To be presented at the Sixth International Workshop on Artificial Intelligence and Statistics (Ft. Lauderdale, USA, January 1997)

Comparing Predictive Inference Methods for Discrete Domains

Petri Kontkanen, Petri Myllymäki, Tomi Silander, and Henry Tirri
Complex Systems Computation Group (CoSCo)*
P.O.Box 26, Department of Computer Science
FIN-00014 University of Helsinki, Finland

Peter Grünwald
CWI, Department of Algorithms and Architectures
P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands

Abstract

Predictive inference is seen here as the process of determining the predictive distribution of a discrete variable, given a data set of training examples and the values for the other problem domain variables. We consider three approaches for computing this predictive distribution, and assume that the joint probability distribution for the variables belongs to a set of distributions determined by a set of parametric models. In the simplest case, the predictive distribution is computed by using the model with the *maximum a posteriori (MAP)* posterior probability. In the *evidence* approach, the predictive distribution is obtained by averaging over all the individual models in the model family. In the third case, we define the predictive distribution by using Rissanen's new definition of *stochastic complexity*. Our experiments performed with the family of Naive Bayes models suggest that when using all the data available, the stochastic complexity approach produces the most accurate predictions in the log-score sense. However, when the amount of available training data is decreased, the evidence approach clearly outperforms the two other approaches. The MAP predictive distribution is clearly inferior in the log-score sense to the two more sophisticated approaches, but for the 0/1-score the MAP approach may still in some cases produce the best results.

1 Introduction

Let us consider the problem of determining the conditional distribution for a set of discrete random variables (the *query*), given the values of all the other problem domain variables (the *input*), and a data set of past cases (the *training set*). For computational reasons, we restrict ourselves here to the standard classification problem, where the query set contains only one variable, although the approach has a straightforward extension to cases with more than one query variable.

For computing the *predictive distribution* for the possible values of the query variable, we consider here three different methods, each exploiting a different joint probability distribution for the variables. All the three joint probability distributions are assumed to belong to a family of distributions determined by a set of parametric models. In the simplest, *maximum a posteriori (MAP)* case, we determine first the model (i.e., the parameter instantiation) with the highest posterior probability, and the predictive distribution is then computed by using this single model. In the *evidence* approach, the predictive distribution is obtained by integrating over all the individual models in the model family. From the information theory point of view, minus the logarithm of the evidence integral is the code length needed for coding the training set, the input, and the prediction, with respect to a given model family. Consequently, the logarithm of evidence can be seen as an approximation of *stochastic complexity (SC)*, the shortest possible codelength for the data as defined in [5]. Rissanen has recently [6] introduced an alternative coding scheme, which in some cases produces much shorter codes than the evidence approach, while retaining the code length approximately the same for the

*URL: http://www.cs.Helsinki.FI/research/cosco/

other cases. In our third approach, we define the predictive distribution by using Rissanen's new definition of stochastic complexity. The three predictive inference methods used are described in more detail in Section 2.

Computing the MAP or SC predictive distribution (or at least a good approximation of it) is often feasible in practice, but the evidence approach requires integrating over the whole model space, which is infeasible for many model families. In order to be able to compare the three approaches mentioned above, in this paper we restrict ourselves to a simple family of models, the set of Naive Bayes classifiers. In this case, all three predictive distributions can be represented in a form allowing a computationally efficient implementation, as we will show in Section 3. The predictive accuracy of the three predictive inference methods in the Naive Bayes case was evaluated empirically by using publicly available classification data sets. The results are presented in Section 4. Despite of the simplicity of the parametric form, the family of Naive Bayes models is widely used in practice, and thus the results are interesting also from a practitioner's point of view.

2 Predictive Inference Methods

In this paper we restrict ourselves to discrete attributes, and model the problem domain by $m+1$ discrete random variables $X_1, \ldots, X_m, Y$. A *data instantiation* $\vec{d}$ is a vector in which all the variables X_i, Y have been assigned a value, $\vec{d} = (X_1 = x_1, \ldots, X_m = x_m, Y = k)$, where $x_i \in \{x_{i1}, \ldots, x_{in_i}\}, k \in \{1, \ldots, K\}$. A *random sample* $D = (\vec{d}_1, \ldots, \vec{d}_N)$ is a set of N i.i.d. (independent and identically distributed) data instantiations, where each $\vec{d}_j$ is sampled from $\mathcal{P}$, the joint distribution of the variables $(X_1, \ldots, X_m, Y)$.

Given the *training data* D, the conditional distribution of a new *test vector* $\vec{d}$ is $\mathcal{P}(\vec{d}|D)$,

$$\mathcal{P}(\vec{d}|D) = \frac{\mathcal{P}(\vec{d}, D)}{\mathcal{P}(D)}. \tag{1}$$

In this paper we focus on the following prediction problem: Given the values of the variables $X_1, \ldots, X_m$, and the training data D, we wish to predict the value of variable Y. More precisely, let $I = (X_1 = x_1, \ldots, X_m = x_m)$ denote the variable assignments given. Now we wish to compute the probabilities $\mathcal{P}(Y = k|I, D)$ for each possible value $k, k = 1, \ldots, K$. Using the basic rules of probability theory we can write

$$\begin{aligned} \mathcal{P}(Y = k|I, D) &= \frac{\mathcal{P}(I, Y = k, D)}{\mathcal{P}(I, D)} = \frac{\mathcal{P}(I, Y = k, D)}{\sum_{k'=1}^{K} \mathcal{P}(I, Y = k', D)} \\ &= \frac{\mathcal{P}(\vec{d}[k], D)}{\sum_{k'=1}^{K} \mathcal{P}(\vec{d}[k'], D)} = \frac{\mathcal{P}(\vec{d}[k]|D)}{\sum_{k'=1}^{K} \mathcal{P}(\vec{d}[k']|D)}, \end{aligned} \tag{2}$$

where $\vec{d}[k]$ denotes the vector $(I, Y = k) = (X_1 = x_1, \ldots, X_m = x_m, Y = k)$. Consequently, the conditional distribution for variable Y can be computed by using the complete data vector conditional distributions (1) for each of the possible complete vectors $\vec{d}[k]$. We call the resulting distribution the *predictive distribution* of Y. This approach can be straightforwardly extended to cases with more than one uninstantiated variable, but it should be noted that the number of terms to be compared grows exponentially with respect to the number of free variables.

Naturally, in practice the "true" problem domain probability distribution $\mathcal{P}$ is unknown, so we are left with an approximation. A common procedure is to restrict the search for a good approximation of $\mathcal{P}$ to some parametric family of models $\mathcal{M}$, where each instantiation of parameters Θ corresponds to a single distribution. In this paper we consider the following three different candidates for approximating $\mathcal{P}$.

2.1 The MAP predictive distribution

Given a prior distribution $P(\Theta)$ over the space of parameters, we can arrive at a posterior distribution $P(\Theta|D)$ by using Bayes' rule:

$$P(\Theta|D) \propto P(D|\Theta)P(\Theta). \tag{3}$$

In the *maximum a posteriori (MAP) probability* approximation, distribution $\mathcal{P}$ is replaced by the distribution corresponding to the single model $\hat{\Theta}(D) = \arg\max_{\Theta} P(\Theta|D)$, i.e., the model maximizing the posterior distribution $P(\Theta|D)$:

$$\mathcal{P}_{\mathrm{map}}(\vec{d},D) = P(\vec{d},D|\hat{\Theta}(D)),$$

In the more classical approach, the MAP model is replaced by the *maximum likelihood (ML)* model $\tilde{\Theta}$, i.e., by the model maximizing the data likelihood $P(D|\Theta)$. In this paper we assume the prior distribution $P(\Theta)$ to be uniform, in which case the MAP model is equal to the ML model, as can clearly be seen from (3). The corresponding predictive distribution (2) is in this case

$$\begin{aligned}\mathcal{P}_{\mathrm{map}}(Y=k|I,D) &= \frac{\mathcal{P}_{\mathrm{map}}(\vec{d}[k],D)}{\sum_{k'=1}^{K}\mathcal{P}_{\mathrm{map}}(\vec{d}[k'],D)} = \frac{P(\vec{d}[k],D|\hat{\Theta}(D))}{\sum_{k'=1}^{K}P(\vec{d}[k'],D|\hat{\Theta}(D))} \\ &= \frac{P(\vec{d}[k]|\hat{\Theta}(D)P(D|\hat{\Theta}(D))}{\sum_{k'=1}^{K}P(\vec{d}[k']|\hat{\Theta}(D))P(D|\hat{\Theta}(D))} = \frac{P(\vec{d}[k]|\hat{\Theta}(D))}{\sum_{k'=1}^{K}P(\vec{d}[k']|\hat{\Theta}(D))}. \end{aligned} \tag{4}$$

The second to last equality follows from the assumption that the data are i.i.d., so the vectors are independent given a model Θ.

2.2 The evidence predictive distribution

A more sophisticated approximation than the MAP approach can be obtained by integrating over the model space, i.e., by *averaging over all the models* Θ:

$$\mathcal{P}_{\mathrm{ev}}(\vec{d},D) = \int P(\vec{d},D|\Theta)P(\Theta)d\Theta. \tag{5}$$

In the Bayesian literature this integral is often called the *evidence*. The resulting predictive distribution is

$$\mathcal{P}_{\mathrm{ev}}(Y=k|I,D) = \frac{\mathcal{P}_{\mathrm{ev}}(\vec{d}[k],D)}{\sum_{k'=1}^{K}\mathcal{P}_{\mathrm{ev}}(\vec{d}[k'],D)} = \frac{\int P(\vec{d}[k],D|\Theta)P(\Theta)d\Theta.}{\sum_{k'=1}^{K}\int P(\vec{d}[k'],D|\Theta)P(\Theta)d\Theta.}. \tag{6}$$

2.3 The stochastic complexity predictive distribution

Rissanen [5, 6] has introduced the notion of *stochastic complexity* of a data set D relative to a class of models $\mathcal{M}$ to be the code length of D when it is encoded using the shortest code obtainable with the help of the class $\mathcal{M}$. Here by 'shortest code' we mean the code that gives as short as possible a code length to *all* possible data sets D.

It is a well-known fact from information theory that for any complete code C , there is a corresponding probability distribution P_C such that for all sets D, $-\log P_C(D)$ is the length of the encoding of D when the encoding is done using C. Similarly, for all probability distributions P over data sets D there is a code C_P such that for any data set D the codelength of D when encoded with C_P is equal to $\lceil -\log P(D) \rceil$. This implies that the stochastic complexity can be written as $-\log \mathcal{P}_{\mathrm{sc}}$ where $\mathcal{P}_{\mathrm{sc}}$ is a probability distribution that, in a sense to be explained later, gives as much probability as possible to all D. This motivates the use of $\mathcal{P}_{\mathrm{sc}}$ for prediction.

In terms of formulas, $-\log \mathcal{P}_{\mathrm{ev}}(D)$ has served as the original definition of the stochastic complexity. Recently, however, Rissanen [6] has shown that there exists a code that is itself not dependent on any prior distributions of parameters and which in general yields even shorter codelengths than the code with lengths $-\log \mathcal{P}_{\mathrm{ev}}(D)$. Here 'shorter' means that for some data sets the codelength will be considerably shorter, while for most data sets it will be negligibly longer. Hence $\mathcal{P}_{\mathrm{sc}}$ will give a much higher probability than $\mathcal{P}_{\mathrm{ev}}$ to some data sets and approximately equal probability to all other ones. This led Rissanen to redefine stochastic complexity using this new code. In the case of discrete data, the new stochastic complexity for data $(\vec{d},D)$ with respect to model class $\mathcal{M}$ can be written as $-\log \mathcal{P}_{\mathrm{sc}}$ with

$$\mathcal{P}_{\mathrm{sc}}(\vec{d},D) = \frac{P(\vec{d},D|\tilde{\Theta}(\vec{d},D))}{\sum_{\vec{d}',D'}P(\vec{d}',D'|\tilde{\Theta}(\vec{d}',D'))},$$

where the sum in the denominator goes over all the all the possible instantiations of the data set $D \cup \vec{d}$, and $\tilde{\Theta}(\vec{d},D) \in \mathcal{M}$ denotes the maximum likelihood model for this data $(\vec{d},D)$.

Using this definition of $\mathcal{P}_{\rm sc}$, we can obtain a predictive distribution as follows:

$$\mathcal{P}_{\rm sc}(Y=k|I,D) \quad = \quad \frac{\mathcal{P}_{\rm sc}(\vec{d}[k],D)}{\sum_{k'=1}^{K}\mathcal{P}_{\rm sc}(\vec{d}[k'],D)} = \frac{\frac{P(\vec{d}[k],D|\tilde{\Theta}(\vec{d}[k],D))}{\sum_{\vec{d}',D'}P(\vec{d}',D'|\tilde{\Theta}(\vec{d}',D'))}}{\sum_{k'=1}^{K}\frac{P(\vec{d}[k'],D|\tilde{\Theta}(\vec{d}[k'],D))}{\sum_{\vec{d}',D'}P(\vec{d}',D'|\tilde{\Theta}(\vec{d}',D'))}} \tag{7}$$

At first sight, this probability may seem hard to compute as we have to sum over all (exponentially many) possible instantiations of the data set $D \cup \vec{d}$. But a closer inspection reveals that the two exponential sums in the rightmost part of (7) cancel out and thus we obtain:

$$\mathcal{P}_{\rm sc}(Y=k|I,D) = \frac{P(\vec{d}[k],D|\tilde{\Theta}(\vec{d}[k],D))}{\sum_{k'=1}^{K}P(\vec{d}[k'],D|\tilde{\Theta}(\vec{d}[k'],D))} = \frac{P(\vec{d}[k]|\tilde{\Theta}(\vec{d}[k],D))P(D|\tilde{\Theta}(\vec{d}[k],D))}{\sum_{k'=1}^{K}P(\vec{d}[k']|\tilde{\Theta}(\vec{d}[k'],D))P(D|\tilde{\Theta}(\vec{d}[k'],D))}. \tag{8}$$

Though at first sight this formula looks similar to that of the maximum likelihood predictor (4), it should be noted that the probabilities $P(D|\tilde{\Theta}(\vec{d}[k'],D))$ do not cancel out here since the maximum likelihood estimator appearing in the denominator of (8) depends on k' and hence is not a constant. Moreover, the maximum likelihood estimator $\tilde{\Theta}(\vec{d}[k],D)$ is now computed by using the data set $D \cup \vec{d}$, not just D.

3 Predictive distribution of the Naive Bayes classifier

In the Naive Bayes classifier, the variables $X_1,\ldots,X_m$ are assumed to be independent, given the values of variable Y (the "class" variable). It follows that the joint probability distribution for a data vector $\vec{d}$ can be written as

$$P(\vec{d}) = P(X_1=x_1,\ldots,X_m=x_m,Y=k) = \sum_{k'=1}^{K}\left(P(Y=k)\prod_{i=1}^{m}P(X_i=x_i|Y=k)\right).$$

Consequently, in the Naive Bayes model family, a single distribution P can be uniquely determined by fixing the values of the parameters $\Theta = (\alpha,\Phi)$, where

$$\alpha = (\alpha_1,\ldots,\alpha_K) \text{ and } \Phi = (\Phi_{11},\ldots,\Phi_{1m},\ldots,\Phi_{K1},\ldots,\Phi_{Km}),$$

with the denotations

$$\alpha_k = P(Y=k), \Phi_{ki} = (\phi_{ki1},\ldots,\phi_{kin_i}), \text{ where } \phi_{kil} = P(X_i=x_{il}|Y=k).$$

In the following we assume that $\alpha_k > 0$ and $\phi_{kil} > 0$ for all k,i, and l. Furthermore, both the class variable distribution $P(Y)$ and the intra-class conditional distributions $P(X_i|Y=k)$ are multinomial, i.e., $Y \sim \text{Multi}(1;\alpha_1,\ldots,\alpha_K)$, and $X_{i|k} \sim \text{Multi}(1;\phi_{ki1},\ldots,\phi_{kin_i})$. Since the family of Dirichlet densities is *conjugate* (see e.g. [2]) to the family of multinomials, i.e. the functional form of parameter distribution remains invariant in the prior-to-posterior transformation, we assume that the prior distributions of the parameters are from this family. More precisely, let $(\alpha_1,\ldots,\alpha_K) \sim \text{Di}\,(\mu_1,\ldots,\mu_K)$, and $(\phi_{ki1},\ldots,\phi_{kin_i}) \sim \text{Di}\,(\sigma_{ki1},\ldots,\sigma_{kin_i})$, where $\{\mu_k,\sigma_{kil} \mid k=1,\ldots,K; i=1,\ldots,m; l=1,\ldots,n_i\}$ are the *hyperparameters* of the corresponding distributions. Assuming that the parameter vectors α and Φ_{ki} are independent, the joint prior distribution of all the parameters Θ is

$$\text{Di}\,(\mu_1,\ldots,\mu_K)\prod_{k=1}^{K}\prod_{i=1}^{m}\text{Di}\,(\sigma_{ki1},\ldots,\sigma_{kin_i}).$$

For simplicity, in our experiments we have used the uniform prior for both the MAP and evidence prediction: all μ_k and σ_{kij} are set to 1. Having now defined the prior distribution, the predictive distributions (4),(6), and (7) can be written more explicitly. Notice that in all of these formulas the normalizing constant, i.e., the denominator, has been left out for notational simplicity. The MAP predictive distribution is proportional to the likelihood of the test vector $\vec{d}[k]$:

$$\mathcal{P}_{\text{map}}(Y = k|I, D) \propto P(\vec{d}[k]|\tilde{\Theta}(D)) = \hat{\alpha}_k \prod_{i=1}^{m} \hat{\phi}_{kix_i}, \text{ where}$$

$$\hat{\alpha}_k = \frac{h_k + \mu_k - 1}{N + \sum_{k'=1}^{K} \mu_{k'} - K}, \quad \hat{\phi}_{kil} = \frac{f_{kil} + \sigma_{kil} - 1}{h_k + \sum_{l=1}^{n_i} \sigma_{kil} - n_i},$$

and h_k and f_{kil} are the sufficient statistics of the training data D, i.e., h_k is the number of data vectors in class k, and f_{kil} is the number of data vectors in class k with variable X_i having value x_{il}. The evidence prediction formula is as follows:

$$\mathcal{P}_{\text{ev}}(Y = k|I, D) \propto \mathcal{P}_{\text{ev}}(\vec{d}[k], D) \propto \frac{\mathcal{P}_{\text{ev}}(\vec{d}[k], D)}{\mathcal{P}_{\text{ev}}(D)} = \frac{h_k + \mu_k}{N + \sum_{k'=1}^{K} \mu_{k'}} \prod_{i=1}^{m} \frac{f_{kil} + \sigma_{kil}}{h_k + \sum_{l=1}^{n_i} \sigma_{kil}}.$$

The $\mathcal{P}_{\text{ev}}$ formula can be derived using the results in [1]. The stochastic complexity predictive distribution is proportional to the likelihood of the combined data set $D^+ = D \cup \vec{d}[k]$:

$$\mathcal{P}_{\text{sc}}(Y = k|I, D) \propto P(\vec{d}[k], D|\tilde{\Theta}(\vec{d}[k], D)) = \prod_{k'=1}^{K} \left((\tilde{\alpha}_{k'})^{h^+_{k'}} \prod_{i=1}^{m} \prod_{l=1}^{n_i} (\tilde{\phi}_{k'il})^{f^+_{k'il}} \right),$$

where $\tilde{\alpha}_k = (h^+_k)/(N+1)$, $\tilde{\phi}_{kil} = f^+_{kil}/h^+_k$, and h^+_k and f^+_{kil} are the sufficient statistics of D^+.

4 Empirical results

In our experiments four public domain classification data sets[1] of varying size were used. Table 1 describes the size (N), the number of attributes ($m+1$), and the number of classes (K) for each of these data sets. Two separate sets of experiments were performed on each data set. In the first set of experiments we explored how the prediction quality of our various approaches depends on the size of the training set D. The second set of experiments allowed us to make a more detailed investigation on the differences in prediction quality between the three approaches for a fixed training set size.

4.1 Prediction performance and the training set size

In this set of experiments (Figures 1–3) we randomly partitioned each data set in a *training reservoir* containing 70% of the data instantiations and a *test set* containing the remaining 30%.

We now randomly took one data instantiation $\vec{d}_1$ out of the training reservoir and used it as our training set $D = (\vec{d}_1)$. We used D to generate the predictive distributions $\mathcal{P}_{\text{map}}(Y|I, D)$,$\mathcal{P}_{\text{ev}}(Y|I, D)$ and $\mathcal{P}_{\text{sc}}(Y|I, D)$ for all I's in the test set. We then compared the predictions thus obtained for each I to the actual outcomes k in a manner to be described below.

Next we extended D by another data instantiation $\vec{d}_2$, unequal to the element(s) already in D but otherwise randomly picked from the training reservoir. Again, for all I's in the training set the three predictive distributions were determined and again, the predictions thus obtained were compared to the actual outcomes k. We repeated this procedure of adding one training element to D, determining all predictive distributions using D and predicting the value of Y for each entry in the test set until D contained the full training reservoir.

Each prediction was evaluated by both log-score and 0/1-score. The *log-score* of a predictive distribution $\mathcal{P}(Y|I, D)$ is defined as $\log \mathcal{P}(Y = k|I, D_i)$ where k is the actual outcome of Y. For *0/1-score* we simply determine the k for which $\mathcal{P}(Y = y_k|I, D)$ is the highest and then predict Y to take on the value k. If the

[1] The data sets can be obtained from the UCI data repository at URL address "http://www.ics.uci.edu/~mlearn/".

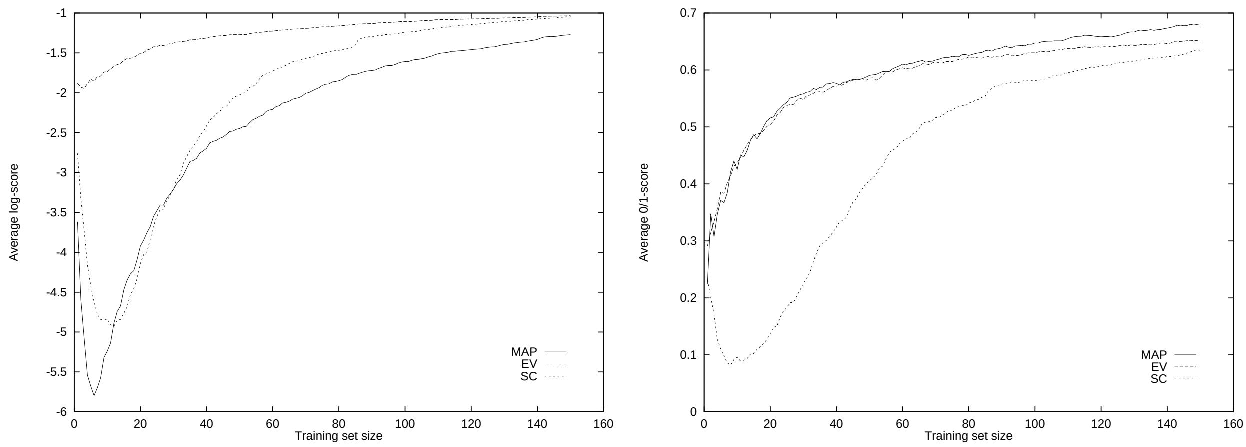

Figure 1: Average performance of methods by log-score (left) and 0/1-score (right) on the Glass database.

actual outcome is indeed k, then the 0/1-score is defined to be 1; if it is not equal to k, the 0/1-score is defined to be 0.

The whole procedure of partitioning the data set and successively predicting using larger and larger subsets of the training reservoir as our training data was repeated 100 times. In Figure 1 the performance of the three predictive distributions on the Glass database is shown for both log-score and 0/1-score. The vertical axis indicates the average score where the average is taken over the predictions of all class values in the test set and all 100 training sets of the size indicated on the horizontal axis.

We can see that both scores rapidly increase with all three methods used as the size of the training set increases. However, if we focus on very small sample sizes, we can make some interesting observations. For the log-score, $\mathcal{P}_{\rm ev}$ performs already nearly optimally, while both the MAP and the SC predictions show weak performance; furthermore, they only become competitive to the evidence prediction for much larger training set sizes. For the 0/1-score, the predictive stochastic complexity and the evidence still show the same behavior, while the MAP predictions tend to behave in a manner more similar to the evidence. Analogous, though sometimes less extreme, behavior was found for all of the four data sets used - the graphs for the log-score for the Australian and Hepatitis databases are shown in Figure 2.

Figures 1 and 2 are only concerned with averages over many training sets; this raises the question of how well the methods perform for individual training sets. In Figure 3 the log-score performance averages for $\mathcal{P}_{\rm ev}$ and $\mathcal{P}_{\rm sc}$ are shown together with the maximum and the minimum prediction performance for the Glass data set. For each training set size N, the maximum (minimum) performance is defined to be the prediction performance of the one training set out of the 100 training sets of size N that had the best (worst) performance on the test set. We see that after seeing about 20 data vectors (about 10% of the data), the worst case evidence prediction suddenly goes up. This "phase transition" behaviour also occurs for the stochastic complexity prediction, but only after about 80 data items. The corresponding graphs for the other three databases used and for the 0/1-score look very similar. The experiments suggest that for Naive Bayes models, the evidence with uniform priors can be a very safe predictor: even for small sample sizes it predicts well in most cases.

We can partially explain these results as follows: if one looks at the actual predictions made, the evidence prediction is much more 'conservative' than the MAP prediction. The latter is in our case equal to the Maximum Likelihood (ML) prediction, and 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. Let us consider a very simple example to illustrate this point. Suppose our data consists of a string of one's and zero's generated by some Bernoulli-process $p = P(X = 1)$. If we have seen an initial string consisting of just one '1', then the ML predictor will determine that the probability of the second symbol being a 1 is unity. However, using evidence prediction with uniform priors, this probability is $\frac{2}{3}$. If the next data item turns out to be a 0, then the log-score of the ML/MAP predictor will be $-\infty$ while that of the evidence will be $\log 2 - \log 3$.

The behaviour of the predictive stochastic complexity lies somewhere in between that of MAP and that

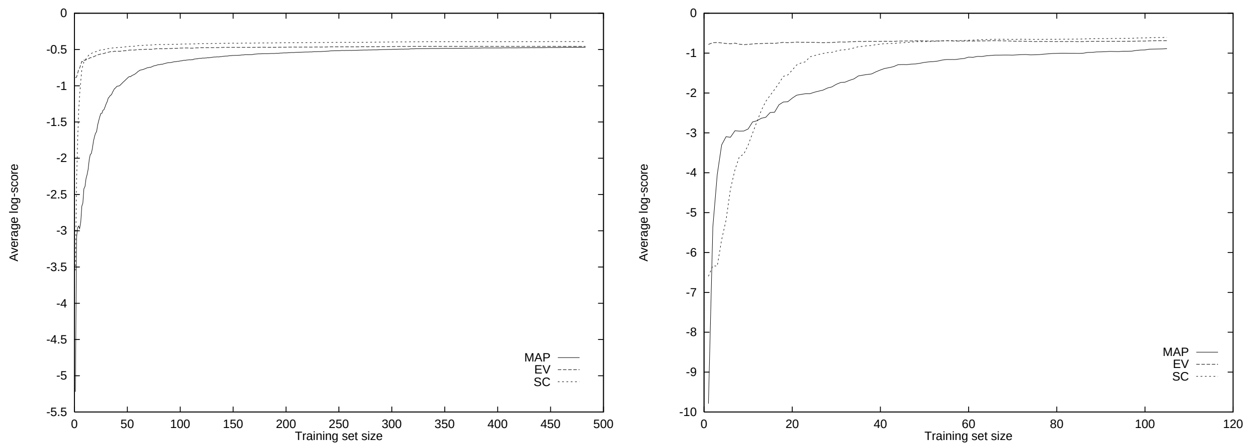

Figure 2: Average performance of methods by log-score on the Australian (left) and Hepatitis (right) databases.

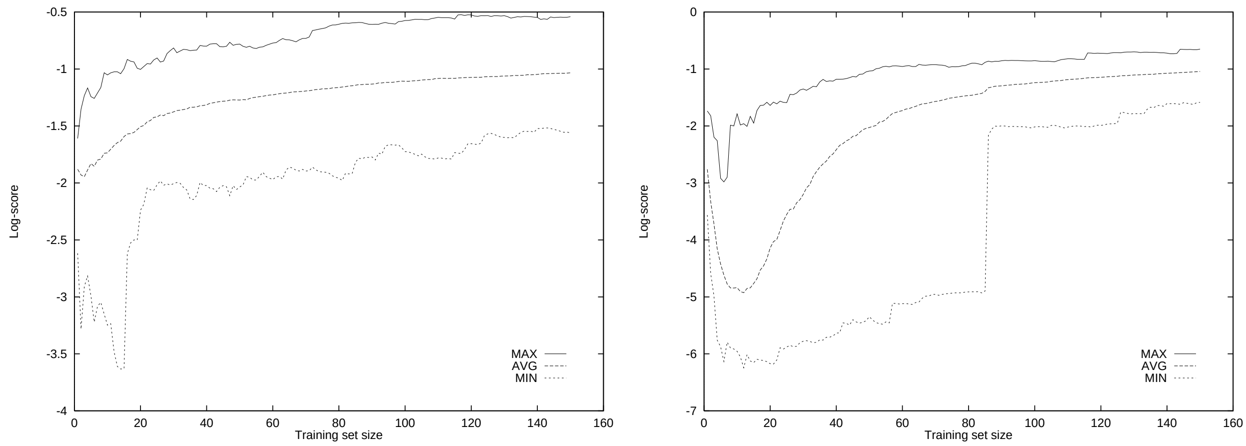

Figure 3: Performance of evidence (left) and stochastic complexity (right) by log-score on the Glass database.

of the evidence. In our example, the probability of the second symbol being a 1 will be $\frac{4}{5}$: the predictive stochastic complexity is less conservative than the evidence, but still more conservative than the MAP, which explains the small sample size behaviour for the log score.

4.2 Predictive performance with fixed training size

In our second set of experiments, we tested our three prediction methods using leave-one-out cross-validation, both for the log-score and the 0/1-score. As the training sets in leave-one-out CV are almost as large as the full data sets, Figures 1-3 already suggest that the methods will show similar performance. Our results on the four data sets are summarized in Table 1. The middle columns show the cross-validated results for the 0/1-score, the three rightmost columns show the results for the log-score. Though the differences in performances are all quite small, we see that for the log-score, the stochastic complexity performs consistently better than the evidence, which itself beats, at least on average, the MAP predictions. For the 0/1-score, the picture is not as clear-cut, but it seems that the MAP prediction performs slightly better than both the evidence and stochastic complexity predictions. One explanation for the latter 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 MAP 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 indeed receive maximum probability.

data set	size	attrs	classes	MAP-01	EV-01	SC-01	MAP-LS	EV-LS	SC-LS
Australian	690	15	2	0.851	0.848	0.852	-0.456	-0.457	-0.389
Glass	214	10	6	0.701	0.668	0.668	-1.216	-0.981	-0.954
Heart	270	14	2	0.830	0.837	0.844	-0.476	-0.439	-0.392
Hepatitis	150	20	2	0.847	0.820	0.813	-0.853	-0.666	-0.554

Table 1: Leave-one-out crossvalidation results on the four data sets used.

5 Conclusion and Future Work

It is well known that the Naive Bayes model studied here can be seen as a degenerated case of the more general family of finite mixtures of multinomials (see e.g.[4]). For finite mixture models the class variable Y is assumed to be a latent variable, the values of which are not given in the training data D. For this missing data case, the three predictive distributions described here can only be solved analytically by summing over all the possible instantiations of the missing data, which are exponential in number. Fortunately, there exists computationally efficient methods for approximating this exponential sum (see e.g., the discussion in [3]), so the three methods can be also applied in the approximative sense in the general finite mixture case. It is interesting to see whether the empirical results obtained here apply also for the finite mixture model family. Interestingly enough, although not the purpose of this paper, our empirical results show that the relatively simple Naive Bayes model used obtains very good prediction accuracy when compared to results obtained by alternative techniques (see the results referenced in [7]). Consequently, it would be interesting to see how the three prediction methods perform when used in conjunction with the more complex finite mixture model family. These research issues will be addressed in our future work.

Acknowledgements

This research has been supported by the ESPRIT Working Group 8556: Neural and Computational Learning (NeuroCOLT), the Technology Development Center (TEKES), and the Academy of Finland.

References

[1] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[2] M.H. DeGroot. *Optimal statistical decisions.* McGraw-Hill, 1970.

[3] P. Kontkanen, P. Myllymäki, and H. Tirri. Comparing Bayesian model class selection criteria by discrete finite mixtures. In D. Dowe, K. Korb, and J. Oliver, editors, *Information, Statistics and Induction in Science (Proceedings of the ISIS'96 Conference)*, pages 364–374, Melbourne, Australia, August 1996. World Scientific, Singapore.

[4] P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report C-1996-9, University of Helsinki, Department of Computer Science, February 1996.

[5] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Company, New Jersey, 1989.

[6] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, January 1996.

[7] H. Tirri, P. Kontkanen, and P. Myllymäki. Probabilistic instance-based learning. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 507–515. Morgan Kaufmann Publishers, 1996.