TRUE}$ and $\overline{b} = \text{FALSE}$ or vice versa. For any $\mathcal{M}$ with $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ and any t and F_i we have

$$\mathcal{M} \models \mathit{Ho}(F_i^b, t+1) \Leftrightarrow \mathcal{M} \models [\mathit{Ho}(F_i^b, t) \vee \mathit{Do}(F_i, b, t+1)] \wedge \neg \mathit{Do}(F_i, \overline{b}, t+1)$$

Proof: trivial & omitted. □

We show the if-direction of (10.24). Suppose $F_i \in \sigma_{t+1}$. Then $\mathcal{M} \models Ho(F_i, t+1)$ and by the claim above, $\mathcal{M} \models \neg Do(F_i, \text{FALSE}, t+1)$. This means CONS contains no axiom of form (10.17) with right-hand side $Do(F_i, \text{FALSE}, t+1)$ and left-hand side such that it holds at time t. As $T_C \sim D$ and $M \cong \mathcal{M}$, there is no rule in D of form (10.18) such that $\{a^1, \ldots, a^n\} \subseteq a_t$ and $F_{j_1}^{b_1}, \ldots, F_{j_m}^{b_m}$ all hold in σ_t. This means $\neg F_i$ is not an effect of a_t in σ_t so $F_i \notin E^-_{a_t}(\sigma_t)$.

From our assumption $F_i \in \sigma_{t+1}$ it also follows that either $\mathcal{M} \models Ho(F_i, t)$ or $\mathcal{M} \models Do(F_i, \text{TRUE}, t+1)$. In the first case, $F_i \in \sigma_t$ and we are done; in the second case, since $\mathcal{M} \models_{\mathbf{c}} T_C$ and thus $\mathcal{M} \models Circum(\text{CONS}; Do)$, using Propositions 9.29 and 9.17, there are no models $\mathcal{M}'$ for CONS with $\mathcal{M}'[Ho] = \mathcal{M}[Ho]$ and $\mathcal{M}' \not\models Do(F_i, \text{TRUE}, t+1)$. This means that for some t, CONS must mention $Do(F_i, \text{TRUE}, t+1)$ in one of its axioms. This can only be an axiom ϕ of form (10.17) and it follows that the left-hand side of this axiom, instantiated to time t, holds in $\mathcal{M}$ and hence that D contains an effect law of form (10.18) corresponding to ϕ such that $\{a^1, \ldots, a^m\} \subseteq a_t$ and $F_{j_1}^{b_1}, \ldots, F_{j_m}^{b_m}$ all hold in σ_t. From Sublemma 10.25 on page 263 it follows that F_i is an effect of a_t in σ_t, so $F_i \in E^+_{a_t}(\sigma_t)$. □

The only-if direction of (10.24) can be proven in a manner completely analogously to the if-direction, again using the claim above. We omit the details.

This finishes the proof of Lemma 10.24. □

The Actual Proof

Before actually giving the proof, we have to introduce yet one more proposition which says that the set of event-time pairs holding in a model always corresponds to the set of abnormalities in the model.

Proposition 10.26 *For any T_C for an instance of our language and any domain description D for a language $\mathcal{L}_3$ such that $T_C \sim D$, let $\mathcal{M}$ be any model with $\mathcal{M} \models_{\mathbf{c}} T_C$. For such an $\mathcal{M}$ we have:*

$$\mathcal{M}[Ab_1] = \{(e,t) \mid \mathcal{M} \models Ho(e,t); e \in \mathbf{E}; t \in \mathbf{N}_0\}$$

Proof: Easy and omitted. □

We now first prove the first part of Theorem 10.11, which is restated below.

Theorem 10.11 (part I) *For any theory T_C for our language and any domain description D for a language $\mathcal{L}_3$ of Baral and Gelfond's such that (a) $T_C \sim D$ and (b) D is not* ambiguous, *we have:*

$M = (\Psi, \Sigma)$ is a model for $D \Rightarrow$
there exists an $\mathcal{M}$ with $M \cong \mathcal{M}$ such that $\mathcal{M}$ is a preferred model of T_C

Proof: By Lemma 10.22 and Lemma 10.23 together we have that there exists an $\mathcal{M}$ with $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$ that is causally consistent to M. Clearly $\mathcal{M} \cong M$. Suppose that $\mathcal{M}$ is *not* preferred, i.e. that there is an $\mathcal{M}^* \models_{\mathbf{c}} T_{\mathrm{C}}$ with $\mathcal{M}^*[Ab_1] \subsetneqq \mathcal{M}[Ab_1]$. By Lemma 10.24, this means that there is an $M^* = (\Psi^*, \Sigma^*)$ such that M^* and $\mathcal{M}^*$ are causally consistent and hence condition **C1** of the definition of modelhood (Definition 10.19) is satisfied for M^*. Applying Lemma 10.22 (first part) we find that for such an M^*, condition **C2** of that definition is satisfied too. Since $\mathcal{M}^*[Ab_1] \neq \mathcal{M}[Ab_1]$, we have by Proposition 10.26 and the fact that $\mathcal{M} \cong M$ and $\mathcal{M}^* \cong M^*$ that $\Sigma^*(s_N) \neq \Sigma(s_N)$. We are assuming M is a model of D, so we have by condition **C3** in Definition 10.19 that it is *not* the case that $\Sigma^*(s_N) \leq \Sigma(s_N)$. Then it is not the case either that $a^*_t \subseteq a_t$ for all t (where a_t is defined as in Definition 10.10, and a^*_t stands for the a_t belonging to Σ^*). Since $M \cong \mathcal{M}$ and $M^* \cong \mathcal{M}^*$, it follows by Definition 10.10 and Proposition 10.26 that it is not the case that $\mathcal{M}^*[Ab_1] \subsetneqq \mathcal{M}[Ab_1]$ and we have arrived at a contradiction. $\Box$

We now restate and prove the second part of Theorem 10.11.

Theorem 10.11 (part II) *For any theory T_{C} for our language and any domain description D for a language $\mathcal{L}_3$ of Baral and Gelfond's such that (a) $T_{\mathrm{C}} \sim D$ and (b) D is not* ambiguous, *we have, for all $\mathcal{M}$:*

$$\mathcal{M} \textit{ is a preferred model of } T_{\mathrm{C}} \Rightarrow$$
$$\textit{there exists a model } M \textit{ for } D \textit{ such that } M \cong \mathcal{M}$$

Proof: Clearly, if $\mathcal{M}$ is a preferred model of T_{C} then also $\mathcal{M} \models_{\mathbf{c}} T_{\mathrm{C}}$. By Lemma 10.24 we have that there exists an M that is causally consistent to $\mathcal{M}$ and that therefore satisfies condition **C1** of the definition of modelhood (Definition 10.19). Clearly, $M \cong \mathcal{M}$. By Lemma 10.22 (first part) we further have that M also satisfies condition **C2** of Definition 10.19. We now prove that M is a model of D by showing that assuming that condition **C3** of Definition 10.19 does *not* hold for M leads to a contradiction.

So assume this condition does not hold for M. Then there exists $M' = (\Psi', \Sigma')$ that satisfies **C1** and **C2** of Definition 10.19 with $\Sigma'(s_N) \neq \Sigma(s_N)$ and $\Sigma'(s_N) \leq \Sigma(s_N)$ and $\Psi'(\Sigma'(s_N))$ is defined. By Lemma 10.23 and Lemma 10.22 (second part) together we have that there is an $\mathcal{M}' \models_{\mathbf{c}} T_{\mathrm{C}}$ with $\mathcal{M}' \cong M'$. We will now 'stretch' $\mathcal{M}'$ into another $\mathcal{M}'' \models_{\mathbf{c}} T_{\mathrm{C}}$ which will have the property that $\mathcal{M}''[Ab_1] \subsetneqq \mathcal{M}[Ab_1]$. In $\mathcal{M}''$ exactly the same events and changes take place as in $\mathcal{M}'$, but the time in between two events/changes may be larger. We now prepare the construction of $\mathcal{M}''$:

Let $\Sigma(s_N) = [a_0, \ldots, a_{last}]$ and $\Sigma'(s_N) = [a'_0, \ldots, a'_{last'}]$. Notice first that, since $\Sigma'(s_N) \neq \Sigma(s_N)$ and $\Sigma'(s_N) \leq \Sigma(s_N)$, there is a strictly increasing sequence of time points $x_0, \ldots, x_{last'}$ such that for all $0 \leq t \leq last'$ we have $\{e : \mathcal{M}' \models Ho(e,t)\} \subseteq \{e : \mathcal{M} \models Ho(e, x_t)\}$ where for at least one t, the inclusion is proper. Also, for all $t > last'$, $\{e : \mathcal{M}' \models Ho(e,t)\}$ is empty.

Now define $\mathcal{M}''$ as follows: for all $0 \leq t \leq last'$, the extension in $\mathcal{M}''$ of Ho, Ab_1 and Do at time x_t is defined to be equal to its extension in $\mathcal{M}'$ at time t; i.e. for all x: $\mathcal{M}' \models Ho(x,t) \Leftrightarrow \mathcal{M}'' \models Ho(x, x_t)$ and analogously for Do and Ab_1. We now have to 'fill up the gaps' in $\mathcal{M}''$, i.e. define the interpretations of Ho and Do for the time points in $\mathcal{M}''$ that are not equal to any of the $x_0, \ldots, x_{last'}$. For this, we set $x_{-1} := -1$. For all $0 \leq t \leq last'$, for all $x_{t-1} < t' < x_t$ and $t' > x_{last'}$, define:

- for all $e \in \mathbf{E}$: $\mathcal{M}'' \models \neg Ho(e,t') \wedge \neg Ab_1(e,t')$.
- for all $x \in \mathbf{E} \cup \mathbf{F}$: $\mathcal{M}'' \models \neg Do(x, \text{TRUE}, t') \wedge \neg Do(x, \text{FALSE}, t')$.
- for all $f \in \mathbf{F}$: $\mathcal{M}'' \models Ho(f,t') \Leftrightarrow \begin{cases} \mathcal{M}' \models Ho(f,t) & \text{if } t' < x_t \\ \mathcal{M}' \models Ho(f, last'+1) & \text{if } t' > x_{last'} \end{cases}$

Clearly, all predicates in the language for $\mathcal{M}''$ have now been defined for all t, f, e, b (notice that Ab_1 only exists for event-time pairs, not for fluent-time pairs, see Chapter 9, Section 9.6.5).

One easily verifies that an $\mathcal{M}''$ as defined above indeed exists, and that from the fact that $\mathcal{M}' \models_{\mathbf{c}} T_{\text{C}}$, it follows that also $\mathcal{M}'' \models_{\mathbf{c}} T_{\text{C}}$ (simply check all the axioms in CONS and EQ; we omit the details). We see that $\{(e,t)|\mathcal{M}'' \models Ho(e,t)\} \subseteq \{(e,t)|\mathcal{M} \models Ho(e,t)\}$ and hence by Proposition 10.26 $\mathcal{M}''[Ab_1] \subseteq \mathcal{M}[Ab_1]$. Since $\Sigma'(s_N) \neq \Sigma(s_N)$, by construction of $\mathcal{M}''$ and Proposition 10.26 we also have $\mathcal{M}''[Ab_1] \neq \mathcal{M}[Ab_1]$. This means $\mathcal{M}''$ is preferred over $\mathcal{M}$ and hence $\mathcal{M}$ is not a preferred model of T_{C} , in contradiction with our assumption. $\square$

Epilogue: Nonmonotonicity and Probability

In Chapter 9, Section 9.6.5, we extended our causal theories with a nonmonotonic component so as to be able to handle causal chains of events and 'surprises'. We used the simplest form of preferential semantics - simply picking the models of a theory with a minimal interpretation of the Ab-predicate. In this Epilogue, we show how this form of nonmonotonicity can be given a probabilistic interpretation, and we shall argue that there is advantage in taking a probabilistic point of view. However, we stress at the outset that we do *not* claim that '*all* uses of nonmonotonicity can (or should) be dealt with probabilistically'; more about this below.

To establish the connection between minimizing abnormality and probability theory, we need to invoke two controversial ideas: the 'Principle of Insufficient Reason' (page xi) (or equivalently, the Maximum Entropy Principle, Chapter 3) and the idea of viewing 'abnormalities' as 'events with small probability'. Both ideas have been used before to endow non-monotonic logics with probabilistic semantics [116, 58, 59, 114, 5]; however, their use remains controversial [114].

Some logicians feel that a probabilistic treatment of uncertainty, which necessarily involves such controversial steps, may lead to the inference of unwarranted conclusions. We will see that, the way they are used here, both the principle of insufficient reason and the probabilistic view on abnormalities can be understood from a point of view inspired by MDL. We shall argue that, if this MDL interpretation is taken seriously, then no unwarranted conclusions can be drawn. On the contrary, we shall show that the probabilistic/MDL interpretation forces one to explicitly state certain assumptions that remain hidden in the non-probabilistic approach. By establishing a connection between minimizing abnormality and the MDL Principle, we relate the present (third) part of the thesis to the first two parts.

Before We Start: Non-Probabilistic Nonmonotonicity Exists!

Before we start we emphasize that we do not claim *all* forms of nonmonotonic reasoning to be related to probability theory. In Chapter 8, Section 8.4 we presented an 'alternative research goal' in which we distinguished between the use of nonmonotonicity for theory completion (implemented by a nonmonotonic entailment relation $\approx$ which we called *completion mapping*) and the use of nonmonotonicity to impose a preference ordering over the set of possible realizations of a domain. In our 'causal theories' of chapters 9 and 10 the distinction returned: the minimization of Do corresponded to a completion mapping, the minimization of Ab_1 and Ab_2 corresponded to finding the most preferred (least surprising) models of our (already completely specified) domains. Here, we shall only be concerned about the latter use of nonmonotonicity, the minimization of abnormality in order to find the most preferred models of a 'completely specified' domain. We do not claim that the use of nonmonotonicity for theory completion is in any way related to probability theory.

1 Minimizing Abnormality and Maximizing Probability

Preliminaries

For simplicity, we will only consider the propositional case (we say something about the first-order case later). We start by assuming the notation and terminology introduced in Chapter 9, page 210. We introduce some additional notation concerning probability distributions over sets of models. Let $\mathbf{X} = \{X_1, \ldots, X_m\}$ be a finite set of propositional variables. By a probability distribution over $\mathbf{X}$ we mean a probability distribution over the sample space $\mathbf{E}$ consisting of all interpretations of $\mathbf{X}$. We write $P(\mathcal{M})$ to denote the probability of interpretation $\mathcal{M}$. Let $\mathbf{M}$ be some set of interpretations over $\mathbf{X}$. By $P(\mathbf{M})$ we denote the probability of the 'event' that the actual interpretation is contained in $\mathbf{M}$. Hence $P(\mathbf{M}) = \sum_{\mathcal{M} \in \mathbf{M}} P(\mathcal{M})$.

For a propositional formula Γ, we define $P(\Gamma) = P(\mathbf{M})$ where $\mathbf{M}$ is the set of $\mathcal{M}$ with $\mathcal{M} \models \Gamma$. Hence $P(\Gamma) = \sum_{\mathcal{M} \models \Gamma} P(\mathcal{M})$. Conditional probability is defined as usual: $P(\Gamma_1 | \Gamma_2) = P(\Gamma_1 \wedge \Gamma_2)/P(\Gamma_2)$. We freely write $P(X_i = \text{TRUE})$ to denote $P(X_i)$ and $P(X_i = \text{FALSE})$ to denote $P(\neg X_i)$. Note that P is defined both over sets of models (as in $P(\mathbf{M})$) and over propositions (as in $P(\Gamma)$).

For now we shall assume that there are only two kinds of elements in $\mathbf{X}$: those whose value we are ignorant about (or, more properly, about which we want to make no assumptions) and those whose truth is considered 'abnormal' or 'unusual'. We can therefore partition $\mathbf{X}$ as follows: $\mathbf{X} = \mathbf{Ind} \cup \mathbf{Ab}$. (**Ind** being short for 'indifferent'). **Ind** contains the atoms we want to make no assumptions about while **Ab** contains the atoms that represent abnormalities.

Preferred Models According to the simple form of nonmonotonic reasoning introduced on page 184 and used in Chapter 9, Section 9.6.5, we should prefer the models of a theory T with the smallest interpretation of the elements in **Ab**; here we call the models that are selected in this way the L-preferred models:

Definition E.1 *We call a model $\mathcal{M}$ for a propositional theory T an L-preferred model of T iff there is no model $\mathcal{M}'$ for T with*

$$\{Ab \mid Ab \in \mathbf{Ab}, \mathcal{M}' \models Ab\} \subsetneqq \{Ab \mid Ab \in \mathbf{Ab}, \mathcal{M} \models Ab\} \tag{E.1}$$

(note that Ab is used as a meta-variable here whose value can be any of the variables X_i that are contained in $\mathbf{Ab}$).

From the probabilistic point of view, our 'preference ordering' over models should be expressed by a probability distribution P over the space of models and we should prefer the models with maximum probability[6]. We will now show that the minimization of abnormality corresponds to picking the most probable model when a certain class of probability distributions is used. We call these distributions *simple*:

Definition E.2 *Let P be a distribution over $\mathbf{X} = \mathbf{Ind} \cup \mathbf{Ab}$. We call P* simple *iff*

1. *All variables in $\mathbf{X}$ are independent: for all $(b_1, \ldots, b_m) \in \mathbf{B}^m$ and $1 \leq i \leq m$,*

$$P(X_i = b_i | X_1 = b_1, \ldots, X_{i-1} = b_{i-1}, X_{i+1} = b_{i+1}, X_m = b_m) = P(X_i = b_i).$$

2. *for all $Ind \in \mathbf{Ind}$, $P(Ind = \text{TRUE}) = P(Ind = \text{FALSE}) = 1/2$.*

3. *for all $Ab \in \mathbf{Ab}$, $P(Ab = \text{TRUE}) = \epsilon_{Ab}$. Here $0 < \epsilon_{Ab} < 1/2$.*

A precise interpretation and discussion of Definition E.2 will be given in Section 2. Briefly, item (1) is an assumption that is implicit in the use of abnormalities. Item (2) expresses our ignorance about the variables in $\mathbf{Ind}$. Item (3) expresses that $\neg Ab$ is preferred over Ab. We will be interested in the following case: our preference ordering over models can be modeled by some simple P, but it is not known which. Hence the specific values of $P(Ab = \text{TRUE}) = \epsilon_{Ab}$ are unknown.

We call $\mathcal{M}$ a *probabilistically preferred* model for theory T under P if $\mathcal{M}$ is one of the models that maximize $P(\mathcal{M}|T)$ ('the probability of $\mathcal{M}$ given that the set of propositions T holds'). We call $\mathcal{M}$ a *potentially probabilistically preferred model* for T if there exists a simple P over $\mathbf{X}$ under which $\mathcal{M}$ is a preferred model for T: if we are given theory T but the only thing we know about P is that it is simple, then it seems reasonable to consider *every* model that is potentially preferred. We write PP-preferred as short for 'potentially probabilistically preferred'. The proposition below shows that picking the PP-preferred models is equivalent to picking the models which are minimal in the usual sense of nonmonotonic logic:

Proposition E.3 *$\mathcal{M}$ is a PP-preferred model for theory T iff $\mathcal{M}$ is an L-preferred model for theory T.*

Proof: (only-if) Suppose $\mathcal{M}$ is a PP-preferred model for T. Let us assume, by means of contradiction, that $\mathcal{M}$ is not L-preferred. Then by definition there exists a model $\mathcal{M}' \models T$ such that (E.1) holds for $\mathcal{M}'$ and $\mathcal{M}$. We will show that, for every simple P, $P(\mathcal{M}'|T)$ must be larger than $P(\mathcal{M}|T)$. This implies that $\mathcal{M}$ is *not* a PP-preferred model of T and hence gives the desired contradiction.

[6] Some people think it strange that preferred models should coincide with 'the most probable' models. From both a Bayesian (subjective) and an MDL-point of view this makes perfect sense as we shall discuss further on.

To show this, note first that, since both $\mathcal{M} \models T$ and $\mathcal{M}' \models T$, we have $P(\mathcal{M}'|T) > P(\mathcal{M}|T) \Leftrightarrow P(\mathcal{M}') > P(\mathcal{M})$. Define $\mathcal{M}[X_i]$ to be the $b_i \in \mathbf{B}$ for which $\mathcal{M} \models X_i \equiv b_i$. Since P is simple, we have by Definition E.2:

$$P(\mathcal{M}') = \prod_{1 \leq i \leq m} P\{X_i = \mathcal{M}'[X_i]\} = P(\mathcal{M}) \cdot \prod_{Ab \in \mathbf{D}} \frac{1 - P(Ab)}{P(Ab)} = P(\mathcal{M}) \cdot \prod_{Ab \in \mathbf{D}} \frac{1 - \epsilon_{Ab}}{\epsilon_{Ab}}$$

where $\mathbf{D}$ is the non-empty set $\{Ab \in \mathbf{Ab} \mid \mathcal{M} \models Ab \text{ and } \mathcal{M}' \not\models Ab\}$. Since all $\epsilon_{Ab} < 1/2$, this shows that $P\{\mathcal{M}'\} > P\{\mathcal{M}\}$.

(if) Assume that $\mathcal{M}$ is an L-preferred model. Let P be such that for all $Ab \in \mathbf{Ab}$ with $\mathcal{M} \models Ab$, $P(Ab) = 1/2 - \epsilon$ where $\epsilon > 0$ is some small number, while $P(Ab') = \epsilon$ for all $Ab' \in \mathbf{Ab}$ with $\mathcal{M} \not\models Ab'$. Clearly, P is simple. We are assuming that $\mathcal{M}$ is L-preferred so there is no model $\mathcal{M}'$ for T such that (E.1) holds. This means that all $\mathcal{M}' \neq \mathcal{M}$ with $\mathcal{M}' \models T$ and $P(\mathcal{M}') \neq P(\mathcal{M})$ satisfy $\mathcal{M}' \models Ab'$ for some $Ab' \in \mathbf{Ab}$ with $\mathcal{M} \not\models Ab'$. This implies that by picking $\epsilon > 0$ small enough, we can get $P(\mathcal{M}) > P(\mathcal{M}')$ and, since $\mathcal{M} \models T$ and $\mathcal{M}' \models T$, also $P(\mathcal{M}|T) > P(\mathcal{M}'|T)$. This implies that $\mathcal{M}$ is a PP-preferred model. □

Summarizing, if we are willing to assume that P is simple but otherwise unknown, then the two approaches to handling abnormalities can be reconciled. We will discuss how exactly P should be interpreted and whether or not the assumption that P is simple can be justified in the next section. First, we show how to extend the definition of 'simple' P to domains involving several levels of abnormalities and to the first-order setting we considered in Chapter 9.

1.1 Extension to Several Levels of Abnormalities

We may extend our concept of 'simple probability distributions' to several 'levels' of abnormality by further partitioning $\mathbf{X}$. For example, if we need two levels, we may set $\mathbf{X} = \mathbf{Ind} \cup \mathbf{Ab}_1 \cup \mathbf{Ab}_2$. In this new setting, a 'simple' distribution P becomes a P such that all variables in $\mathbf{X}$ are still independent, while

$$\begin{aligned} &\text{for all } Ab_1 \in \mathbf{Ab}_1: \quad && P(Ab_1) < 1/2\ ; \\ &\text{for all } Ab_2 \in \mathbf{Ab}_2: \quad && 0 < P(Ab_2) < \min_{Ab_1 \in \mathbf{Ab}_1} (P(Ab_1))^k. \end{aligned} \tag{E.2}$$

where k is the number of propositional variables in $\mathbf{Ab}_1$. One easily verifies that in this way, models without abnormalities in $\mathbf{Ab}_2$ will always be PP-preferred over models with abnormalities in $\mathbf{Ab}_2$, no matter the number of abnormalities in $\mathbf{Ab}_1$. Such a two-step hierarchy of abnormalities is often needed in nonmonotonic reasoning; for example, in Chapter 9, Section 9.6.5 we needed one level of abnormality to model surprises and another level to model the occurrence of unexpected events, since 'surprises' where considered more 'abnormal' than unexpected events (recall that a surprise was defined as a change of fluent value without an event according for the change). Clearly, if needed we can also introduce a set $\mathbf{Ab}_3$ of abnormalities associated with even smaller probabilities, and in the same way a set $\mathbf{Ab}_4$, $\mathbf{Ab}_5$ etc. In this way we can probabilistically represent every finite hierarchy of abnormalities.

The First-Order Case Thus far, our analysis has only been for propositional domains. In general, it is not clear how to extend the present mapping of preference orderings to probability distributions to the full first-order case (for an attempt to establish a connection between probability theory and first-order logic, see [5]). Nevertheless, as soon as a domain is essentially finite (and hence could in principle be modeled by a propositional theory), an appropriate version of 'simple' probability distributions can be defined and the mapping holds again.

Therefore, if we restrict the first-order theories that we considered in Chapter 9 to domains containing only a finite number of fluents, events and time points, the minimization of abnormality introduced in Section 9.6.5 can once more be interpreted probabilistically. In fact, the minimization procedure given by Definition 9.23 of page 230 can be given a probabilistic interpretation completely analogously to the minimization procedure defined in Definition E.1 of this Epilogue. We will not show this in full detail; instead, we will only roughly indicate how it may be achieved. Let T be a causal theory defined as in Definition 9.27, page 232. By restricting ourselves to a finite number of time points and a finite set of fluents, we can treat the whole domain by propositional logic and regard each instantiation of each predicate as a separate atom. We can then regard each instantiation of Ab_1 as a separate weak abnormality and each instantiation of Ab_2 as a separate strong abnormality. Definition 9.23 then reduces to the following preference scheme: select the causal models for T with the smallest (in the subset sense) number of abnormalities $Ab_2 \in \mathbf{Ab}_2$ and, among these, further select the models with the smallest (in the subset sense) number of $Ab_1 \in \mathbf{Ab}_1$. It is easy to show (analogously to Proposition E.3) that this is equivalent to selecting the potentially preferred causal models for T, if we assume a 'simple' probability distribution where 'simple' is now defined also for $Ab_2 \in \mathbf{Ab}_2$ (as in Equation E.2).

2 Interpretation in the Spirit of MDL

2.1 Bayesian interpretation

From the Bayesian (subjectivist) view on probability theory, probabilities can be used to represent degrees of belief. If an agent assigns a high probability to proposition A, this means that he (it) has a high degree of belief in A. The notion of 'degree of belief' can be given a concrete interpretation in terms of betting games: the probability $P(A)$ that an agent assigns to A determines at what odds the agent would accept the bet 'A is true': if $P(A) = p$, then the agent should accept a bet at odds 1-to-q if $q < p$ and he should refuse such a bet if $q > p$ (this interpretation has been emphasized by De Finetti; see [38, 114]).

From the point of view of the MDL Principle, the connection between probabilities and betting is somewhat different and more direct: the MDL Principle allows us to *identify* probability distributions with betting strategies. This turns the probabilistic interpretation of default reasoning into a much less abstract 'gambling' interpretation, as we will now show.

2.2 MDL Interpretation

As we discussed in Chapter 2, Section 2.2, the probability distributions that MDL is concerned with are to be interpreted as *models* of data, not to be confused with the traditional notion of a probability distribution 'according to which data are drawn'. We saw that each such a 'modeling' distribution P can be interpreted as a code with codeword lengths $L(D) = -\log P(D)$ for all D; the better P fits D, the shorter the codelength $L(D)$ (and conversely, each code can be interpreted as a probability distribution). We also saw (Chapter 2, Section 2.7) that every 'modeling' probability distribution P may be interpreted as defining a *betting strategy.* We will now explain what this means in a logical framework.

Betting Strategies in a Logical Framework In a logical framework, P is defined over the models of a logical theory T. The betting strategy concerns situations in which the agent does not know the true model $\mathcal{M}$ and it can express its beliefs about the actual situation by a probability distribution P over the set of possible models $\mathbf{M}$. The phrase 'an agent adopts distribution P' means the following: imagine the agent is asked to take part in a game in which it would have to place bets on the different models at equal odds (i.e. it has to divide its capital over all models $\mathcal{M}$; then, the true model $\mathcal{M}$ is revealed and an amount of c times the bet placed on $\mathcal{M}$ is paid to the agent, where c is equal for all $\mathcal{M}$). In such a game, the agent would divide its capital over the set of possible models such that the capital put on model $\mathcal{M}$ is proportional to $P(\mathcal{M})$. If the agent is rational, then $P(\mathcal{M}_1) > P(\mathcal{M}_2)$ iff the agent thinks that $\mathcal{M}_1$ is more likely than $\mathcal{M}_2$. We will henceforth assume that an agent always adopts the betting strategy that it considers optimal.

It is important to note that the agent invests some of its capital in each x with $P(x) > 0$. This MDL-way of associating bets with probabilities is different from the Bayesian way we described above. To see this, let $\mathbf{E} = \{1, 2\}$. In the Bayesian (De Finetti's) view, if an agent's belief is expressed by $P(X = 1) = 1/2$, this means it is willing to invest *every* fraction $q < 1/2$ of its capital in the game with pay-off 1 if X turns out to be 1 and pay-off 0 otherwise. From the MDL point of view (at least in our interpretation of MDL), $P(X = 1) = 1/2$ is primarily identified *only* with a willingness to invest 50 % of one's capital on outcome 1 and 50 % on outcome 2, where both bets are placed at odds 1-to-2 (see Chapter 2, Section 2.7). Under this bet the total amount of capital will not change whatever the actual outcome. It thus constitutes an 'empty' bet, that can be properly used to express complete ignorance about which of the two outcomes will actually take place.

Armed with our interpretation of probability distributions as gambling strategies, we will now interpret the definition of 'simple' probability distributions (Definition E.2) in gambling terms. Using Proposition E.3 this will also give us an interpretation of the models 'with the least abnormalities' in gambling terms.

Definition E.2 Revisited Assume an agent uses a 'simple' distribution P (Definition E.2) to express its knowledge about a domain involving (1) abnormalities (variables in $\mathbf{Ab}$) and (2) atoms about which it does not want to make any assumptions (variables in $\mathbf{Ind}$). The question is whether P, viewed as a betting strategy, correctly represents the agent's qualitative knowledge about the domain.

Simple distributions P were defined in three steps. In the first step, all variables in $\mathbf{X}$ were assumed to be independent. In betting terms, this says that the agent's bets on the outcome of variable X_i are *not influenced* by knowledge of the outcomes of (any combination of) the other variables $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_m$. Whether this 'subjective independence' assumption can be justified or not will be discussed below.

The second step in the definition of simple P assigns the uniform distribution over the variables the agent wants to make no assumptions about. It is a form of the Principle of Insufficient Reason (and as such, a special case of Maximum Entropy, and as such, a version of MDL; see Chapter 3). It says that the agent should place an 'empty' bet (see above) on the variables it wants to make no assumptions about: it places half of its capital on Ind = TRUE and at the same time half of its capital on Ind = FALSE at odds 1-to-2. This is clearly the best it can do if it wants to make no assumptions about Ind, since every other assignment of probabilities (bets) leads to a potentially higher loss in betting.

The third item in Definition E.2 says that the agent is prepared to bet a larger percentage of its capital on outcome $\neg Ab$ than on outcome Ab; since ϵ_{Ab} is not specified, the item says nothing about the exact percentage. This item merely translates the agent's qualitative beliefs about a domain in terms of betting strategies. Items 1 and 2 need more explanation which follows below. We start with item 1.

Independence Assumption Item 1 expresses that our 'degree of belief' in proposition $X_i = b_i$ is not influenced by variables different from X_i. This will of course not be justified in all domains. In domains where it is not, it will lead to 'stupid' bets. But Proposition E.3 tells us that in such domains, minimization of abnormalities will also lead to a preference of 'stupid' models. A good example is the Yale Shooting domain in its original formulation (page 190), where the 'law of persistence' was formalized as follows:

$$\forall f, t \,.\, \neg Ab(f,t) \supset [Ho(f,t) \equiv Ho(f,t+1)]$$

If we use a probability distribution P' over the atoms in the language that corresponds to our intuitive beliefs about the domain, then the following holds: while $Ab(f,t)$ (denoting that fluent f changes value) has a very small probability *a priori*, if no observations are available, it can nevertheless receive a very *high* probability once additional information becomes available. For example, if it is given that $Ho(Shoot, 2) \wedge Ho(Loaded, 2)$ (*Shoot* takes place with a loaded gun) then $Ab(Alive, 2)$ suddenly receives a very high probability: conditioned on $Ho(Shoot, 2) \wedge Ho(Loaded, 2)$, our degree of belief in $Ab(Alive, 2)$ becomes much higher. Hence the percentage of our capital we would be prepared to put on $Ab(Alive, 2)$ is *not* independent of additional knowledge we may have about the domain: the distribution P' which corresponds to our intuitive beliefs does *not* render the abnormalities $Ab(f,t)$ probabilistically independent of the outcomes of the other atoms in the domain. *Such a situation cannot be modeled by a simple minimization of abnormality.*

Two Mistakes In the early days of nonmonotonic reasoning, it was believed that a single mechanism (like minimizing abnormality) could be used to handle most if not all kinds of different defaults. The probabilistic interpretation of abnormalities shows that minimizing abnormality only works under strong independence assumptions. The advantage of the probabilistic formulation is that it makes these assumptions explicit. As such, it 'explains' why the basic kind of default reasoning – based on minimizing abnormality – turned out to be so problematic and brittle.

While the original mistake in the nonmonotonic reasoning community had been to use the mechanism of minimizing abnormality in domains where it cannot be applied, the resulting difficulties led some researchers to make (what we view as) the opposite mistake: they started to distinguish between concepts which, from the probabilistic/betting strategy point of view, are identical. We give a little example. In Geffner [51] we find two kinds of abnormalities: 'explained' and 'unexplained' abnormalities. For example, the abnormality that *Alive* becomes $\neg$*Alive* when a loaded gun is fired is 'explained' while the abnormality of a gun unloading itself is 'unexplained'. In the probabilistic view, an abnormality stands for a proposition which is assigned low probability according to some distribution P. In Chapter 2, Section 2.2, we discussed the MDL interpretation of probability distributions as *models* (not in the logical sense of a mathematical structure): a probability distribution P assigns to each possible realization of data (in this case, to each structure $\mathcal{M}$) a number that indicates how well that data (structure) is *explained* by the model P. As such, an 'abnormality' is by definition 'something that is not well-explained'. In a more proper terminology, what was called an 'explained' abnormality above would then be no abnormality at all. More generally, in the MDL view of probabilities as models, the expression $P(X|Y)$ can be informally regarded as measuring 'how well Y explains X under the model P', or, equivalently, 'how expected X is given Y under model P' or yet equivalently, 'how surprising $\neg X$ is given Y under model P'. Note that the 'event' Y itself may represent a probabilistic model, not necessarily a set of outcomes (logical models). If there is no conditioning event Y, then $P(X)$ can be regarded as expressing how well the model P itself explains X.

Dangerous Use of Probabilities This view of probability distributions as 'explanation measurers' has been criticized by logicians and philosophers [25]. Whereas we just advocated the use of probabilistic semantics on the grounds that it makes explicit some assumptions (about independence) which remain hidden in the abnormality approach, these researchers fear the contrary effect: they argue that by adopting a probability distribution over a set of models one implicitly makes additional, unwarranted assumptions about the domain under consideration. Their objections are often related to the 'Ex Nihilo Nihil' criticism concerning the Maximum Entropy Principle and its special case, the Principle of Insufficient Reason (Chapter 3, Section 3.6.2): it is feared that, by using $P(\text{Ind} = \text{TRUE}) = 1/2$ to express our ignorance about Ind, we are 'getting something (knowledge) out of nothing (ignorance)'. For example, if $\mathrm{Ind}_1, \mathrm{Ind}_2 \in \mathbf{Ind}$, then a simple probability distribution assigns $P(\mathrm{Ind}_1 = \text{TRUE}) = P(\mathrm{Ind}_2 = \text{TRUE}) = 1/2$ and renders Ind_1 and Ind_2 independent. It seems that one can then infer that

$$P((\mathrm{Ind}_1 \vee \mathrm{Ind}_2) = \text{TRUE}) = 3/4.$$

Hence the degree of belief in $\mathrm{Ind}_1 \vee \mathrm{Ind}_2$ would be higher than the degree of belief in $\neg(\mathrm{Ind}_1 \vee \mathrm{Ind}_2)$ and mere ignorance about Ind_1 and Ind_2, together with the knowledge that they are independent, would give us a kind of 'knowledge' about the proposition $\mathrm{Ind}_1 \vee \mathrm{Ind}_2$. However, we used P *only* to determine which models lead to the smallest loss in proportional betting, where the bets where placed sequentially on the outcomes of the variables in $\mathbf{X}$. We did *not* use it to place bets on (or make predictions about) propositions like $\mathrm{Ind}_1 \vee \mathrm{Ind}_2$. The bets we did place on the variables in **Ind** where 'empty', neither decreasing nor increasing our total capital whatever the actual outcome. Therefore, as long as we only use P to infer what model would have given the minimal loss in proportional betting (and *do not use it to infer anything else*), the assignment $P(\mathrm{Ind}_1 = \text{TRUE}) = 1/2$ is completely harmless. The assignment is, in fact, an example of the 'safe' version of Maximum Entropy we introduced in Chapter 4 (Figure 4.1): whatever the actual outcomes of the variables in **Ind**, our *expectation* according to P of the logarithm of our loss in proportional betting on the variables in **Ind** will be equal to the logarithm of the loss we will *actually* make; hence, if only used for proportional betting, P truely satisfies Leslie Ellis' Ex Nihilo Nihil dictum. This is explained at length in Chapter 4, Section 4.2.3; see also page 76.

3 Conclusion: Use Probability Theory, but not Always

We think that many of the domains considered in the nonmonotonic reasoning community can be appropriately modeled by a probabilistic semantics: as soon as the models of a domain are to be ordered in terms of 'how surprising they are', a probabilistic semantics seems appropriate; as should be clear from our 'betting strategy interpretation', such a semantics remains applicable in many cases where probabilities (such as $P(\mathrm{Ind} = \text{TRUE}) = 1/2$) cannot be related to frequencies, as long as we are careful about what inferences to make based on these probabilities. Even when defaults stand for conventions (see the example on page 186) they often induce an ordering on the models of a domain in terms of how 'surprising' they are. In such a case, it seems to us, a probabilistic semantics is appropriate. Above we showed that it can make explicit certain (independence) assumptions underlying the reasoning; many additional reasons can be found in [116] (for a general discussion and criticism, see [25]).

However, we disagree with those who think that there is no fundamental rôle for nonmonotonic formalisms whatsoever. As we pointed out at the beginning of this Epilogue, nonmonotonic mechanisms can also be used for *theory completion*: instead of writing down the full set of laws to which we want our domain to be subject, we only specify a subset of them and we use a nonmonotonic completion mapping to make these theories 'complete'. In fact, in the first-order version of our theory formulated in

Section 9.6.5, we first used the minimization of *Do* to do this theory completion; this, it seems to us, had nothing to do with probability theory. We then proceeded to select the least surprising models of the completed theory; here a probabilistic semantics is appropriate.

Bibliography

[1] J. Amsterdam. Temporal reasoning and narrative conventions. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR' 91)*, pages 15–21, 1991.

[2] M. Anthony and N.Biggs. *Computational Learning Theory.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1992.

[3] K.R. Apt. *From Logic Programming to Prolog.* Prentice Hall, 1997.

[4] K.R. Apt and M. Bezem. Acyclic programs. *New Generation Computing*, 1991.

[5] F. Bacchus, A.J. Grove, J.Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, pages 75–143, 1997.

[6] L.J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics.* PWS-Kent, Boston, 1989.

[7] A.B. Baker. A simple solution to the Yale shooting problem. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR' 89)*, pages 11–20, 1989.

[8] A.B. Baker. Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence*, 49:5–23, 1991.

[9] V. Balasubramanian. Statistical inference, Occam's Razor, and statistical mechanics on the space of probability distributions. *Neural Computation*, 9:349–368, 1997.

[10] C. Baral and M. Gelfond. Logic programming and knowledge representation. *Journal of Logic Programming*, 19,20:73–148, 1994.

[11] C. Baral and M. Gelfond. Reasoning about actual and hypothetical occurrances of concurrent and non-deterministic actions. In *Proceedings First Workshop on Nonmonotonic Reasoning, Action and Change*, 1995. Available via ftp://ftp.newcastle.edu.au/pub/papers/nrac95/baral.final.ps.

[12] C. Baral, M. Gelfond, and A. Provetti. Representing actions: Laws, observations and hypotheses. *Journal of Logic Programming*, 12:1–44, 1997.

[13] A.R. Barron. Complexity regularization with application to artificial neural networks. In G. Roussas, editor, *Nonparametric Functional Estimation and Related Topics*, pages 561–576, Dordrecht, 1990. Kluwer Academic Publishers.

[14] A.R. Barron and T.M. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37(4):1034–1054, 1991.

[15] R.A. Baxter and J.O. Oliver. MDL and MML: Similarities and differences. Technical Report 207, Department of Computer Science, Monash University, 1994.

[16] A. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 1996.

[17] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics. Springer-Verlag, New York, revised and expanded second edition, 1985.

[18] J.M. Bernardo and A.F.M Smith. *Bayesian theory*. John Wiley, 1994.

[19] C. Boutilier and M. Goldszmidt. The frame problem and Bayesian network action representations. In *Proceedings of the Canadian Conference on Artificial Intelligence (CCAI-96)*, 1996.

[20] H. Bozdogan. Personal Communication.

[21] G. Brewka. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*, volume 12 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1991.

[22] M. Browne. Cross-validation methods. *Journal of Mathematical Psychology*, 1998. To appear.

[23] G.J. Chaitin. On the length of programs for computing finite binary sequences: statistical considerations. *Journal of the ACM*, 16:145–159, 1969.

[24] P. Cheeseman. In defense of probability. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 1002–1009, 1985.

[25] P. Cheeseman. An inquiry into computer understanding. *Computational Intelligence*, 4(1):58–66,67–129, 1988. Discussion: pages 67–129.

[26] B.S. Clarke and A.R. Barron. Information-theoretic asymptotics of Bayes methods. *IEEE Transactions on Information Theory*, IT-36(3):453–471, 1990.

[27] B.S. Clarke and A.R. Barron. Jeffreys' prior is asymptotically least favorable under entropy risk. *Journal of Statistical Planning and Inference*, 41:37–60, 1996.

[28] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, 1990.

[29] G.F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[30] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley Interscience, New York, 1991.

[31] R.T. Cox. *The Algebra of Probable Inference.* Johns Hopkins University Press, 1961.

[32] J.M. Cozzolino and M.J. Zahner. The maximum-entropy distribution of the future market price of a stock. *Operations Research*, 21:1200–1211, 1973.

[33] A. Darwiche and J. Pearl. Symbolic causal networks. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, 1994.

[34] A. Darwiche and J. Pearl. Symbolic causal networks for reasoning about actions and plans. In *Working notes of the AAAI Spring Symposium on Decision-Theoretic Planning*, 1994.

[35] A.P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society, Series A*, 41(1):1–31, 1979.

[36] A.P. Dawid. Present position and potential developments: Some personal views, statistical theory, the prequential approach. *Journal of the Royal Statistical Society, Series A*, 147(2):278–292, 1984.

[37] A.P. Dawid. Prequential analysis, stochastic complexity and Bayesian inference. In J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, editors, *Bayesian Statistics*, volume 4, pages 109–125. Oxford University Press, 1992. Proceedings of the Fourth Valencia Meeting.

[38] B. de Finetti. *Theory of Probability. A critical introductory treatment.* John Wiley & Sons, London, 1974.

[39] M.H. DeGroot. *Optimal statistical decisions.* McGraw-Hill, 1970.

[40] W. Derkse. *On Simplicity and Elegance: An Essay in Intellectual History.* PhD thesis, University of Amsterdam, 1993.

[41] D. Dubois and H. Prade. *Possibility theory: an approach to computerized processing of uncertainty.* Plenum, New York, 1988.

[42] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis.* John Wiley, 1973.

[43] C. Elkan. A perfect logic for reasoning about action. manuscript, 1989. University of Toronto.

[44] R.L. Ellis. Remarks on an alleged proof of the method of least squares, contained in a late number of the *edinburgh review.* In W. Walton, editor, *Mathematical and other Writings of R.L. Ellis*, pages 53–61. Cambridge University Press, Cambridge, 1863.

[45] C. Evans. Negation-as-failure as an approach to the hanks and mcdermott problem. In *Proceedings of the Second International Symposium on Artificial Intelligence*, Monterrey, Mexico, 1989.

[46] M. Feder. Maximum entropy as a special case of the minimum description length criterion. *IEEE Transactions on Information Theory*, 32(6):847–849, 1986.

[47] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, 1968. Third edition.

[48] R.A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Series A*, 222:309–368, 1925.

[49] K. Friedman and A. Shimony. Jaynes' maximum entropy prescription and probability theory. *Journal of Statistical Physics*, 9:265–269, 1971.

[50] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[51] H. Geffner. Causal theories for nonmonotonic reasoning. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 524–530, 1990.

[52] H. Geffner. Causality, constraints and the indirect effects of actions. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 555–560, 1997.

[53] M. Gelfond and V.L. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[54] M. Gelfond and V.L. Lifschitz. Representing actions in extended logic programming. In K.R. Apt, editor, *Proceedings of the Tenth Joint International Conference and Symposium on Logic Programming*, pages 559–573, 1992.

[55] S. Geman, E. Bienenstock, and R. Doversat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[56] M.L. Ginsberg, editor. *Readings in Nonmonotonic Reasoning*. Morgan Kaufmann, Los Altos, California, 1987.

[57] E. Giunchiglia and V. Lifschitz. Dependent fluents. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1964–1969, 1995.

[58] M. Goldszmidt, P. Morris, and J. Pearl. A maximum entropy approach to nonmonotonic reasoning. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 15(3):220–232, 1993.

[59] M. Goldszmidt and J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artificial Intelligence*, 84(1-2):57–112, 1996.

[60] P.D. Grünwald. Causal networks and nonmonotonic temporal reasoning. In *Proceedings Eighth Dutch Conference on Artificial Intelligence (NAIC-96)*, pages 157–166, Utrecht, 1996.

[61] P.D. Grünwald. A minimum description length approach to grammar inference. In G. Scheler S. Wermter, E. Riloff, editor, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, number 1040 in Springer Lecture Notes in Artificial Intelligence, pages 203–216. 1996.

[62] P.D. Grünwald. Causation and nonmonotonic temporal reasoning. In G. Brewka, C. Habel, and B. Nebel, editors, *KI-97: Advances in Artifical Intelligence*, number 1303 in Springer Lecture Notes in Artifical Intelligence, pages 159–170, 1997.

[63] P.D. Grünwald. Nonmonotonic temporal reasoning as a search for explanations. In *Proceedings NRAC '97 (Second IJCAI Workshop on Nonmonotonic Reasoning, Action and Change)*, pages 91–102, Nagoya, Japan, 1997.

[64] P.D. Grünwald. The sufficient cause principle and reasoning about action. Technical Report INS-R9709, CWI Amsterdam, 1997. Available at ftp://ftp.cwi.nl/pub/pdg/R9709.ps.Z.

[65] P.D. Grünwald, P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Minimum encoding approaches for predictive modeling. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98)*, 1998.

[66] J. Gustafsson and P. Doherty. Embracing occlusion in specifying the indirect effects of actions. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR' 96)*, 1996.

[67] S. Hanks and D. McDermott. Default reasoning, non-monotonic logics and the frame problem. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, pages 328–333, 1986.

[68] B.A. Haugh. Simple causal mimimizations for temporal persistence and projection. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 218–223, 1987.

[69] D. Haussler, M. Kearns, H.S. Seung, and N.Z. Tishby. Rigorous learning curve bounds from statistical mechanics. *Machine Learning*, 25:195–236, 1996.

[70] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, One Microsoft Way, Redmond, WA 98052, 1996.

[71] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[72] J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the theory of neural computation*. Lecture Notes of the Santa Fe Institute. Addison-Wesley, 1991.

[73] E.T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

[74] E.T. Jaynes. Information theory and statistical mechanics. In K.W. Ford, editor, *Statistical Physics (1962 Brandeis Lectures)*, pages 181–218, New York, 1963. W.A. Benjamin, Inc.

[75] E.T. Jaynes. Where do we stand on maximum entropy? In R.D. Levine and M. Tribus, editors, *The Maximum Entropy Formalism*, pages 15–118. MIT Press, Cambridge, MA, 1978.

[76] E.T. Jaynes. *Papers on Probability, Statistics and Statistical Physics*. Kluwer Academic Publishers, second edition, 1989.

[77] E.T. Jaynes. Probability theory: the logic of science. Available at ftp://bayes.wustl.edu/Jaynes.book/, 1996. Jaynes' forthcoming monumental treatise on probability theory as extended logic. This preliminary version (1996) is available via the web.

[78] H. Jeffreys. *Theory of Probability*. Oxford University Press, London, third edition, 1961.

[79] F.V. Jensen. *An Introduction to Bayesian Networks*. Springer, New York, 1996.

[80] G. N. Kartha. Soundness and completeness theorems for three formalizations of action. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 724–729, 1993.

[81] G.N. Kartha. Two counterexamples related to Baker's approach to the frame problem. *Artificial Intelligence*, 69:379–391, 1994.

[82] R.E. Kass and P.W. Voss. *Geometrical Foundations of Asymptotic Inference*. Wiley Interscience, 1997.

[83] H.A. Kautz. The logic of persistence. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, pages 401–405, 1986.

[84] M. Kearns, Y. Mansour, A.Y. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27:7–50, 1997.

[85] A.N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer-Verlag, 1933.

[86] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.

[87] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P.D. Grünwald. Bayesian and information-theoretic priors for Bayesian network parameters. In *Proceedings of the Tenth European Conference on Machine Learning (ECML-98)*, Chemnitz, Germany, 1998.

[88] L.G. Kraft. A device for quantizing, grouping and coding amplitude modulated pulses. Master's thesis, Dept. of Electrical Engineering, M.I.T., Cambridge, Mass., 1949.

[89] A. Krogh and G. Mitchison. Maximum entropy weighting of aligned sequences of proteins or DNA. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (ISMB '95)*, pages 215–221, 1995.

[90] P. Langley and S. Sage. Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI '94)*, pages 399–406, 1994.

[91] P.S. Laplace. *Essai philosophique sur les Probabilités*. H. Remy, Bruxelles, fifth edition, 1829.

[92] S.L. Lauritzen. *Graphical Models*. Oxford Science Publications, 1996.

[93] M. Li and P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, revised and expanded second edition, 1997.

[94] V.L. Lifschitz. Computing circumscription. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 121–127, 1985.

[95] V.L. Lifschitz. Formal theories of action. In M. L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*. Morgan Kaufmann Publishers, Los Altos, CA, 1987.

[96] V.L. Lifschitz. Pointwise circumscription. In M. L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*. Morgan Kaufmann Publishers, Los Altos, CA, 1987.

[97] V.L. Lifschitz. Frames in the space of situations. *Artificial Intelligence*, 46:365–376, 1990.

[98] V.L. Lifschitz and A. Rabinov. Miracles in formal theories of action. *Artificial Intelligence*, 38:225–237, 1989.

[99] F. Lin. Embracing causality in specifying the indirect effects of actions. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.

[100] F. Lin. Embracing causality in specifying the indeterminate effects of actions. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996.

[101] F. Lin and R. Reiter. State constraints revisited. *Journal of Logic and Computation*, 4(5):655–677, 1994.

[102] F. Lin and Y. Shoham. Provably correct theories of action (preliminary report). In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, 1991.

[103] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.

[104] N. McCain and H. Turner. Causal theories of action and change. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 1997.

[105] N. McCain and H. Turner. On relating causal theories to other formalisms. Unpublished Manuscript, 1997.

[106] J. McCarthy. Circumscription – a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1,2):27–39,171–172, 1980.

[107] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[108] D. McDermott. A critique of pure reason. *Computational Intelligence*, 3(3):151–237, 1987. Discussion: pages 161–237.

[109] D. McDermott and J. Doyle. Non-monotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[110] R.S. Michalski, J.G. Carbonell, and T.M. Mitchell. *Machine Learning, An Artificial Intelligence Approach*. Morgan Kaufmann, 1983.

[111] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, London, 1994.

[112] L. Morgenstern and L.A. Stein. Why things go wrong: A formal theory of causal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 5–23, 1988.

[113] I.J. Myung. Maximum entropy interpretation of decision bound and context models of categorization. *Journal of Mathematical Psychology*, 38:335–365, 1994.

[114] J.B. Paris. *The Uncertain Reasoner's Companion*. Number 39 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1994.

[115] J.B. Paris and A. Vencovská. A note on the inevitability of maximum entropy. *International Journal of Approximate Reasoning*, 4(3):183–224, 1990.

[116] J. Pearl. *Probabilistic Reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[117] J. Pearl. Graphical models, causality, and intervention. *Statistical Science*, 8(3):266–273, 1993.

[118] J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–709, 1995.

[119] J. Pearl. Causation, action, and counterfactuals. Technical Report R-223-U, University of California, Los Angeles, Computer Science Department, 1995. Presented at UNICOM Seminar, London, April 3-5, 1995.

[120] J. Pearl. Causation, action and counterfactuals. In Y. Shoham, editor, *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge (TARK-VI)*, pages 51–73. Morgan Kaufmann, 1996.

[121] D. Poole. What the lottery paradox tells us about default reasoning. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR' 89)*, pages 333–340, 1989.

[122] J. Quinlan and R. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[123] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings IEEE*, 77(2):257–286, 1989.

[124] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[125] B.D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[126] J. Rissanen. Modeling by the shortest data description. *Automatica*, 14:465–471, 1978.

[127] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society, series B*, 49:223–239, 1987. Discussion: pages 252–265.

[128] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Company, 1989.

[129] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, 1996.

[130] J.S. Rowlinson. Probability, information and entropy. *Nature*, 225(1196), 1970.

[131] E.J. Sandewall. Filter preferential entailment for the logic of action in almost continuous worlds. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, 1989.

[132] E.J. Sandewall. The range of applicability of nonmonotonic logics for the inertia problem. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, 1993.

[133] E.J. Sandewall. *Features and Fluents.* Oxford University Press, 1994.

[134] E.J. Sandewall. Assessments of ramification methods that use static domain constraints. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR' 96)*, 1996.

[135] E.J. Sandewall and Y. Shoham. Nonmonotonic temporal reasoning. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4. Oxford University Press, 1995.

[136] L.K. Schubert. Explanation closure, action closure and the Sandewall test suite for reasoning about change. *Journal of Logic and Computation*, 4(5):679–700, 1994.

[137] R.D. Shachter. Probabilistic inference and influence diagrams. *Operations Research*, 36(4):589–604, 1988.

[138] G. Shafer. *A mathematical theory of evidence.* Princeton University Press, 1976.

[139] M. Shanahan. *Solving the Frame Problem – A Mathematical Investigation of the Common Sense Law of Inertia.* MIT Press, Cambridge, MA, 1997.

[140] J.L. Shapiro and A. Prügel-Bennett. Maximum entropy analysis of genetic algorithm operators. In *Springer Lecture Notes in Computer Science (Proceedings AISB-95)*, volume 993, pages 14–24, 1995.

[141] A. Shimony. The status of the principle of maximum entropy. *Synthese*, 63:35–53, 1985.

[142] Y. Shoham. A semantical approach to nonmonotonic logics. In M.L. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 227–250. Morgan Kaufmann, Los Altos, California, 1987.

[143] Y. Shoham. Chronological ignorance: experiments in nonmonotonic temporal reasoning. *Artificial Intelligence*, 36:279–331, 1988.

[144] J.E. Shore and R.W. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory*, IT-26:26–37, 1980.

[145] R.J. Solomonoff. A formal theory of inductive inference, part 1 and part 2. *Information and Control*, 7:1–22, 224–254, 1964.

[146] M. Steijvers and P.D. Grünwald. A recurrent network that performs a context-sensitive prediction task. In *Proceedings Eighteenth Annual Conference of the Cognitive Science Society*, San Diego, CA, 1996.

[147] L.A. Stein and L. Morgenstern. Motivated action theory: A formal theory of causal reasoning. *Artificial Intelligence*, 71(1):1–42, 1994.

[148] A. Stolcke. *Bayesian Learning of Probabilistic Language Models.* PhD thesis, ICSI, Berkeley, 1994.

[149] J. Takeuchi and A.R. Barron. Asymptotically minimax regret by Bayes mixtures. In *Proceedings of the 1998 International Symposium on Information Theory (ISIT 98)*, 1998.

[150] M. Thielscher. The logic of dynamic systems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1956–1962, 1995.

[151] M. Thielscher. Ramification and causality. *Artificial Intelligence*, 89:317–364, 1997.

[152] Y. Tikochinsky, N.Z. Tishby, and R.D. Levine. Alternative approach to maximum entropy inference. *Physical Review A*, 30:2638–2644, 1984.

[153] Y. Tikochinsky, N.Z. Tishby, and R.D. Levine. Consistent inference of probabilities for reproducible experiments. *Physical Review Letters*, 52:1357–1360, 1984.

[154] H. Tirri, P. Kontkanen, and P. Myllymäki. Probabilistic instance-based learning. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96)*, pages 507–515, 1996.

[155] N.Z. Tishby. Statistical physics models of supervised learning. In D. Wolpert, editor, *The Mathematics of Generalization*, volume XX of *SFI Studies in the Sciences of Complexity*, pages 215–242. Addison-Wesley, 1995.

[156] D.S. Touretzky, J.F. Horty, and R.H. Thomason. A clash of intuitions: The current state of nonmonotonic multiple inheritance systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 476–482, 1989.

[157] J. Uffink. Can the maximum entropy principle be explained as a consistency requirement? *Studies in History and Philosophy of modern Physics*, 26B:223–261, 1995.

[158] J. Uffink. The constraint rule of the maximum entropy principle. *Studies in History and Philosophy of modern Physics*, 27:47–79, 1996.

[159] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, Berlin, 1982.

[160] P.M.B. Vitányi and M. Li. Minimum Description Length induction, Bayesianism, and Kolmogorov Complexity, 1997. Submitted to *IEEE Transactions on Information Theory*.

[161] C.S. Wallace and D.M. Boulton. An information measure for classification. *Computing Journal*, 11:185–195, 1968.

[162] C.S. Wallace and P.R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society, Series B*, 49:240–251, 1987. Discussion: pages 252–265.

[163] L. Wasserman. Bayesian model selection. *Journal of Mathematical Psychology*, 1998. To appear.

[164] S.S Wilks. *Mathematical statistics.* John Wiley, 1962.

[165] S. Wright. The method of path coefficients. *Annals of Mathematical Statistics*, 5:161–215, 1934.

[166] K. Yamanishi. On-line maximum likelihood prediction with respect to general loss functions. *Journal of Computer and System Sciences*, 55(1):105–118, 1997.

[167] K. Yamanishi. A decision-theoretic extension of stochastic complexity and its applications to learning. *IEEE Transactions on Information Theory*, 44(4):1424–1439, 1998.

List of Symbols

$\widetilde{L}$: minimum expected codelength, 96
L_{sc}: stochastic complexity, 36
ln: logarithm to base e, 12
log: logarithm to base 2, 12
LOSS: loss function, 32, 70
LOSS_{lg}: logarithmic loss, 44

$\mathcal{M}$: model class (parts I and II), 10, 17
$\mathcal{M}$: structure, interpretation or model of a logical theory (part III), 189, 210
$\mathbf{M}$: class of models for a logical theory, 212
$\mathcal{M}_{nb}$: Naive Bayes model class, 146
$\mathcal{M}_{me}$: maximum entropy model class, 55
$\mathbf{M}(\phi, t)$: set of P satisfying $E_P[\phi(X)] = t$, 53

$\mathbf{N}_0$: the nonnegative integers, 188, 224

$o(1)$: asymptotics notation, 40

P^*: 'true' distribution, 26, 34
$\mathcal{P}$: predictive distribution, 131
P: probability distribution, probabilistic model, 12, 13
Pa_i: set of parents of node i in a Bayesian network structure, 137
$\mathcal{P}_{av}$: evidence predictive distribution, 132
P_{av}: marginal distribution (Bayesian evidence), 39
$\mathcal{P}_{jef}$: Jeffrey's evidence predictive distribution, 136
P_{jef}: marginal distribution based on Jeffrey's prior, 134
$\mathcal{P}_{map}$: MAP predictive distribution, 132
P_{me}: maximum entropy distribution, 53
$\mathcal{P}_{sc}$: stochastic complexity predictive distribution, 133
P_{sc}: stochastic complexity distribution, 37

R: Rectangle, 114
$\mathbf{R}$: the real numbers, 12
$\mathcal{R}$: regret, 36

s: width between adjacent parameter values, 166
SCORE: score (predictive accuracy), 148

T: transpose, 53
T: logical theory / causal theory, 210
T_{C} : Two-point or first-order causal theory of our approach, 242, 252
T_{DF}: causal theory for the baby-and-the-table scenario, 228
$T_{\text{H-M}}$: naive theory for Yale Shooting Scenario, 190
T_{LIN} : logical theory in Lin's approach, 244
T_{MT} : domain description of McCain and Turner's approach, 242
T_{SW}: causal theory for basic two-switches scenario, 217
$T_{\text{SW,2}}$: causal theory for advanced switches scenario, 218
T_{WT}: causal theory for walking turkey scenario, 216
T_{YSP}: causal theory for Yale Shooting Scenario, 227
$\mathbf{T}$: set of names for time points, 249

$\mathbf{U}$: range of a function appearing in a maximum entropy distribution (part I), 53
$\mathbf{U}$: unobserved propositional variables (part III), 210
$\vec{u}, \vec{U}$: vector of clamped (given) (random) variables, 131
UNA: Uniqueness-of-Names-Axioms, 225

$\mathbf{V}$: observable propositional variables, 210
$\mathbf{V}_{\mathbf{E}}$: set of event–time pairs, 212

Index

Nederlandse Samenvatting

Centraal in dit proefschrift staat het *Beginsel van de Minimale Beschrijvingslengte* (in het Engels 'Minimum Description Length Principle'; vanaf nu 'MDL Principe' genoemd). Het MDL Principe stelt 'leren' gelijk aan 'comprimeren'. In zijn eenvoudigste vorm ziet het er als volgt uit:

MDL Principe De beste hypothese om een verzameling gegevens te verklaren is de hypothese H die de som van ...

- de beschrijvingslengte van de hypothese H en
- de beschrijvingslengte van de gegevens, wanneer de gegevens beschreven worden met behulp van hypothese H,

... minimaliseert.

In deze vorm zorgt het MDL Principe voor een afweging tussen *complexiteit* van de hypothese en de *fout* die de hypothese maakt op de gegevens. Het MDL Principe kan worden toegepast op alle vormen van *inductieve inferentie*. Met 'inductieve inferentie' wordt bedoeld het postuleren van algemene wetmatigheden op grond van een beperkte hoeveelheid gegevens. In dit proefschrift richten we ons met name op het gebruik van MDL in de statistiek en in het vakgebied genaamd 'machinaal leren' (machine learning). Dit is het deelgebied van de Kunstmatige Intelligentie dat zich bezighoudt met het leren door computers.

Het proefschrift bestaat uit drie delen. Deel I bevat een introductie tot het MDL Principe (hoofdstuk 1–3) en een bijdrage aan de theorievorming rond MDL (hoofdstuk 4–5). Hoofdstuk 1 geeft een algemene introductie en kan gelezen worden zonder kennis van statistiek of informatietheorie. Hoofdstuk 2 en 3 geven een voortgezette introductie, met de nadruk op drie zaken: ten eerste, de tamelijk ongebruikelijke interpretatie die MDL aan *kansverdelingen* toekent: volgens MDL dient men een empirisch bepaalde kansverdeling in eerste instantie te beschouwen als een code (preciezer, code-lengte functie). Ten tweede, het begrip 'stochastische complexiteit'. Dit is de centrale notie in de theorievorming rond MDL. Ten derde, het verband tussen MDL en het *Beginsel van de Maximale Entropie*, een principe dat oorspronkelijk bedoeld was als methode voor het 'redeneren met onzekerheid'. De theorie die wordt ontwikkeld in hoofdstuk 4 en 5 geeft een eerste aanzet tot het beantwoorden van de volgende vraag: hoe kan het dat simplistische modellen voor ingewikkelde processen vaak toch bruikbaar zijn? Het volgende geldt voor vrijwel alle praktische toepassingen van de statistiek: de uitkomst van de statistische analyse van de gegevens is een model dat in feite onjuist is, vaak zelfs een grove simplificatie. Toch worden 'simplistische' modellen die op deze manier verkregen zijn met succes gebruikt voor het voorspellen en classificeren van toekomstige gegevens. De centrale vraag in deel I van dit proefschrift is: wanneer kan men een 'simplistisch' model zonder problemen gebruiken? De hoofdconclusie luidt dat een simplistisch model op twee manieren gebruikt kan worden: een 'riskante' en een 'veilige'. Als het op de veilige manier gebruikt wordt, dan zal het simplistische model in het algemeen 'betrouwbaar' zijn. Dat wil zeggen dat het model zelf een correcte indruk geeft van de voorspellingsfout die men zal maken als men het gebruikt om toekomstige data mee te voorspellen – zelfs als het model een grove simplificatie is van het proces dat daadwerkelijk aan de gegevens ten grondslag ligt. Deze 'betrouwbaarheid' van een incorrect model kan in veel gevallen zelfs formeel bewezen worden (Hoofdstuk 5, Sectie 5.3, Stellingen 5.16-5.19).

Deel II (hoofdstuk 6 en 7) gaat over praktische toepassingen van het MDL Principe. Centrale vraag is hier: werkt het MDL Principe in de praktijk beter, even goed of minder goed dan andere statistische principes? De gevonden empirische verschillen kunnen voor een groot deel uit de bestaande theorie verklaard worden. Hoofdstuk 6 vergelijkt MDL met methoden uit de Bayesiaanse en klassieke statistiek. Hoofdstuk 7 vergelijkt MDL met het nauw verwante MML (Minimum Message Length): in tegenstelling tot wat vaak gedacht wordt, zijn er kleine theoretische verschillen tussen deze twee aanpakken. Deze leiden tot verschillend gedrag in praktische leerproblemen. De hoofdconclusie van Deel II is dat geavanceerde vormen van zowel MDL als Bayesiaanse inferentie vaak verrassend goed presteren wanneer slechts zeer weinig data gegeven is. MDL lijkt beter te presteren dan MML als weinig data gegeven is, zij het dat het verschil vrijwel verwaarloosbaar is.

Deel III gaat over een onderwerp dat slechts indirect aan MDL gerelateerd is: het ontwikkelen van een theorie over *gezond-verstand redeneren* ('common-sense reasoning') over *gebeurtenissen en veranderingen*. Dit soort theorieën wordt bestudeerd in het 'logicistische' paradigma van de Kunstmatige Intelligentie. In dit paradigma probeert men op wiskundige logica gebaseerde automatische redeneersystemen te ontwikkelen. Zulke systemen kunnen vervolgens worden toegepast in, bijvoorbeeld, robots. Hoofdstuk 8 geeft een inleidend overzicht van dit soort redeneersystemen. Centrale vragen in deel III zijn: hoe kunnen de sterke eigenschappen van bestaande redeneersystemen gecombineerd worden? Wat is de rol van *causaliteit* in dit soort redeneersystemen? En, hoe is de manier waarop deze systemen met onzekerheid omgaan gerelateerd aan kansrekening en de MDL-interpretatie van kansverdelingen? Ter beantwoording van deze vragen stellen wij (in hoofdstuk 9) een nieuw redeneersysteem voor, dat gebaseerd is op het *Beginsel van de Voldoende Oorzaak* ('sufficient cause principle'). Dit is onderdeel van een theorie over causaliteit die is ontwikkeld voor statistische toepassingen, dus buiten het vakgebied van de kunstmatige intelligentie. We bewijzen formeel (in hoofdstuk 10) dat ons redeneersysteem gezien kan worden als een generalisatie van enkele bestaande redeneersystemen (met name de systemen voorgesteld door (1) McCain en Turner, (2) Lin en (3) Baral, Gelfond en Provetti). We laten zien dat deze bestaande redeneersystemen vaak impliciet gebruik maken van het sufficient cause principe. We laten ook zien dat zij problematisch gedrag kunnen vertonen zodra zij ervan afwijken. Hoofdconclusie van deel III is dat het sufficient cause principe ons toestaat om een groot deel van zowel de successen als de mislukkingen van bestaande redeneersystemen te verklaren.

Het voorgestelde redeneersysteem maakt gebruik van *niet-monotone logica*. Deel III eindigt met een Epiloog waarin een formeel verband gelegd wordt tussen aan de ene kant deze niet-monotone logica en aan de andere kant probabilistische redeneermethoden. Er wordt een probabilistische semantiek voor eenvoudige vormen van niet-monotoon redeneren gegeven. Er wordt betoogd dat de 'kansen' die in deze semantiek optreden volgens het MDL Principe geïnterpreteerd dienen te worden.

Curriculum Vitae

Peter Daniel Grünwald was born May 13th, 1970 in Geldrop, the Netherlands. In 1988 he did his final exam Gymnasium β (Grammar School) at the Lorentz Lyceum in Eindhoven. He then moved to Amsterdam, where from 1988 until 1994 he studied Computer Science at the Free University of Amsterdam. During this time, he also obtained his 'propaedeutic exam' in Mathematics (1991), he worked as a teaching assistant at the Department of Artificial Intelligence (1991–1993) and he spent six months at the IRIT (Institut de Recherche en Informatique de Toulouse) and the Université Paul Sabatier in Toulouse, France (1993-1994). In 1994 he graduated 'cum laude' with specializations in Artificial Intelligence and Theoretical Computer Science and a minor in Psychology. 1994 he started working as a Ph.D. student ('Onderzoeker in Opleiding') at the CWI in Amsterdam. During this time he spent two months with the CoSCo group at the University of Helsinki, Finland (1997). The research he did as a Ph.D. student has resulted in the present thesis. He has been awarded an NWO TALENT–grant for a post-doc position at Stanford University, Palo Alto, California, starting October 1998.

Publications

[1] P.D. Grünwald, P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Minimum encoding approaches for predictive modeling. In *Proceedings of the Fourteenth International Conference on Uncertainty in Artificial Intelligence (UAI '98)*, Madison, WI, 1998.

[2] P.D. Grünwald. Causation and nonmonotonic temporal reasoning. In G. Brewka, C. Habel, and B. Nebel, editors, *KI-97: Advances in Artifical Intelligence*, number 1303 in Springer Lecture Notes in Artifical Intelligence, pages 159–170, 1997.

[3] P.D. Grünwald. A minimum description length approach to grammar inference. In G. Scheler S. Wermter, E. Riloff, editor, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, number 1040 in in Springer Lecture Notes in Artificial Intelligence, pages 203–216, 1996.

[4] P.D. Grünwald. Model selection based on minimum description length. Accepted for publication in *Journal of Mathematical Psychology*, special issue on Model Selection. To appear in 1999.

[5] P.D. Grünwald. Ramifications and sufficient causes. In *Working notes COMMON SENSE '98 (fourth symposium on logical formalizations of commonsense reasoning)*, 1998. Available at www.dcs.qmw.ac.uk/conferences/CS98/CS98Papers.html.

[6] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P.D. Grünwald. Predictive distributions and Bayesian networks. Accepted for publication in *Journal of Statistics and Computing*. To appear in 1999.

[7] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P.D. Grünwald. Comparing predictive inference methods for discrete domains. In *Proceedings AISTATS-97*, pages 311–318, Ft.Lauderdale, USA, 1997.

[8] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P.D. Grünwald. Bayesian and information-theoretic priors for Bayesian network parameters. In *Proceedings of the Tenth European Conference on Machine Learning (ECML-98)*, Chemnitz, Germany, 1998.

[9] M. Steijvers and P.D. Grünwald. A recurrent network that performs a context-sensitive prediction task. In *Proceedings Eighteenth Annual Conference of the Cognitive Science Society*, San Diego, CA, 1996.

Only publications that have appeared or are to appear in journals, books or refereed international conferences have been listed. The material in this thesis is partly based on some of these publications. Specifically, parts of chapters 1 and 2 are new and parts are based on [4]. The material in chapters 3, 4 and 5 is new and as yet unpublished. Chapter 6 is a combination of [6], [7] and [8]. Chapter 7 is based on [1]. Part III of the thesis is partially based on [2] and [5]. The material in publications [3] and [9] does not appear in this thesis.

Titles in the ILLC Dissertation Series:

ILLC DS-1993-01: **Paul Dekker**
Transsentential Meditations; Ups and downs in dynamic semantics

ILLC DS-1993-02: **Harry Buhrman**
Resource Bounded Reductions

ILLC DS-1993-03: **Rineke Verbrugge**
Efficient Metamathematics

ILLC DS-1993-04: **Maarten de Rijke**
Extending Modal Logic

ILLC DS-1993-05: **Herman Hendriks**
Studied Flexibility

ILLC DS-1993-06: **John Tromp**
Aspects of Algorithms and Complexity

ILLC DS-1994-01: **Harold Schellinx**
The Noble Art of Linear Decorating

ILLC DS-1994-02: **Jan Willem Cornelis Koorn**
Generating Uniform User-Interfaces for Interactive Programming Environments

ILLC DS-1994-03: **Nicoline Johanna Drost**
Process Theory and Equation Solving

ILLC DS-1994-04: **Jan Jaspars**
Calculi for Constructive Communication, a Study of the Dynamics of Partial States

ILLC DS-1994-05: **Arie van Deursen**
Executable Language Definitions, Case Studies and Origin Tracking Techniques

ILLC DS-1994-06: **Domenico Zambella**
Chapters on Bounded Arithmetic & on Provability Logic

ILLC DS-1994-07: **V. Yu. Shavrukov**
Adventures in Diagonalizable Algebras

ILLC DS-1994-08: **Makoto Kanazawa**
Learnable Classes of Categorial Grammars

ILLC DS-1994-09: **Wan Fokkink**
Clocks, Trees and Stars in Process Theory

ILLC DS-1994-10: **Zhisheng Huang**
Logics for Agents with Bounded Rationality

ILLC DS-1995-01: **Jacob Brunekreef**
On Modular Algebraic Protocol Specification

ILLC DS-1995-02: **Andreja Prijatelj**
Investigating Bounded Contraction

ILLC DS-1995-03: **Maarten Marx**
Algebraic Relativization and Arrow Logic

ILLC DS-1995-04: **Dejuan Wang**
Study on the Formal Semantics of Pictures

ILLC DS-1995-05: **Frank Tip**
Generation of Program Analysis Tools

ILLC DS-1995-06: **Jos van Wamel**
Verification Techniques for Elementary Data Types and Retransmission Protocols

ILLC DS-1995-07: **Sandro Etalle**
Transformation and Analysis of (Constraint) Logic Programs

ILLC DS-1995-08: **Natasha Kurtonina**
Frames and Labels. A Modal Analysis of Categorial Inference

ILLC DS-1995-09: **G.J. Veltink**
Tools for PSF

ILLC DS-1995-10: **Giovanna Cepparello**
Studies in Dynamic Logic

ILLC DS-1995-11: **W.P.M. Meyer Viol**
Instantial Logic. An Investigation into Reasoning with Instances

ILLC DS-1995-12: **Szabolcs Mikulás**
Taming Logics

ILLC DS-1995-13: **Marianne Kalsbeek**
Meta-Logics for Logic Programming

ILLC DS-1995-14: **Rens Bod**
Enriching Linguistics with Statistics: Performance Models of Natural Language

ILLC DS-1995-15: **Marten Trautwein**
Computational Pitfalls in Tractable Grammatical Formalisms

ILLC DS-1995-16: **Sophie Fischer**
The Solution Sets of Local Search Problems

ILLC DS-1995-17: **Michiel Leezenberg**
Contexts of Metaphor

ILLC DS-1995-18: **Willem Groeneveld**
Logical Investigations into Dynamic Semantics

ILLC DS-1995-19: **Erik Aarts**
Investigations in Logic, Language and Computation

ILLC DS-1995-20: **Natasha Alechina**
Modal Quantifiers

ILLC DS-1996-01: **Lex Hendriks**
Computations in Propositional Logic

ILLC DS-1996-02: **Angelo Montanari**
Metric and Layered Temporal Logic for Time Granularity

ILLC DS-1996-03: **Martin H. van den Berg**
Some Aspects of the Internal Structure of Discourse: the Dynamics of Nominal Anaphora

ILLC DS-1996-04: **Jeroen Bruggeman**
Formalizing Organizational Ecology

ILLC DS-1997-01: **Ronald Cramer**
Modular Design of Secure yet Practical Cryptographic Protocols

ILLC DS-1997-02: **Nataša Rakić**
Common Sense Time and Special Relativity

ILLC DS-1997-03: **Arthur Nieuwendijk**
On Logic. Inquiries into the Justification of Deduction

ILLC DS-1997-04: **Atocha Aliseda-LLera**
Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence

ILLC DS-1997-05: **Harry Stein**
The Fiber and the Fabric: An Inquiry into Wittgenstein's Views on Rule-Following and Linguistic Normativity

ILLC DS-1997-06: **Leonie Bosveld - de Smet**
On Mass and Plural Quantification. The Case of French 'des'/'du'-NP's.

ILLC DS-1998-01: **Sebastiaan A. Terwijn**
Computability and Measure

ILLC DS-1998-02: **Sjoerd D. Zwart**
Approach to the Truth: Verisimilitude and Truthlikeness

ILLC DS-1998-03: **Peter Grünwald**
The Minimum Description Length Principle and Reasoning under Uncertainty