Universal Modeling: Introduction to 'Modern' MDL

Peter Grünwald
CWI and EURANDOM
www.grunwald.nl

Lectures for the Machine Learning Summer School, Max-Planck-Institute for biological Cybernetics, Tübingen, Germany, August 4-16 2003
Extended and Revised September 24th, 2003

Part 0: What is all this about ??

Minimum Description Length Principle

Rissanen 1978, 1987, 1996,
Barron, Rissanen and Yu 1998

- 'MDL' is a method for inductive inference,
- in particular developed and suited for model selection problems

Minimum Description Length Principle

- MDL is based on the correspondence between 'regularity' and 'compression':
  - The more you are able to compress a sequence of data, the more regularity you have detected in the data
  - Example:

    00100100100100100100100100....001

    01011011100100111010001010 1....010

Minimum Description Length Principle

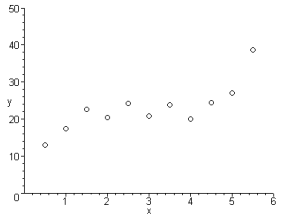
- MDL is based on the correspondence between 'regularity' and 'compression':
  - The more you are able to compress a sequence of data, the more regularity you have detected in the data…
  - …and thus the more you have learned from the data:
    - 'inductive inference' as trying to find regularities in data (and using those to make predictions of future data)

Model Selection

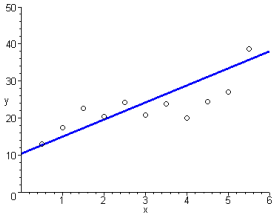
Given data $x^n = x_1, \ldots, x_n$ and 'models' $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \ldots$,
which model *best explains* the data ?

- Need to take into account
  - Error (minus Goodness-of-fit)
  - Complexity of models
- Examples
  - Variable (order) selection in regression
  - Selection of order in (hidden) Markov Models

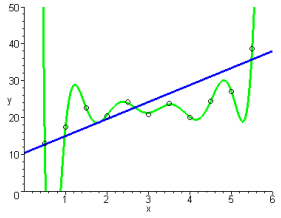
Example: Regression



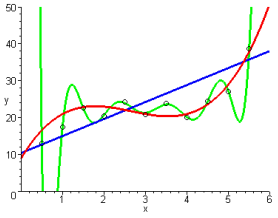
Example: Regression



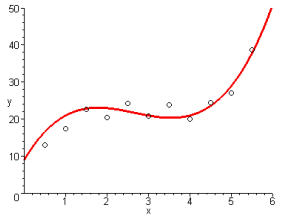
Example: Regression



Example: Regression



Example: Regression



Modern MDL!

Kinds of MDL

- Algorithmic, 'ideal' MDL (Li and Vitányi '97)
- MML (Wallace '68 (!), '87)
- 2-part code MDL (Rissanen '78, '83)
- Universal model based MDL (Rissanen '96, Barron, Rissanen, Yu '98, Grünwald '03)

Five MDL Lectures

Part I: basics
- information-theoretic preliminaries: Probabilities and code lengths, universal code/model

Part II: basic MDL model selection
- Four interpretations; How to use it in practice

Part III: the general MDL Principle;
- extensions/difficulties

Part IV: classification
- promises and problems

Part V: MDL and the others
- does it work in practice?
- justification/comparison to other approaches

Part I: Basics

Part I: Overview

- Probability and Code Length
- Universal Models
- MDL Model Selection

Codes

$\mathcal{X}$ (countable) 'data alphabet'

A (uniquely decodable) code C is a one-to-one map from $\mathcal{X}$ to $\{0,1\}^+ = \cup_{n \geq 1} \{0,1\}^n$

$L_C(x)$ denotes the length (in bits) needed to describe x.

Example 1: uniform code

- Let $\mathcal{X} = \{a, b, c, d\}$
- One possible code for $\mathcal{X}$ is given by

$$C(a) = 00, C(b) = 01, C(c) = 10, C(d) = 11$$

- Then for all x, $L_C(x) = 2$.
- But of course infinitely many other (not necessarily uniform-length) codes are possible as well!

Code Length & Probability

- Let P be a probability distribution. Since $\sum_x P(x) \leq 1$ only few x can have 'large' probability
- Let C be a code for $\{0,1\}^m$. . Since the fraction of sequences that can be compressed by more than k bits is less than $2^{m-k}/2^m = 2^{-k}$, only very few symbols can have small code length.
- This suggests an **analogy!**

Code Lengths 'are' probabilities…

- Let C be a (uniquely decodable) code over countable set $\mathcal{X}$. Then there exists a (possibly defective) probability distribution P_C such that
  $$\text{for all } x : L_C(x) = -\log P_C(x)$$
- P_C is a 'proper' probability distribution iff the code C is 'complete'.
  (follows from Kraft-McMillan inequality)

…and probabilities 'are' code lengths!

- Let P be a probability distribution over countable set $\mathcal{X}$. Then there exists a code C_P for $\mathcal{X}$ such that
  $$\text{for all } x : L_{C_P}(x) = \lceil -\log P(x) \rceil$$

The Most Important Slide!

There is a 1-1 correspondence between probability distributions and code length functions, such that small probabilities correspond to large code lengths and vice versa:

$$\text{for all } x^n \in \mathcal{X}^n : L(x^n) = -\log P(x^n)$$

The Most Important Slide!

There is a 1-1 correspondence between probability distributions and code length functions, such that small probabilities correspond to large code lengths and vice versa:

$$\text{for all } x^n \in \mathcal{X}^n : L(x^n) = -\log P(x^n)$$

Note: data alphabet is now a sample
$$x^n \equiv (x_1, \ldots, x_n) \in \mathcal{X}^n$$

The Most Important Slide!

There is a 1-1 correspondence between probability distributions and code length functions, such that small probabilities correspond to large code lengths and vice versa:

$$\text{for all } x^n \in \mathcal{X}^n : L(x^n) = -\log P(x^n)$$

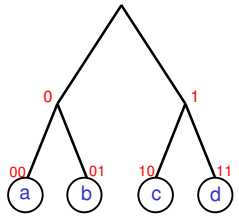
Example: P is 1st 0rder Markov Chain – if P fits data well (regularities in data are well-captured by P), the code based on P compresses much.

Example 1: uniform code/distr.

- Let $\mathcal{X} = \{a, b, c, d\}$
- Let $P(a) = P(b) = P(c) = P(d) = \frac{1}{4}$
- One of the codes corresponding to P is
  $C(a) = 00, C(b) = 01, C(c) = 10, C(d) = 11$
- We have for all x : $-\log P(x) = L_C(x) = 2$

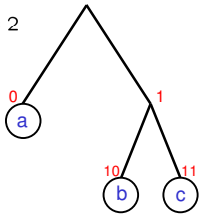
Prefix codes → distributions

Every prefix code can be represented by a binary tree.
For example:



Prefix codes → distributions

Non-uniform example:
$L_C(a) = 1$
$L_C(b) = L_C(c) = 2$



$P(a) = \frac{1}{2}$
$P(b) = P(c) = \frac{1}{4}$

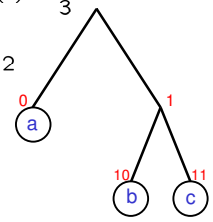
For all x : $-\log P(x) = L_C(x)$

General Recipe (Kraft)

- Codes → Distributions:
  - For every uniquely decodeable code, there is a prefix code with the same length function (McMillan)
  - Let y be a leaf node of some arbitrary binary tree, and let $D(y)$ be the length of path between y and root node. Then
$$\sum_{y: y \text{ is leaf node}} 2^{-D(y)} \leq 1$$
  - For every prefix code there is a binary tree with leaf nodes representing the encoded symbols x such that for all x:
$$L(x) = D(x)$$
- Distributions → Codes: slightly more complicated

Example 3: distributions → codes

$P(a) = P(b) = P(c) = \frac{1}{3}$
$L_C(a) = 1$
$L_C(b) = L_C(c) = 2$



For all x, difference $|-\log P(x) - L_C(x)| < 1$

Example 3: distributions → codes

- Now suppose we extend $P(a) = P(b) = P(c) = 1/3$ to n outcomes by independence
- For all x^n , we have
$$-\log P(x^n) = n \log 3$$
- It seems that, for some sequences, e.g.,
$x^n = cccccc....cccc$, we have
$$L_C(x^n) = 2n$$
which is much larger!?!?
- …but it's not: rather than concatenating codewords, we should directly assign probabilities to x^n

The Most Important Slide!

There is a 1-1 correspondence between probability distributions and code length functions, such that small probabilities correspond to large code lengths and vice versa:

$$\text{for all } x^n \in \mathcal{X}^n : L(x^n) = -\log P(x^n)$$

Remarks

- In this correspondence, probability distributions (mass functions) are treated as mathematical objects and *nothing else*.
- Extend correspondence to continuous sample space through discretization; P may stand for *density*
- Distributions and codes over sequences of outcomes: still max. 1 bit round-off error
- Neglect difference and *identify* code length functions and probability mass functions

Part I: Overview

- Probability and Code Length
- Universal Models
- MDL Model Selection

Universal Codes

- $\mathcal{L}$: set of code (length function)s available to encode data $x^n = x_1, \ldots, x_n$
- Suppose we think that one of the code(length function)s in $\mathcal{L}$ allows for substantial compression of x^n
- GOAL: encode x^n using minimum number of bits!

Universal Codes

- Simply encoding x^n using the $\hat{L} \in \mathcal{L}$ that minimizes code length $\hat{L}(x^n) = \inf_{L \in \mathcal{L}} L(x^n)$ does not work (encoding cannot be decoded)
- But there exist codes $L_{\mathcal{L}}$ which, for any sequence x^n are 'almost' as good as $\inf_{L \in \mathcal{L}} L(x^n)$
- These are called 'universal codes' for $\mathcal{L}$

Universal Codes

- Example: $\mathcal{L}$ finite
- There exists a 2-part code $L_{\mathcal{L}}$ such that for some constant K, for all n, x^n, all $L \in \mathcal{L}$:

$$L_{\mathcal{L}}(x^n) \leq L(x^n) + K$$

- In particular,

$$L_{\mathcal{L}}(x^n) \leq \inf_{L \in \mathcal{L}} L(x^n) + K$$

- Note that K does not depend on n, while typically, $L(x^n)$ grows linearly in n

Universal Models

- Let $\mathcal{M}$ be a probabilistic model, i.e. a family (set) of probability distributions
- Assume $\mathcal{M}$ finite: $\mathcal{M} = \{P(\cdot|\theta_1), \ldots, P(\cdot|\theta_M)\}$
- There exists a code $L_{\mathcal{M}}$ such that for all n, x^n, θ :

$$L_{\mathcal{M}}(x^n) \leq -\log P(x^n|\theta) + K$$

- Hence, exists distribution $P_{\mathcal{M}}$ such that

$$-\log P_{\mathcal{M}}(x^n) \leq -\log P(x^n|\theta) + K$$

- i.e. $P_{\mathcal{M}}(x^n) \geq K' \cdot P(x^n|\theta)$
- $P_{\mathcal{M}}$ is a 'universal model' (distribution) for $\mathcal{M}$

Terminology

- Statistics:
  - Model = family of distributions
- Information theory:
  - Model = single distribution
  - Model class = family of distributions
- Universal model is a single distribution acting as a representative of/defined relative to a set of distributions

Bayesian Mixtures are universal models

- Let W be a prior over $\mathcal{M}$. The Bayesian marginal likelihood P_{Bayes} is defined as:

$$P_{\mathrm{Bayes}}(x^n|\mathcal{M}) = \sum_{j=1}^{M} P(x^n|\theta_j)W(\theta_j)$$

Bayesian Mixtures are universal models

- Let W be a prior over $\mathcal{M}$. The Bayesian marginal likelihood is defined as:

$$P_{\mathrm{Bayes}}(x^n|\mathcal{M}) = \sum_{j=1}^{M} P(x^n|\theta_j)W(\theta_j)$$

- This is a universal model, since

$$\text{For all } n, x^n, \theta\text{: } -\log P_{\mathrm{Bayes}}(x^n|\mathcal{M}) = -\log \sum_{j=1}^{M} P(x^n|\theta_j)W(\theta_j)$$
$$\leq -\log P(x^n|\theta) - \log W(\theta)$$

2-part MDL code is a universal model (code)

- The ML (*maximum likelihood)* distribution $\hat{\theta}(x^n)$ is the θ achieving $\inf_{P(\cdot|\theta)\in\mathcal{M}}\{-\log P(x^n|\theta)\}$
- Code x^n by first coding $\hat{\theta}(x^n)$, then coding x^n 'with the help of' $\hat{\theta}(x^n)$:

$$L_{2p}(x^n) = -\log W(\hat{\theta}(x^n)) - \log P(x^n|\hat{\theta}(x^n))$$

2-part vs. Bayes universal models

- Bayes' mixture typically 'better' universal model in that it assigns larger probability (shorter code length) to outcomes.
- What does 'better' really mean?
- What *prior* leads to short code lengths?

Optimal Universal Model

Look for P^* such that regret

$$-\log P^*(x^n) - [-\log P(x^n|\hat{\theta}(x^n))]$$

is small *no matter what* x^n *are;* i.e. look for

$$\inf_{P^*} \sup_{x^n \in \mathcal{X}^n} \{-\log P^*(x^n) - [-\log P(x^n|\hat{\theta}(x^n))]\}$$

Optimal Universal Model - II

$$\inf_{P^*} \sup_{x^n \in \mathcal{X}^n} \{-\log P^*(x^n) - [-\log P(x^n|\hat{\theta}(x^n))]\}$$

is achieved for Normalized Maximum Likelihood (NML) distribution (Shtarkov 1987):

$$P_{\mathrm{NML}}(x^n|\mathcal{M}) = \frac{P(x^n|\hat{\theta}(x^n))}{\sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}(y^n))}$$

Summary: Universal Codes

- Given set of candidate codes $\mathcal{L}$
- Universal code is a code satisfying for all n, x^n :
  $L_{\mathcal{L}}(x^n) \leq \inf_{L \in \mathcal{L}} L(x^n) + \text{small regret}.$
  - If $\mathcal{L}$ finite, then there exist 2-part codes which satisfy for all n, x^n:
    $L_{\mathcal{L}}(x^n) = \inf_{L \in \mathcal{L}} L(x^n) + \log |\mathcal{L}|,$
  - if $\mathcal{L}$ infinite, then regret typically grows with n
    - Often, logarithmic regret is achieveable

Summary: Universal Models

- Given set of candidate distributions $\mathcal{M}$
- Universal model is distribution with for all n, x^n :
  $-\log P(x^n|\mathcal{M}) \leq \inf_{P(\cdot|\theta) \in \mathcal{M}} -\log P(x^n|\theta) + \text{small regret}.$

  That is,

  $-\log P(x^n|\mathcal{M}) \leq -\log P(x^n|\hat{\theta}(x^n)) + \text{small regret}.$
  - If $\mathcal{M}$ finite, then finite regret achieveable
  - if $\mathcal{M}$ infinite, then regret typically grows with n
    - For parametric families logarithmic regret is achieveable

Summary: Optimal Universal Model

- Minimax (optimal in the worst-case) regret is achieved by the Normalized Maximum Likelihood Distribution :

$$P_{\mathrm{NML}}(x^n|\mathcal{M}) = \frac{P(x^n|\hat{\theta}(x^n))}{\sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}(y^n))}$$

Part II: basic MDL model selection

Part II: Overview

1. MDL model selection for parametric models
2. Four interpretations:
   - Compression Interpretation
   - Counting/Geometric Interpretation
   - Bayesian Interpretation
   - Predictive Interpretation
3. How to use it in practice

MDL Model Selection

- Suppose we are given data $x^n = x_1, \ldots, x_n$
- We want to select between models $\mathcal{M}_1$ and $\mathcal{M}_2$, as explanations for the data. MDL tells us to pick the $\mathcal{M}_i$ for which the associated optimal universal model $P_{\mathrm{NML}}(\cdot|\mathcal{M}_i)$ assigns the largest probability to the data:

$$\mathcal{M}_{mdl} = \arg\sup_{\mathcal{M}_i} P_{\mathrm{NML}}(x^n|\mathcal{M}_i) = \arg\inf_{\mathcal{M}_i} -\log P_{\mathrm{NML}}(x^n|\mathcal{M}_i)$$

MDL Model Selection

Select $\mathcal{M}_i$ minimizing $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}_i)$, i.e. minimizing

$$-\log P(x^n|\hat{\theta}_i(x^n)) + \log \sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}_i(y^n))$$

error (= minus fit) term **complexity** term ($\leq \log M$)

(this is just 'MDL model selection between two simple models'; it is not 'the MDL Principle')

Four Interpretations

- Compression interpretation
- Counting/Geometric interpretation
- Bayesian interpretation
- Predictive interpretation

Compression Interpretation

Select $\mathcal{M}_i$ minimizing $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}_i)$, i.e. minimizing

$$-\log P(x^n|\hat{\theta}_i(x^n)) + \log \sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}_i(y^n))$$

error (= minus fit) term **complexity** term ($\leq \log M$)

- select model that compresses data most, *treating all distributions within model on equal footing*;
- selected model detects most (non-spurious) regularity in data

Counting Interpretation of MDL

Select $\mathcal{M}_i$ minimizing $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}_i)$, i.e. minimizing

$$-\log P(x^n|\hat{\theta}_i(x^n)) + \log \sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}_i(y^n))$$

error (= minus fit) term **complexity** term ($\leq \log M$)

Something like 'total fit' model gives to data

Log 'effective' number of distributions

Counting Interpretation of MDL

$$\sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}(y^n)) = \sum_{\theta : P(\cdot|\theta) \in \mathcal{M}} \sum_{y_n \in \mathcal{X}^n : \hat{\theta}(y^n) = \theta} P(y^n|\theta) =$$

$$\sum_{\theta} P(\{y^n \in \mathcal{X}^n : \hat{\theta}(y^n) = \theta\}|\theta) = \sum_{\theta} [1 - P(\{x^n \in \mathcal{X}^n : \hat{\theta}(x^n) \neq \theta\}|\theta)] =$$

$$M - \sum_{\theta} P(\hat{\theta} \neq \theta|\theta)$$

number of distributions

total amount of confusion

Counting Interpretation of MDL

Select $\mathcal{M}_i$ minimizing $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}_i)$, i.e. minimizing

$$-\log P(x^n|\hat{\theta}_i(x^n)) + \log \textstyle\sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}_i(y^n))$$

error (= minus fit) term — **complexity** term ($\leq \log M$)

Something like 'total fit' model gives to data

Log number of 'distinguishable' distributions

Parametric Model Classes

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$

$$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log \int \sqrt{\det I(\theta)}d\theta + o(1)$$

- Here:

k	Number of free parameters in $\mathcal{M}$
$I(\theta)$	Fisher information matrix at θ
$o(1)$	Goes to 0 as $n \to \infty$

Geometric Interpretation of MDL

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$

$$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log \int \sqrt{\det I(\theta)}d\theta + o(1)$$

- Compare **BIC** (Schwartz '78), old **'MDL Criterion'** (Rissanen '78): select $\mathcal{M}$ minimizing:

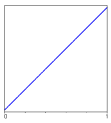
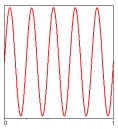
$$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi}$$

Geometric Interpretation of MDL

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$

$$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log \int \sqrt{\det I(\theta)}d\theta + o(1)$$

- Complexity is not only related to number of parameters! (Balasubramanian, Myung, Pitt '00)

Bernoulli vs. Crazy Bernoulli embedded in First-Order Markov

Geometric Interpretation of MDL

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$

$$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log \int \sqrt{\det I(\theta)}d\theta + o(1)$$

nr of parameters — Curvature term

Error Term — **Complexity** — **Terms**

Geometric Interpretation of MDL

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$

$$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log \int \sqrt{\det I(\theta)}d\theta + o(1)$$

$$=$$

$$\log \textstyle\sum_{y^n \in \mathcal{X}^n} P(y^n|\hat{\theta}(y^n))$$

$$=$$

Log ratio of total nr of ***distinguishable*** ('essentially different') distributions in $\mathcal{M}$ to distinguishable distributions in $\mathcal{M}$ close to $\hat{\theta}(x^n)$ (Balasubramanian '98)

Geometric Interpretation of MDL

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$
$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log\int\sqrt{\det I(\theta)}d\theta + o(1)$
$=$
$\log\sum_{y^n\in\mathcal{X}^n} P(y^n|\hat{\theta}(y^n))$
$=$

MDL truly is a '***normalized*** **Maximum Likelihood Principle'!**

Log ratio of total nr of ***distinguishable*** (`essentially different') distributions in $\mathcal{M}$ to distinguishable distributions in $\mathcal{M}$ close to $\hat{\theta}(x^n)$ (Balasubramanian '98)

Bayesian Model Selection vs. MDL

- Recall the Bayesian universal model
$$P_{\mathsf{Bayes}}(x^n|\mathcal{M}) = \int_{\theta:P(\cdot|\theta)\in\mathcal{M}} P(x^n|\theta)w(\theta)d\theta$$
- Bayesian model selection between $\mathcal{M}_1$ and $\mathcal{M}_2$ tells us to select the $\mathcal{M}_i$ maximizing
$$P(\mathcal{M}_i \mid x^n) = \frac{P_{\mathsf{Bayes}}(x^n \mid \mathcal{M}_i)W(i)}{\sum_j P_{\mathsf{Bayes}}(x^n \mid \mathcal{M}_j)W(j)}$$
- with uniform prior W this is the $\mathcal{M}_i$ maximizing
$$P_{\mathsf{Bayes}}(x^n \mid \mathcal{M}_i)$$

Bayesian Model Selection vs. MDL

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$
$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log\int\sqrt{\det I(\theta)}d\theta + o(1)$
- Under regularity conditions: $-\log P_{\mathrm{Bayes}}(x^n|\mathcal{M}) \approx$
$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} - \log w(\hat{\theta}) + \log\sqrt{\det I(\hat{\theta})} + o(1)$
- Always within O(1) ; hence, for large enough n, Bayes and MDL select the same model

Bayesian Model Selection vs. MDL

- Under regularity conditions: $-\log P_{\mathrm{NML}}(x^n|\mathcal{M}) =$
$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} + \log\int\sqrt{\det I(\theta)}d\theta + o(1))$
- Under regularity conditions: $-\log P_{\mathrm{Bayes}}(x^n|\mathcal{M}) \approx$
$-\log P(x^n|\hat{\theta}(x^n)) + \frac{k}{2}\log\frac{n}{2\pi} - \log w(\hat{\theta}) + \log\sqrt{\det I(\hat{\theta})} + o(1)$
- If we take *Jeffreys-Bernardo prior*,
$w(\theta) = \sqrt{\det I(\theta)}/\int_\theta\sqrt{\det I(\theta)}d\theta$
within o(1): Bayes and NML become *indistinguishable*

Bayes and MDL, remarks

- Jeffreys' prior was proposed as a 'non-informative Bayesian prior' by Jeffreys in 1939
- Jeffreys' prior is uniform prior *not* on parameter space but on the space of distributions with the 'natural metric' that measures distances between distributions by how distinguishable they are.

Four Interpretations

- Compression interpretation
- Counting/Geometric interpretation
- Bayesian interpretation
- Predictive interpretation

Predictive Interpretation

- Interpret $-\log P(x)$ as 'loss' incurred when predicting using P while actual outcome was x

$$\mathrm{Loss}(x, P) \equiv -\log P(x)$$

- Bayesian marginal likelihood can be rewritten as accumulated log-loss prediction error

$$-\log P_{\mathrm{Bayes}}(x^n) = -\log \prod_{i=1}^{n} \frac{P_{\mathrm{Bayes}}(x^i)}{P_{\mathrm{Bayes}}(x^{i-1})} =$$

$$\sum_{i=1}^{n} -\log P_{\mathrm{Bayes}}(x_i | x_1, \ldots, x_{i-1}) = \sum_{i=1}^{n} \mathrm{Loss}(x_i, P_{\mathrm{Bayes}}(\cdot | x^{i-1}))$$

- Here $P_{\mathrm{Bayes}}(\cdot | x_1, \ldots, x_{i-1})$ is the Bayesian predictive distribution (posterior mixture)

Predictive Interpretation, II

- Bayesian predictive distribution given by

$$P_{\mathrm{Bayes}}(x_i \mid x^{i-1}) = \frac{\int P(x^i|\theta) w(\theta) d\theta}{\int P(x^{i-1}|\theta) w(\theta) d\theta} = \int P(x_i|\theta) w(\theta | x_1, \ldots, x_{i-1}) d\theta$$

- For large n, Bayesian posterior concentrates very sharply around ML distribution
- Therefore, Bayes predictive distribution resembles ML distribution more and more

Predictive Interpretation, II

- Idea (Dawid/Rissanen): for large n, Bayesian predictive distribution resembles ML distribution more and more; therefore, may try to approximate $P_{\mathrm{Bayes}}(\cdot | x_1, \ldots, x_{i-1})$ by

$$P(\cdot | \hat{\theta}(x_1, \ldots, x_{i-1}))$$

or more generally by

$$P(\cdot | \tilde{\theta}(x_1, \ldots, x_{i-1}))$$

for any 'likelihood-based estimator' $\tilde{\theta}$

Predictive Interpretation, III

- It turns out that (under regularity conditions)

$$-\sum_{i=1}^{n} \log P(x_i | \hat{\theta}(x^{i-1})) = -\log P(x^n | \hat{\theta}(x^n)) + \tfrac{k}{2} \log n + O(1)$$

- Hence, 'predictive code' is a universal model (the fourth kind we encounter!)
- MDL model selection picks the model $\mathcal{M}$ such that sequential prediction of the future given the past within the observed data leads to lowest accumulated sequential prediction error.

Predictive Interpretation, IV

- MDL can be cast in terms of prequential validation (Dawid '84)
- similar to cross-validation
- essential difference: in MDL/prequential validation each outcome predicted *exactly once*

Part II: Overview

1. MDL model selection for parametric models
2. Four interpretations:
   - Compression Interpretation
   - Counting/Geometric Interpretation
   - Bayesian Interpretation
   - Predictive Interpretation
3. How to use it in practice

How to use MDL in practical Model Selection Problems

In order of preference:

1. Try o(1)-universal models: NML distributions or non-informative Bayesian mixtures *or*
2. Use predictive MDL
   - with sequential Bayes-MAP estimates
3. Use two-part code MDL with 'good' codes *or*
4. Use asymptotic expansion (k/2 log n +…) (or maybe not -- be **super-careful**!) *or*
5. *Use another O(1)-universal model*

Computational Issues

- we **NEVER NEVER** have to do any real coding!
- Code length of Bayesian universal model can be approximated with Markov Chain Monte Carlo
- $-\log P(x^n \mid \hat{\theta}(x^n))$ and $\hat{\theta}(x^n)$easily computable for exponential families; otherwise, may find local maximum of likelihood function (e.g. with EM)
-

Computational Issues

- we **NEVER NEVER** have to do any real coding!
- Code length of Bayesian universal model can be approximated with Markov Chain Monte Carlo
- $-\log P(x^n \mid \hat{\theta}(x^n))$ and $\hat{\theta}(x^n)$easily computable for exponential families; otherwise, may find local maximum of likelihood function (e.g. with EM)
- Problematic aspect: now complexity term should really be recomputed as well!
  - if $\hat{\theta}(x^n)$ represents a local maximum, then $\log \sum_{y^n} P(y^n \mid \hat{\theta}(y^n))$ becomes much smaller!

Non-nested models

- We can certainly compare models of entirely different functional form, but same nr of parameters!
- Consider two standard psychological models for relating stimulus strength to perceived strength:
  - Fechner's model
    $$y = a \ln(x + b) + \text{Gaussian noise}$$
  - Stevens' model
    $$y = ax^b + \text{Gaussian noise}$$
- Can use MDL to find which of the two is better for our experimental data!

Part III: difficulties; extensions the general MDL Principle

Difficulties/extensions

- MDL model selection when…
  1. comparing infinitely many models
  2. also need parameter estimates
  3. complexity term infinite
- …solution suggests 'general MDL Principle', beyond model selection

Comparing finitely many models

- Let $\mathcal{M}_1, \ldots \mathcal{M}_K$ be the list of candidate models. MDL selects
$$\arg \min_{i=1..K} -\log P_{\text{nml}}(x^n|\mathcal{M}_i)$$
- Reinterpretation: MDL selects $\mathcal{M}_i$ minimizing the total two-part code length for the data, where data are encoded by (1) uniform code for the model and (2) optimal universal code for the data given the model
$$\arg \min_{i=1..K} -\log P_{\text{nml}}(x^n|\mathcal{M}_i) + L(i)$$
- Here for $i = 1..k, L(i) = \log K$

Comparing infinitely many models

- Select $\arg \min_{i \in \{1,2,\ldots\}} -\log P_{\text{nml}}(x^n|\mathcal{M}_i) + L(i)$
- where now $L(i)$ is the length of some code for *all* the integers, e.g. $L(i) = 2 \log i$
- If we simply picked $\mathcal{M}_i$ minimizing $-\log P_{\text{nml}}(x^n|\mathcal{M}_i)$ then indeed, things might go wrong:
  - If all the $\mathcal{M}_i$ are singleton sets, then we may overfit forever (for example, each $\mathcal{M}_i$ is a Markov chain of some order; the list is such that all Markov chains with rational-valued parameter of each order is included)

General MDL Principle, part I

- Relative to the given set of candidate models,
  - you first devise a single code to encode all possible sequences,
    - This code will be "partly two-part, partly one-part"
  - you then do all inferences based on that code
- Apparently needed to avoid overfitting (our main goal!)
- we will see that it is needed to get a coherent grand picture!

Comparing infinitely many models

- Better not use two-part code for the parameters
  - NML, Bayes give much smaller regret (relative code-lengths)
- We are *forced* to use two-part code for encoding model index
  - Because we want to select a model, we explicity have to encode it
  - Note: complexity of models *not* due to model index!

MDL for parameter estimation

- Indeed, MDL for parameter estimation within a given model $\mathcal{M}$ is **not** the same as maximum likelihood
- Instead, MDL tells you to devise a universal two-part code with smallest possible minimax regret relative to $\mathcal{M}$
- For actual given data x^n, you would then pick the θ minimizing the two-part code length!
  - Because we want to select a parameter (was: model), we explicity have to encode it!
  - Leads to truncated (low-precision) estimates!

MDL for parameter estimation

- Example: $\mathcal{M}$ is Bernoulli model
- Look for code L achieving

$$\inf_L \sup_{x_1,\ldots,x_n} \{ L(x_1, \ldots, x_n) - [-\log P(x^n \mid \hat{\theta}(x^n))] \}$$

....under constraint that L is of form

$$L(x_1, \ldots, x_n) = L'(\theta) - \log P(x_1, \ldots, x_n|\theta)$$

for all $x_1, \ldots, x_n$ and some code L' on (subset of) Θ

MDL for parameter estimation

- Example: $\mathcal{M}$ is Bernoulli model
- ML estimator $\widehat{\theta}$ is equal to frequency of 1's
  - takes value in set $\{0, 1/n, 2/n, \ldots, 1\}$
  - $n+1$ possibilities -> need $\approx \log n$ bits to describe θ
- (rough) MDL estimator is nearest point to $\widehat{\theta}$ in set $\{\frac{1}{\sqrt{n}}, \frac{2}{\sqrt{n}}, \ldots, \frac{\sqrt{n}-1}{\sqrt{n}}\}$
  - need $\approx \log \sqrt{n} = \frac{1}{2} \log n$ bits to describe θ

Difficulties/extensions

- MDL model selection when…
  1. comparing infinitely many models
  2. also need parameter estimates
  3. complexity term infinite/undefined
- …solution suggests 'general MDL Principle', beyond model selection

What if NML distribution undefined?

- In many interesting applications, NML distribution undefined
  - Examples: linear regression, normal distribution: P_{nml} should have density
    $$\frac{f(x^n|\hat{\mu},\hat{\sigma}^2)}{\int_{x^n} f(x^n|\hat{\mu}(x^n),\hat{\sigma}^2(x^n))dx^n}$$
  - Undefined since complexity
    $$\int_{x^n} f(x^n|\hat{\mu}(x^n),\hat{\sigma}^2(x^n))dx^n$$
    diverges!

What if NML distribution undefined?

- In many interesting applications, NML distribution undefined
- In such cases typically also $\int \sqrt{I(\theta)}d\theta$ diverges
- Hence Jeffreys' prior improper
- However, integral typically remains small even if parameters get quite close to boundary of parameter space

Undefined NML, II

- Simplistic solution:
  - start with
    $$f_{\mathrm{nml}}(x^n|\mathcal{M}_{[\tau,K]}) := \frac{f(x^n|\hat{\mu},\hat{\sigma}^2)}{\int_{x^n} f(x^n|\hat{\mu}(x^n),\hat{\sigma}^2(x^n))dx^n}$$
    where (ML) parameters are restricted to
    $$-K \leq \mu \leq K \qquad \sigma^2 > \tau$$
  - this is finite for each pair of 'hyperparameters' K and τ

Undefined NML, II

- Explicitly encode hyperparameters by encoding integers a, b with
  $$\tau = 2^{-a} \qquad K = 2^b$$
- We need
  $$-\log f^*(x^n|\mathcal{M}) := \inf_{\tau,K} \; -\log f_{\mathrm{nml}}(x^n|\mathcal{M}_{[\tau,K]}) + L(\tau) + L(K)$$
  bits
  - Unless for outrageous data sets, need much less bits for hyperparameters than for ordinary parameters

Undefined NML, II

- Explicitly encode hyperparameters by encoding integers a, b with
  $$\tau = 2^{-a} \qquad K = 2^b$$
- We get as our new code length:

$$-\log f^*(x^n|\mathcal{M}) := \inf_{\tau,K} \; -\log f_{\mathrm{nml}}(x^n|\mathcal{M}_{[\tau,K]}) + L(\tau) + L(K)$$

$L(\tau) = 2\log a$, $L(K) = 2\log b$

More sophisticated ideas

– Rissanen's Renormalization (2001)

– Barron and Liang's conditional minimax universal codes

- Elegant solution for variable selection in regression

– Many others (hot topic!)

General Picture

- $\mathcal{M}$ such that there is no universal model $P_{\mathcal{M}}$ achieving minimax optimal regret
- Carve up $\mathcal{M}$ into subsets $\mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \ldots$
  and define P^* such that for each x^n, the regret
  $$-\log P^*(x^n|\mathcal{M}) - [-\log P(x^n|\hat{\theta}(x^n))]$$
  is almost as small as the regret of $P_{\mathrm{nml}}(\cdot \mid \mathcal{M}_j)$, achieving minimax regret for the *smallest* $\mathcal{M}_j$ containing $P(\cdot|\hat{\theta}(x^n))$

General Picture

- $\mathcal{M}$ such that there is no universal model $P_{\mathcal{M}}$ achieving minimax optimal regret
- Carve up $\mathcal{M}$ into subsets $\mathcal{M}_1 \subseteq \mathcal{M}_2 \subseteq \ldots$
  and define P^* such that for each x^n, the regret
  $$-\log P^*(x^n|\mathcal{M}) - [-\log P(x^n|\hat{\theta}(x^n))]$$
  is almost as small as the regret of $P_{\mathrm{nml}}(\cdot \mid \mathcal{M}_j)$, achieving minimax regret for the *smallest* $\mathcal{M}_j$ containing $P(\cdot|\hat{\theta}(x^n))$
- P^* is called (by me) quasi-minimax optimal univ. model. It achieves 'nearly', 'almost' minimax regret:
  $$-\log P^*(x^n \mid \mathcal{M}) = \inf_j \; -\log P_{\mathrm{nml}}(x^n \mid \mathcal{M}_j) + \text{small}.$$

General Principle

- We were doing exactly the same thing when trying to find the best model from a countably infinite list of models
- 'Luckiness' idea:
  - Let $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$ be the union of 1st- and 2nd order MC models, and compare the NML distribution $P_{\mathrm{nml}}(\cdot|\mathcal{M}_2)$ with the distribution
    $$-\log P^*(x^n) = \inf_{k \in \{1,2\}} \; -\log P_{\mathrm{nml}}(x^n|\mathcal{M}_k) + 1$$
    which we implicitly used in model selection

'Luckiness Idea'

- If you're lucky, you need much less bits using the code P^* than the code $P_{\mathrm{nml}}(\cdot|\mathcal{M}_2)$
- If you're not lucky, you need hardly any more bits (max. 1) using the code P^* than the code $P_{\mathrm{nml}}(\cdot|\mathcal{M}_2)$
  - Related to Luckiness principle in Computational Learning Theory (Herbrich and Williamson, 2001)
  - "even if a set of classifiers has infinite VC-dim, your input (x)- data may be such that 'you're lucky' and the effective, 'conditional' VC-dim is very small".

The MDL Principle for coding

- Let $\mathcal{M}$ be the union of all models under consideration. MDL tells you to design a single universal code for $\mathcal{M}$ based on two sub-principles:
- 1. Minimax Principle: try to be as 'honest' as possible, associating $\mathcal{M}$ with minimax regret universal code
- 2. Luckiness/Quasi-minimax Principle: if regret becomes too large, carve up $\mathcal{M}$ into submodels and use a 'quasi-minimax regret' universal model
  - Never much worse than minimax regret model
  - If you're lucky, considerably better than minimax regret model

The MDL Principle for modeling

- Let $\mathcal{M}$ be set of all contemplated distributions
- interested in 'best' explanation for data,
- explanation is an element of a set $\{\mathcal{M}_1, \mathcal{M}_2, \ldots\}$ such that $\cup_k \mathcal{M}_k = \mathcal{M}$
- MDL Principle:
  1. Set up a quasi-minimax regret universal model P^* for $\mathcal{M}$
  2. P^* itself consists of a (quasi-) minimax regret two-part universal code for encoding k and a (quasi-) minimax one-part universal code for encoding x^n given $\mathcal{M}_k$

Examples of General Principle

- old two-part code MDL for models which are best viewed as countable set
  (rather than continuously parameterizable)
- boundary problems of NML;
  improper Jeffreys' prior
- non-parametrics

Two-Part Code MDL

(old!) two-part code MDL (Rissanen '78) :

– Let $\mathcal{H}$ be a set of hypotheses. Given data D pick the $h \in \mathcal{H}$ that minimizes the sum of
  - the description length of the hypothesis h
  - the description length of the data D when encoded 'with the help of the hypothesis h '

Two-Part Code MDL

- For probability models this becomes:
  - Let $\mathcal{P}$ be a countable set of distributions
  - Then two-part code MDL tells us to select the achieving

$$\min_{P \in \mathcal{P}} \{ L(P) - \log P(x^n) \}$$

- For every prior/code length function, this is a universal model, so may be called 'some version of MDL'
- But it's not 'sophisticated MDL' – sophisticated MDL uses (quasi-) minimax regret universal models

Part IV: MDL and Classification

Classification: Overview

1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

Introduction

- MDL mostly developed and studied for probability models
- Yet often applied to models/model classes that are not (directly) interpretable as probability distributions
- Here we apply it to models that are families of classifiers
  - decision trees
  - support vector machines
  - neural networks…

Introduction - II

- There is no unique definition of 'the' MDL Principle for classification
- Yet there is a certain standard approach that has been employed by most authors:
  - Quinlan and Rivest (1989),
  - Rissanen & Wax (1989),
  - Kearns et al. (1997) ;
  - several others…

Introduction - III

- Standard approach has pleasant but also unpleasant properties:
  - strange experimental results (Kearns et al. 1997 (?))
  - can be inconsistent! (Grünwald & Langford, 2003)
    - Even with infinite data, MDL does not identify the classifier with the smallest 'generalization error' (probability of making a wrong prediction) – it asymptotically overfits!
- Several adjustments exist
  - Barron (1991), Yamanishi (1998), McAllester's PAC-Bayes (1999)
  - these are provably consistent
  - but loose some of the pleasant properties of standard approach

Classification: Overview

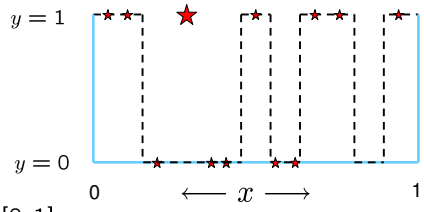
1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

Classification

- Given:
  - Feature space $\mathcal{X}$
  - Label space $\mathcal{Y} = \{0, 1\}$
  - data $D = ((x_1, y_1), \ldots, (x_n, y_n))$
  - countable set $\mathcal{H}$ of hypotheses (classifiers) $h : \mathcal{X} \to \mathcal{Y}$
- Goal: find a $h \in \mathcal{H}$ that makes few mistakes on future data from the same source
  - We say 'h has small generalization error'
  - if data are noisy, then it is not a good idea to adopt the h that minimizes nr of mistakes on the given data
  - leads to over-fitting

Example: intervals (toy) domain

Kearns et al., 1995



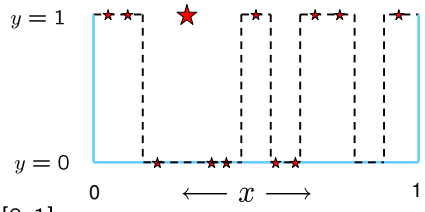
$\mathcal{X} = [0, 1]$

$\mathcal{H}_k$: set of functions $h : \mathcal{X} \to \{0, 1\}$ that switch value k times

$\mathcal{H} = \bigcup_{k=1,2,\ldots} \mathcal{H}_k$

Example: intervals domain

Kearns et al., 1995



$\mathcal{X} = [0, 1]$

$\mathcal{H}_k$: set of functions $h : \mathcal{X} \to \{0, 1\}$ that switch value k times

$\mathcal{H} = \bigcup_{k=1,2,\ldots} \mathcal{H}_k$

the h in picture is in $\mathcal{H}_6$ and makes 1 *classification error* on data D

Two-part code MDL

- We use the oldest, crudest version of MDL (two-part code MDL, Rissanen '78)
- Problematic aspects of MDL for classification are not solved by using modern versions of MDL such as normalized maximum likelihood
  - Grünwald & Langford, 2003
- Using two-part code allows us to keep our story as simple as possible

Two-Part Code MDL

Two-part code MDL:

– Let $\mathcal{H}$ be a set of hypotheses. Given data D pick the $h \in \mathcal{H}$ that minimizes the sum of

- the description length of the hypothesis h
- the description length of the data D when encoded 'with the help of the hypothesis h '

Two-Part Code MDL

Pick $h \in \mathcal{H}$ minimizing

$$\mathrm{DL}(h) + \mathrm{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

Two-Part Code MDL

Pick $h \in \mathcal{H}$ minimizing

$$\mathrm{DL}(h) + \mathrm{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

Encoding of $x_1, \ldots, x_n$ takes $\mathrm{DL}(x_1, \ldots, x_n)$ bits; this term does not involve h . Therefore it plays no role in minimization and can be dropped!

Two-Part Code MDL

Pick $h \in \mathcal{H}$ minimizing

$$\mathrm{DL}(h) + \mathrm{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

Any function on $\mathcal{H}$ satisfying Kraft inequality

Coding Hypotheses

- $\mathrm{DL}(h) = -\log W(h)$, W can be thought of as 'prior' ; many reasonable possibilities
- example code for intervals domain: encode $h \in \mathcal{H}$ in three steps:
  1. Encode number of switches k
  2. Encode 'granularity' d
  3. Code location of k switches within $\{0, \frac{1}{d}, \frac{2}{d}, \ldots, \frac{d-1}{d}\}$

Coding Data

Pick $h \in \mathcal{H}$ minimizing

$$\mathrm{DL}(h) + \mathrm{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n)$$

Code $y_1, \ldots, y_n$ by coding
a. number of mistakes
b. location (index) of mistakes

Coding Data: $\mathrm{DL}(y^n \mid x^n, h)$

- Define:
  - mistake count M_h
    number of mistakes h makes on D
  - 0/1-loss: for $y, \hat{y} \in \{0, 1\}$:
    $L_{01}(y, \hat{y}) := |y - \hat{y}|$
- Formally, $M_h := \sum_{i=1}^{n} L_{01}(y_i, h(x_i))$

Standard approach to coding data

$$\mathrm{DL}^*(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n) = \\ = \log(n+1) + \log \binom{n}{M_h}$$

nr of bits needed to encode total nr of mistakes

nr of bits needed to encode location of mistakes

2p-code length intervals domain

$$\min_{h \in \mathcal{H}} \left\{ \mathrm{DL}(y^n \mid x^n, h) + \mathrm{DL}(h) \right\} =$$

$$\min_{h_{k,d} \in \mathcal{H}} \left\{ \log \binom{n}{M_h} + \log\log k + \log\log d + \log \binom{d}{k} \right\}$$

error term — complexity term

- familiar trade-off between error and complexity
- we can and did leave out $\log(n+1)$ term

MDL Version C0

- We call the coding scheme for 'coding data with the help of hypothesis' MDL Version C0.
- (slight variations of) MDL C0 used by
  - Quinlan and Rivest (1989),
  - Rissanen & Wax (1989),
  - Kearns et al. (1997) ;
  - even Wallace & Boulton (1968)
- But is it the 'right' way to do things?

Potential Problems:

1. Many different coding schemes of data given hypothesis $\mathrm{DL}(y^n|h, x^n)$ possible
   - Comparison strongly indicates that MDL C0 is basically the 'right' coding scheme.
2. Theoretical results on MDL C0
   - (in sharp constrast to probabilistic MDL), analysis strongly indicates that nevertheless something's wrong with MDL C0

Classification: Overview

1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

1. Alternative coding schemes

- Two other coding schemes have been proposed in the literature.
  - seemingly very different, they both lead to same hypothesis selection criterion as MDL C0
  - shows that MDL C0 is special case of general procedure, applicable to arbitrary loss functions
- Evidence that what we're doing is o.k.!

MDL C1: *entropification*

Rissanen 1989, implicit in Vovk 1990
Meir and Merhav 1995, Yamanishi 1998
Grünwald 1998

- Suppose we have a code such that for all h, all (x^n, y^n) , the code length is an increasing affine function of the loss:

$$\mathrm{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + \alpha$$
$$= \beta M_h + \alpha$$

- Here $\beta > 0$; α may depend on n, but not on h

MDL C1: *entropification*

Rissanen 1989, Meir and Merhav 1995,
Yamanishi 1998, Grünwald 1998,
implicit in Vovk 1990 and others

- Suppose we have a code such that for all h, all (x^n, y^n) , the code length is an increasing affine function of the loss:

$$\mathrm{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + \alpha$$
$$= \beta M_h + \alpha$$

- then 'error term' in $\mathrm{DL}(y^n|x^n, h) + \mathrm{DL}(h)$ expresses exactly the error function we are interested in!

entropification

- We can construct a code satisfying
  $\mathrm{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + \alpha$
  by first constructing a conditional probability distribution:
  $P(y|x,h,\beta) := \frac{1}{Z(\beta)} e^{-\beta L_{01}(y;h(x))}$
  $Z(\beta) := \sum_{y \in \{0,1\}} e^{-\beta L_{01}(y;h(x))}$

Note: $Z(\beta)$ does not depend on h or X !

entropification

- We can construct a code satisfying
  $\mathrm{DL}(y^n \mid x^n, h) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + \alpha$
  by first constructing a conditional probability distribution:
  $P(y|x,h,\beta) := \frac{1}{Z(\beta)} e^{-\beta L_{01}(y;h(x))}$
  $Z(\beta) := \sum_{y \in \{0,1\}} e^{-\beta L_{01}(y;h(x))}$
  $P(y^n|x^n,h,\beta) := \prod_{i=1}^n \frac{1}{Z(\beta)} e^{-\beta L_{01}(y_i;h(x_i))}$
- then
  $-\ln P(y^n|x^n,h,\beta) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$

entropification

- For each h, β we constructed a corresponding conditional probability distribution satisfying , for all $D = (x^n, y^n)$,
  $-\ln P(y^n|x^n,h,\beta) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$
- By Kraft inequality, there must also exist a (conditional) code defined on data sequences of length n , satisfying
  $\mathrm{DL}(y^n|x^n,h,\beta) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$
- This is the code we'll use!

entropification

- For each h, β we constructed a corresponding conditional probability distribution satisfying , for all $D = (x^n, y^n)$,
  $-\ln P(y^n|x^n,h,\beta) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$
- By Kraft inequality, there must also exist a (conditional) code defined on data sequences of length n , satisfying
  $\mathrm{DL}(y^n|x^n,h,\beta) = \beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + n \ln Z(\beta)$
  - Code length measured in nats
  - Important: *no* claim that $P(\cdot \mid \cdot, h, \beta)$ generates the data; purely artificial construction to make sure that code length of data given h = linear function of loss h makes on data

entropification

- MDL now becomes: select $h \in \mathcal{H}$ minimizing
  $\beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h)$
- Problem: how to choose β ?
  - different β lead to different choices of h
  - β measures how strongly the 0/1-error should be weighted compared to the 'complexity' of h
    - β viewed as learning rate, inverse 'temperature'

entropification

- MDL now becomes: select $h \in \mathcal{H}$ minimizing
  $\beta \sum_{i=1}^n L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h)$
- Problem: how to choose β ?
  - different β lead to different choices of h
  - β measures how strongly the 0/1-error should be weighted compared to the `complexity' of h
- Intuitive Solution
  - **learn** not just h , but also β from the data

entropification

- MDL now becomes: select $h \in \mathcal{H}$ achieving

$$\min_{h \in \mathcal{H}, \beta \in [0,\infty]} \left\{ \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathsf{DL}(h) + \mathsf{DL}(\beta) \right\}$$

- We'll see in a minute that this does (almost) exactly the same as MDL C0 …

Don't worry about $\mathsf{DL}(\beta)$ for now!

entropification

- MDL now becomes: select $h \in \mathcal{H}$ achieving

$$\min_{h \in \mathcal{H}, \beta \in [0,\infty]} \left\{ \beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathsf{DL}(h) + \mathsf{DL}(\beta) \right\}$$

- We'll see in a minute that this does (almost) exactly the same as MDL C0 …
- …we do this by giving a *third* coding scheme easily shown to be equivalent with MDL C0 and MDL C1 ('entropification')

MDL C2: Probabilistic coding

- Original two-part code MDL (Rissanen '78) was really designed for probability models:
  - Let $\mathcal{P}$ be a countable set of (conditional) distributions on $\mathcal{Y}$ given $\mathcal{X}$
  - Then *probabilistic* two-part code MDL tells us to select the $P \in \mathcal{P}$ achieving

$$\min_{P \in \mathcal{P}} -\log P(y^n \mid x^n) + \mathsf{DL}(P)$$

MDL Version C2: Probabilistic coding

- Original two-part code MDL (Rissanen '78) was designed for probability models only:
  - Let $\mathcal{P}$ be a countable set of (conditional) distributions on $\mathcal{Y}$ given $\mathcal{X}$
  - Then *probabilistic* two-part code MDL tells us to select the $P \in \mathcal{P}$ achieving

$$\min_{P \in \mathcal{P}} -\ln P(y^n \mid x^n) + \mathsf{DL}(P)$$

- We'll recast classification in probabilistic terms

MDL C2

- Define for each $h \in \mathcal{H}$ and 'noise level' $\theta \in [0,1]$ associated Boolean regression model, i.e.

$$Y_i = h(X_i) \text{ xor } Z_i$$

where

$$Z_i \in \{0,1\}, P(Z_i = 1) = \theta, X_i, Y_i, Z_i \text{ i.i.d.}$$

- Let $P(\cdot, | \cdot, h, \theta)$ be the associated conditional distribution:

$$P(y^n | x^n, h, \theta) = \theta^{M_h} (1-\theta)^{n - M_h}$$

MDL Version C2

- MDL C2 tells us to pick (h, θ) minimizing

$$-\ln P(y^n | x^n, h, \theta) + \mathsf{DL}(h) + \mathsf{DL}(\theta) =$$
$$-M_h \ln \theta - (n - M_h) \ln(1-\theta) + \mathsf{DL}(h) + \mathsf{DL}(\theta)$$

MDL C2

- MDL C2 tells us to pick (h, θ) minimizing

$$-\ln P(y^n|x^n, h, \theta) + \mathrm{DL}(h) + \mathrm{DL}(\theta) =$$
$$-M_h \ln \theta - (n - M_h) \ln(1 - \theta) + \mathrm{DL}(h) + \mathrm{DL}(\theta)$$

- MDL C1 tells us to pick (h, β) minimizing

$$\beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h) + \mathrm{DL}(\beta) =$$
$$\beta M_h + n \ln(1 + e^{-\beta}) + \mathrm{DL}(h) + \mathrm{DL}(\beta)$$

- substituting $\beta_\theta := \ln \frac{1-\theta}{\theta}$ shows this is the same!

MDL C2 = MDL C1

- Conclusion:
  - MDL C1 and C2 yield exactly the same hypothesis for the same data,even though codes were motivated differently:
    - version 1: code length of data linear function of loss
    - version 2: probabilistic assumption that data generated by some deterministic process + noise
  - Can encode β by encoding corresponding θ, using $\ln(n + 1)$ nits

MDL C2 vs MDL C0

- MDL C2 tells us to pick (h, θ) minimizing

$$-\ln P(y^n|x^n, h, \theta) + \mathrm{DL}(h) + \mathrm{DL}(\theta) =$$
$$-M_h \ln \theta - (n - M_h) \ln(1 - \theta) + \mathrm{DL}(h) + \mathrm{DL}(\theta)$$

- $\min_{\theta \in [0,1]} \{-M_h \ln \theta - (n - M_h) \ln(1 - \theta)\}$ is achieved for maximum likelihood $\hat{\theta}$:

$$\hat{\theta} := \frac{M_h}{n} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i))$$

so that

$$\min_\theta \{-M_h \ln \theta - (n - M_h) \ln(1 - \theta)\} =$$
$$n[-\hat{\theta} \ln \hat{\theta} - (1 - \hat{\theta}) \ln(1 - \hat{\theta})] = n\mathbf{H}(\hat{\theta})$$

where $\mathbf{H}(\theta)$ is the binary entropy of a coin with bias θ

MDL C2 vs MDL C0

- MDL C2 tells us to pick h minimizing

$$-\ln P(y^n|x^n, h, \theta) + \mathrm{DL}(h) + \mathrm{DL}(\theta) =$$
$$n\mathbf{H}(\hat{\theta}) + \mathrm{DL}(h) + \ln(n + 1)$$

MDL C2 vs MDL C0

Recall: for MDL Version C0 we had

$$\mathrm{DL}(y_1, \ldots, y_n \mid h, x_1, \ldots, x_n) =$$
$$= \ln(n + 1) + \ln \binom{n}{M_h} =$$
$$= \ln(n + 1) + \mathbf{H}\left(\frac{M_h}{n}\right) - \frac{1}{2} \ln n + O(1) =$$
$$= \mathbf{H}(\hat{\theta}) + \frac{1}{2} \ln n + O(1)$$

- standard application of Stirling's approximation

MDL C2 vs MDL C1

- MDL C2 tells us to pick h minimizing

$$-\ln P(y^n|x^n, h, \theta) + \mathrm{DL}(h) + \mathrm{DL}(\theta) =$$
$$n\mathbf{H}(\hat{\theta}) + \mathrm{DL}(h) + \ln(n + 1)$$

- MDL C0 tells us to pick h minimizing

$$n\mathbf{H}(\hat{\theta}) + \mathrm{DL}(h) + \frac{1}{2} \ln n \quad \left[+O(1)\right]$$

- (almost) the same!

MDL C0 = MDL C1 = MDL C2

- Conclusion: all three versions essentially the same!
- Henceforth take MDL C1 as canonical since
  1. it suggests how to extend the approach to different settings (predictors, loss functions)
  2. useful to learn not just h, but also β

More on β.

- MDL C1 tells us to minimize
  $$\beta \sum_{i=1}^{n} L_{01}(y_i; h(x_i)) + n \ln Z(\beta) + \mathrm{DL}(h) + \mathrm{DL}(\beta)$$
- Keeping h fixed and minimizing only over β, min is achieved for $\hat{\beta} = \ln(1-\hat{\theta}) - \ln \hat{\theta}$ with $\hat{\theta} = M_h/n$
  - $\hat{\beta}$ implicitly represents loss that h makes on data
  - Maybe can be used as estimate of h's loss on future data?
- $\hat{\beta}_h < 0$ corresponds to h that makes $> 50\%$ mistakes
  - then $\overline{h}$ is a better predictor than h for given data

Extensions

- Approach can be generalized to (quite) arbitrary symmetric loss fns (Grünwald 98)
  - Example: for the squared error, an analogous story has been known for many years
- Recently, shown that approach can even be generalized to non-symmetric loss functions
  - e.g. $L(1;1) = L(0;0) = 0$; $L(1;0) = 1$; $L(0;1) = 10^6$
  - considerably more complicated

Does it 'work'?

- Would like to show some consistency or rate-of-convergence results, saying that

  'assuming that data are distributed according to some distribution P^*, then with high P^* probability, the hypothesis inferred by MDL C0 converges to the 'best' hypothesis in (closure of) $\mathcal{H}$'

Does it 'work'?

Baby-Theorem (Grünwald 1998, others) Suppose data $(X_1,Y_1),(X_2,Y_2),\ldots,$ are independently and identically distributed according to some distribution P^* on $\mathcal{X} \times \mathcal{Y}$.

Let $\bar{\theta} := \inf_{h \in \mathcal{H}} E_{P^*}[L_{01}(Y; h(X))] = \inf_{h \in \mathcal{H}} P^*(Y \neq h(X))$.
Let $\bar{\beta} := \ln(1-\bar{\theta}) - \ln \bar{\theta}$.

Let $\mathcal{H}$ be **finite**, let DL be a code length function such that $\mathrm{DL}(h)$ is finite for all $h \in \mathcal{H}$. Let $(\hat{h}_n, \hat{\beta}_n)$ be the hypothesis inferred by *MDL-CS* based on the first n outcomes. Then with P^*-probability 1,

$$E_{P^*}[L(Y; \hat{h}_n(X))] \to \inf_{h \in \mathcal{H}} E_{P^*}[L(Y; h(X))] \text{ as } n \to \infty.$$
$$\hat{\beta}_n \to \bar{\beta} \text{ as } n \to \infty.$$

Does it 'work'?

Baby-Theorem (Grünwald 1998, others) Suppose data $(X_1,Y_1),(X_2,Y_2),\ldots,$ are independently and identically distributed according to some distribution P^* on $\mathcal{X} \times \mathcal{Y}$.

Let $\bar{\theta} := \inf_{h \in \mathcal{H}} E_{P^*}[L_{01}(Y; h(X))] = \inf_{h \in \mathcal{H}} P^*(Y \neq h(X))$.
Let $\bar{\beta} := \ln(1-\bar{\theta}) - \ln \bar{\theta}$.

Let $\mathcal{H}$ be **finite**, let DL be a code length function such that $\mathrm{DL}(h)$ is finite for all $h \in \mathcal{H}$. Let $(\hat{h}_n, \hat{\beta}_n)$ be the hypothesis inferred by *MDL-C0* based on the first n outcomes. Then with P^*-probability 1,

$$E_{P^*}[L(Y; \hat{h}_n(X))] \to \inf_{h \in \mathcal{H}} E_{P^*}[L(Y; h(X))] \text{ as } n \to \infty.$$
$$\hat{\beta}_n \to \bar{\beta} \text{ as } n \to \infty.$$

MDL is asymptotically **optimal**

MDL is asymptotically **reliable**

Does it 'work'?

- In words, MDL-C0 is 'consistent':
  - MDL-C0 is capable of finding the 'best' hypothesis, with smallest 'generalization error' (optimality)
  - $\hat{\beta}_n$ can be interpreted as consistent estimator of $P^*(Y \neq \tilde{h}_n(X))$, the generalization error of the hypothesis output by MDL-C0 (reliability).

Does it work?

- Baby-theorem can be extended to infinite $\mathcal{H}$ with finite VC-dimension, or to various forms of 'parametric' $\mathcal{H}$
- More generally, theorem holds for any type of $\mathcal{H}$ satisfying uniform law of large numbers
- But these are typically ***not*** the type of $\mathcal{H}$ we want to apply MDL to!
  - Example: intervals domain/decision trees: $\mathcal{H}$ has infinite VC-dimension

Part IV: Overview

1. Introduction
2. MDL for classification, basic approach
3. The Promise
   - Basic approach has some great properties!
4. The Problem
   - Basic approach shows problematic behaviour
5. Conclusions

Problems for MDL-CS

- What about grown-up versions of our baby-theorem for arbitrary countable $\mathcal{H}$ with $\mathrm{DL}(h)$ arbitrary codes ?
- For probabilistic MDL, general consistency/rate of convergence results exist
  - e.g., Barron and Cover 1991
  - related to Bayesian consistency proofs
- For MDL-C0, no such results exist
- …and in fact, they do not hold!

The Problem

- MDL C1 may be interpreted as applying MDL to a set of countable conditional probability distributions….so it may seem that Barron and Cover's results are still applicable…
- …but they aren't!

The Problem

- Why aren't standard consistency results applicable?
  - These all assume that the 'true' distribution P^* is in (the information closure of) $\mathcal{P}$
  - Our constructed probability distributions implicitly assume that misclassification probability is independent of X :
  - We have, for all $\mathcal{R}_1, \mathcal{R}_2 \subset \mathcal{X}$ with $P^*(X \in \mathcal{R}_i) > 0$

$$P(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_1, \tilde{h}, \tilde{\beta}) = P(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_2, \tilde{h}, \tilde{\beta})$$

- Only if this also holds for 'true' distribution, i.e. if

$$P^*(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_1) = P^*(Y \neq \tilde{h}(X) \mid X \in \mathcal{R}_2)$$

  can B&C's result be applied
- But this is a very strong and unrealistic assumption!

The Problem

- In fact, none of the existing proofs of consistency of MDL or Bayesian procedures for countable models (sets of prob. distributions) can be applied without making unreasonable assumptions on P^*
- Very recently, we showed that in fact, two-part code MDL can indeed be **inconsistent!**
  - Grunwald & Langford, 2003 (under submission/revision)
- Problem not just for MDL but also for 'Bayesian classification under misspecification'

The Problem - II

- We strongly suspect that also more sophisticated versions of MDL (based on normalized maximum likelihood, Bayesian marginal likelihood) can be inconsistent
- …but no proof yet.

Adjusting MDL-C0

- Barron (1991) and Yamanishi (1998) consider adjustments of the MDL-complexity penalty that are provably consistent for inference of predictors for a given loss function
  - classification as special case
- PAC-Bayes: McAllester (1998, 1999, 2001) considers adjustments of Bayesian inference for classification that are provably consistent 'under misspecification'
- Freund, Mansour, Shapire (2003) – another pseudo-Bayesian, provably consistent inference method for classification

Previous Solutions

- All these adjustments typically punish complexity of hypothesis much more heavily than ordinary MDL
- Advantage:
  - this ensures that no asymptotic overfitting takes place…
- Disadvantages:
  - no (straightforward) coding interpretation
  - learning 'slow' compared to ordinary MDL…perhaps slower than necessary?

cf Tsybakov 1999

Example: Yamanishi's MLC

Yamanishi 1998

- MDL-CS:

$$\min_{\beta\in\mathbb{R},h\in\mathcal{H}} \; \beta L_{01}(D;h) + n\psi(\beta) + \mathrm{DL}(h) =$$

$$\min_{h\in\mathcal{H}} \; \hat{\beta}_h L_{01}(D;h) + n\psi(\hat{\beta}_h) + \mathrm{DL}(h)$$

where $\hat{\beta}_h = \ln(1 \quad \hat{\theta}_h) \quad \ln \hat{\theta}_h$ stays away from 0!

- Yamanishi's MLC:

$$\min_{h\in\mathcal{H}} \; \beta_n L_{01}(D;h) + n\psi(\beta_n) + \mathrm{DL}(h)$$

where $\beta_n = \Theta(\sqrt{\frac{\ln n}{n}})$ goes to 0!

Example: Yamanishi's MLC

Yamanishi 1998

- Yamanishi's MLC:

$$\min_{h\in\mathcal{H}} \; \beta_n L_{01}(D;h) + n\psi(\beta_n) + \mathrm{DL}(h)$$

$$\beta_n = \Theta(\sqrt{\frac{\ln n}{n}})$$

- Equivalently,

$$\min_{h\in\mathcal{H}} \; L_{01}(D;h) + \frac{1}{\beta_n}\mathrm{DL}(h)$$

- Compare to Barron's (1991) regularization:

$$\min_{h\in\mathcal{H}} \; L_{01}(D;h) + \lambda\sqrt{n\mathrm{DL}(h)}$$

where λ is some positive constant

Ubiquitous $\sqrt{n}$!

- McAllester's PAC-Bayes also leads to a model selection criterion with $\sqrt{n}$ factor in front of complexity term
  - some important refinements though
- $\sqrt{n}$ also hidden in Freund, Mansour, Shapire's work

Problems

- Approaches that are provably consistent have $\beta_n \to 0$ as n increases. Problems (in my view):
  1. There is no clear coding interpretation any more (following Rissanen, I would like to keep the coding interpretation if at all possible)
  2. β_n cannot be interpreted as an estimator of the loss h will make on future data any more (following intuition, I would like to keep this interpretation if at all possible!)
  3. Complexity penalties may (?) sometimes be larger than necessary (viz Tsybakov's recent work)
     - Smaller penalties may give better rates of convergence for certain classes of 'true' P^*

Classification – Conclusion I

- Two-part code MDL can fail for classification
- More sophisticated versions of MDL/Bayes can fail as well (did not discuss this in detail)
- In practice though, MDL often slightly underfits rather than overfits!
  - Possible reason: code length based on local rather than global optima in error surface (?)

Classification – Conclusion II

- 'raw' MDL suited and designed for probability models
  - typically consistent if well-specified, i.e. if 'true' data-generating distribution in (closure) of model $\mathcal{M}$
  - Consistent under misspecification under certain conditions, e.g. if $\mathcal{M}$ is a convex set of distributions
- MDL turns non-probability models (e.g. classifiers) into codes (probability distributions) first; the resulting model is typically misspecified and, unfortunately not convex…so that we may get inconsistency

Part V: MDL and the others: justifications/comparisons

Part V: Overview

- MDL in practice: does it work?
  - practical justification of MDL?
- MDL and frequentist statistics;
  - frequentist justification of MDL
- MDL and Bayesian statistics
  - Bayesian justification of MDL, or MDL justification of Bayes?
- MDL and other information-theoretic methods
  - MML, Maximum Entropy, Kolmogorov MSS

MDL in practice: does it work?

- Distinguish between
  - MDL for probability models:
    by and large, yes!!!
  - MDL for general predictors/loss functions:
    - problematic behaviour;
    - not very well-developed yet! (different talk)

MDL in practice: does it work?

MDL for probability models:

- MDL/Bayes with Jeffreys prior for discrete data-problems (e.g. Markov chain model selection) – works extremely well! (www.mdl-research.org)
- Predictive/prequential MDL: generally works very well!
- The 'parameter space boundary problem' has plagued NML-MDL applications a lot…
- Use of asymptotic approximations – very mixed results; sometimes not so good

MDL in practice: does it work?

- MDL has been quite helpful in cognitive psychology since it could help explain observed differences in model flexibility between different models with the same number of parameters
- Much more experimentation with non-Bayesian universal models needs to be done!

Part V: Overview

- MDL in practice: does it work?
  - practical justification of MDL?
- MDL and frequentist statistics;
  - frequentist justification of MDL
- MDL and Bayesian statistics
  - Bayesian justification of MDL, or MDL justification of Bayes?
- MDL and other information-theoretic methods
  - MML, Maximum Entropy, Kolmogorov MSS

Does it 'work' in frequentist sense?

- Main concern: consistency / rates of convergence
  - Suppose data are actually distributed according to a distribution P^* in one of the models under consideration (P^* is the 'true' distribution)
  - Roughly speaking, a learning method is consistent if, with high P^*-probability , the distribution $\hat{P}$ inferred by the procedure converges to P^*
  - This should hold for 'all' $P^* \in \mathcal{M}_1 \cup \mathcal{M}_2 \cup \ldots$
  - Definition slightly adjusted for model selection

Does it 'work' in frequentist sense?

- Two-Part code MDL: we can apply Blackwell & Dubins (1962) famous result about consistency of Bayes with countable family of otherwise almost completely arbitrary distributions
  - small prior (large code length) for true distribution means that learning takes longer, but eventually, MDL will select the true distribution
  - Provides external motivation for taking 'minimax' priors/code-lengths!
- Extended and elaborated by Barron & Cover (1991): 'sophisticated' two-part code MDL is consistent under very general conditions
  - two-part code MDL is consistent also if true distribution is only a limit of distributions with finite code length
  - Gives instructions on how to discretize continuous model spaces to get good 'rates of convergence'

Does it 'work' in frequentist sense?

- Also, more 'modern' versions of MDL (NML, Bayes, prequential) are typically consistent
- Rule of thumb: MDL procedures are 'consistent' whenever Bayes' procedures are consistent
  - rates of convergence comparable to Bayes.
  - Surprising exception: (Csiszár, Shields 2000)
- Means (well…) that, at least asymptotically, MDL effectively counters overfitting!

Aside: MDL Philosophy

Rissanen's extreme position:

- The assumption that there exists a 'probability distribution generating the data' is untenable in many interesting applications (e.g. speech recognition, computer vision)
- Basing a statistical inference procedure on the assumption that a true distribution exists, and calculating the strategy that finds this distribution as fast as possible, is then methodologically flawed. It is based on an inherently untestable (!) and probably false assumption, and therefore unclear what it does in practice

Aside: MDL Philosophy

- Instead, statistical procedures should be based on properties of the data and the model alone and not on anything inherently unobservable such as a 'true distribution'
  - e.g., the NML distribution compresses data most in the worst-case over all sequences, relative to a given model, independent of whatever process generated that data! No assumption that the model (or anything else for that matter) generated the data
  - if you use MDL for online coding/prediction you have guaranteed relative performance on **all** data sequences, and not just with high probability under true model!

Aside: MDL Philosophy

- Nevertheless, consistency of a statistical method is important also for Rissanen :
  - In the idealized case where a true distribution really exists and is in our model, the method better finds it with high probability, given enough data!
- Rissanen simply insists that the method is not constructed under the assumption of an idealized and unrealistic state of affairs; but if that assumption holds, the method better give good results!
  - Consistency as a sanity check rather than a design principle

Part V: Overview

- MDL in practice: does it work?
  - practical justification of MDL?
- MDL and frequentist statistics;
  - frequentist justification of MDL
- MDL and Bayesian statistics
  - Bayesian justification of MDL, or MDL justification of Bayes?
- MDL and other information-theoretic methods
  - MML, Maximum Entropy, Kolmogorov MSS

MDL and Bayes

- Heated debates galore! ('MDL is just Bayes')
- First insight:
  - Two tenets of Bayesian statistics:
    1. All uncertainty should be handled using probability
    2. All decisions should be done based on (expectations according to) prior/posterior
  - MDL sticks with 1, not 2 (NML code!)

Let's be careful!

1. Formally there certainly exist some differences
2. Practically
   - as long as MDL/Bayes only used for model selection, differences are quite small
   - If MDL/Bayes are also used for prediction against arbitrary loss function, quite different!
3. Philosophically differences are substantial

1. Formal differences

- MDL does not restrict type of universal model used; Bayes forces use of Bayesian marginal likelihood/2-part code universal model
  - MDL has more freedom – can use prequential/NML
- When MDL is implemented as a Bayesian universal model, then the used prior is artificially constructed to achieve minimax/quasi minimax code lengths; it does not reflect prior knowledge
  - MDL has less freedom

2(a) Practical Comparison – model selection

- MDL/Bayes usually give very similar results:
  - NML and Bayes universal model with Jeffreys' prior often very similar, even for small sample sizes
  - in 'objective Bayesian' branch of Bayes, often use the same priors as MDL, so MDL very much like Bayes
  - more generally, prior usually does not play a large role – so still MDL behaves like Bayes
  - But there are exceptions – think of NML model with local maximum likelihood complexity term. There seems to be no analogue of that in Bayes!

2(b) Practical Comparison – minimizing expected loss

- If inferred model is used for prediction, then MDL and Bayes become quite different again:
- Let $\ell : \mathcal{X} \times \mathcal{A} \to [0, \infty)$ be some loss function; suppose data $x_1, \ldots, x_{i-1}$ observed; have to make decision about new outcome x_i
  - Bayes: optimal decision/action is the one minimizing posterior expected loss:

$$\inf_{a \in \mathcal{A}} E_{\Theta \sim W(\cdot|x_1,\ldots,x_{i-1})} E_{X_i \sim \theta}[L(X; a)]$$

.

2(b) Practical Comparison – minimizing expected loss

- If inferred model is used for prediction, then MDL and Bayes become quite different again:
- Let $\ell : \mathcal{X} \times \mathcal{A} \to [0, \infty)$ be some loss function; suppose data $x_1, \ldots, x_{i-1}$ observed; have to make decision about new outcome x_i
  - Bayes: optimal decision/action is the one minimizing posterior expected loss:

$$\inf_{a \in \mathcal{A}} E_{\Theta \sim W(\cdot|x_1,\ldots,x_{i-1})} E_{X_i \sim \theta}[L(X; a)]$$

  - MDL: this is meaningless! Prior constructed to achieve minimax code-length and has no meaning beyond that.

2(b) Practical Comparison – minimizing expected loss

- MDL Priors for Bayesian/2-part universal models are artificially constructed to achieve minimax code-lengths
- Do not represent degree-of-belief in models/distributions
- Do not have long-run frequency interpretation
- Then not at all clear why making predictions with small expected loss (according to prior) would ever lead to small actual loss

2(c) Philosophical Comparison

- Bayes:
  - prior represents degree of belief in different 'states of nature (distributions)' ;
  - given enough data, posterior concentrates on true state of nature (distribution): 'belief becomes correct'
- MDL – very different:
  - there is no such thing as a true distribution; let alone a randomized process by which a true state of nature is generated; prior is a tool to compress data in stages
  - We only assume that data exhibits regularities; and that the same regularities will also be present in future data coming from the same phenomenon. We try to find those regularities by compressing data as much as possible!

2(c) Philosophical Comparison

- Statisticians and ML researchers often use Bayes for models they a priori know to be completely wrong:
  - Naïve Bayes
  - Markov models for speech recognition
- If they were strictly Bayesian, they would put prior probability 0 instead of 1 on these models!
- Most justifications of Bayesian statistics are implicitly based on the true distribution having prior density > 0 ; so why does Bayes often still work well when you know beforehand this is not the case?
- Rissanen answers, angrying the Bayesians: '*Bayes works well because it resembles MDL, which has better justification!*'
- Is he right?

2(c) Philosophical Comparison

- In MDL we certainly don't believe that a hidden Markov model generates speech. But we do believe that some hidden Markov models allow for substantial compression of speech signals.
- By putting priors on hidden Markov models we can create a universal model that, given enough data, lets us learn 'what Markov model best compresses speech'
- We then hope that this most-compressing Markov structure leads to good predictions of speech signals
  - Strong point: use of priors justified without the need that prior of 'true distribution' > 0
  - Weak point: why should good compression lead to good speech recognition?

But does any of this matter?

- 'I am a practitioner and I want to use Bayesian model selection. Should I also learn about MDL?'
  - Yes, because
    - it offers you methods like NML and prequential coding which you won't find in any Bayesian textbook
    - it teaches you to be very careful about how to use your posterior
- 'I am a practitioner and I want to use MDL model selection. Should I also learn about Bayes?'
  - OF COURSE!
    - Much more research, much more experience, much better developed

Let's be even more careful: Brands of Bayesian Statistics

'modern' Bayesian Statistics has (at least) three founding fathers, each with (quite) different ideas

L. Savage
The Foundations of Statistics (1954)

B. De Finetti
Theory of Probability ('1937',1974)

H. Jeffreys
Theory of Probability (1939, 1961)

MDL and Savage

- Mainstream Bayesian statistics mostly based on Savage's ideas
- But what about De Finetti?

MDL and De Finetti

- MDL (that is, Rissanen) considers probabilities of data as *subjective* - probabilities are something to be used for prediction or description, the 'true' distribution does not exist other than as a mental construct

Rissanen: 'There is no such thing as a 'true distribution'. We only have the data'

- … so, in the end, this *is* very similar to De Finetti's ideas:

De Finetti: 'Probabilities Do Not Exist'

Part V: Overview

- MDL in practice: does it work?
  - practical justification of MDL?
- MDL and frequentist statistics;
  - frequentist justification of MDL
- MDL and Bayesian statistics
  - Bayesian justification of MDL, or MDL justification of Bayes?
- MDL and other information-theoretic methods
  - MML, Maximum Entropy, Kolmogorov MSS

MDL and MML

- MML is a method for hypothesis/model selection that is quite similar to MDL in some ways yet very different in other ways
  - Wallace and Boulton (1968 (!), 1975),
  - Wallace and Freeman (1987), …
- MML bases all inferences on 2-part codes
  - no NML, Bayes mixture
- MML's two-part codes assign optimal expected code lengths
  - Expectation based on a Bayesian subjective prior on hypotheses: *'MML is Bayesian!'*

Maximum Entropy and MDL

- MDL associates with family of distributions $\mathcal{M}$ a single distribution $P_{\mathrm{nml}}(\cdot \mid \mathcal{M})$ that achieves minimax relative code-lengths
  - (logarithmic regret)
- MaxEnt associates with convex family of distributions $\mathcal{M}$ a single distribution $P_{\mathrm{me}}(\cdot \mid \mathcal{M})$ that achieves minimax absolute code-lengths
  - (logarithmic loss)

Topsøe 1979 / Grünwald 1998 / Grünwald & Dawid 2003

Kolmogorov Complexity/MSS

- Rissanen does not believe that true distributions or models exist. He thinks the goal of inductive inference should be to pick the model that 'captures the most regularity in the data'
  - i.e. best summarizes the data, give the meaningful information in the data
  - He tries to justify MDL in terms of the **Kolmogorov Minimal Sufficient Statistic**
    - Vereshchagin and Vitányi 2002
    - based on lossy rather than lossless compression

Wrapping up: General Conclusions; Current Research

Some General Conclusions

- MDL Principle: inference by minimizing code lengths, using codes based on 'uniform' (minimax) and 'quasi-uniform' (almost minimax) codes
- Code lengths can be computed in various ways
  - Actual *codes* are *never ever* constructed!
  - Be *very* careful with using asymptotic expansions!
- *Not* Bayes, *not* BIC
  - but close in spirit to De Finetti-Bayes

Current/Future Research

- Mainly developed for probability models
  - current research, 'hot topics':
    - What if NML distribution/Jeffreys' prior undefined? dealing with boundaries of parameter space
    - nonparametric density estimation
    - compare to AIC, cross-validation etc.
- MDL for predictors (classifiers etc.) still problematic
  - doesn't always work!
  - current research:
    - Does it 'usually' work?
    - Can MDL be adjusted for bad cases?

Additional Topics

- MDL for regression (big topic)
- MDL for time series
- MDL for clustering
- MDL and 'universal prediction'
- Universal models of the second kind (worst-case expected rather than actual regret) ('Barron's MDL')

Thank you for your attention!