On Predictive Distributions and Bayesian Networks

P. Kontkanen (pkontkan@cs.helsinki.fi),
P. Myllymäki (myllymak@cs.helsinki.fi),
T. Silander (tsilande@cs.helsinki.fi) and
H. Tirri (tirri@cs.helsinki.fi)
Complex Systems Computation Group (CoSCo)[*]
P.O.Box 26, Department of Computer Science
FIN-00014 University of Helsinki, Finland

P. Grünwald (grunwald@cs.stanford.edu)
Department of Computer Science
Stanford University
Stanford, CA 94306, USA

Abstract. In this paper we are interested in discrete prediction problems for a decision-theoretic setting, where the task is to compute the predictive distribution for a finite set of possible alternatives. This question is first addressed in a general Bayesian framework, where we consider a set of probability distributions defined by some parametric model class. Given a prior distribution on the model parameters and a set of sample data, one possible approach for determining a predictive distribution is to fix the parameters to the instantiation with the *maximum a posteriori* probability. A more accurate predictive distribution can be obtained by computing the *evidence* (*marginal likelihood*), i.e., the integral over all the individual parameter instantiations. As an alternative to these two approaches, we demonstrate how to use Rissanen's new definition of *stochastic complexity* for determining predictive distributions, and show how the evidence predictive distribution with Jeffreys' prior approaches the new stochastic complexity predictive distribution in the limit with increasing amount of sample data. To compare the alternative approaches in practice, each of the predictive distributions discussed is instantiated in the Bayesian network model family case. In particular, to determine Jeffreys' prior for this model family, we show how to compute the (expected) Fisher information matrix for a fixed but arbitrary Bayesian network structure. In the empirical part of the paper the predictive distributions are compared by using the simple tree-structured Naive Bayes model, which is used in the experiments for computational reasons. The experimentation with several public domain classification datasets suggest that the evidence approach produces the most accurate predictions in the log-score sense. The evidence-based methods are also quite robust in the sense that they predict surprisingly well even when only a small fraction of the full training set is used.

Keywords: Bayesian Networks, Predictive Inference, MDL, MML, Jeffreys' prior

[*] http://www.cs.Helsinki.FI/research/cosco/

1. Introduction

In discrete prediction problems the task is to select one action from a finite set of possible alternatives. All possible outcomes, corresponding to the set of possible actions given, result in some gain or utility, the value of which depends on the correct (but unknown) action in the decision problem in question. From the decision-theoretic point of view (see e.g. [2]), the optimal procedure in this case is to choose the action with the *maximal expected utility*. To be able to maximize the expected utility, one needs to determine the *predictive distribution* for all the possible actions, by using the problem domain probability distribution. In real-life situations, however, the problem domain probability distribution is not known explicitly, and it has to be estimated from sample data and (possibly) some prior information. In this paper our purpose is to compare different alternatives for computing the predictive distribution in this context.

Here we will assume that the probability distributions to be considered are restricted to a limited set of discrete distributions defined by some fixed parametric model form. Given some sample data (the *training data*), and an incomplete *query vector*, where the values of some of the problem domain variables are not given, the task is to compute the predictive distribution for the missing part of the query vector. In the *maximum a posteriori (MAP)* approach, the predictive distribution is determined by using the model (i.e., parameter instantiation) with the highest posterior probability, given the training data and a prior distribution for the parameters. In the *evidence* approach, the predictive distribution is obtained by integrating over all the possible parameter instantiations, in other words, over all the distributions representable by the chosen model form.

As pointed out in [26], in information-theoretic terms, minus the logarithm of the evidence integral can be regarded as a formalization of *stochastic complexity (SC)*, i.e., the shortest possible codelength for coding the data with respect to the chosen model form. Recently Rissanen [27] has 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 other cases. In the third approach considered here, we define the predictive distribution by using Rissanen's new definition of stochastic complexity. A recent, comprehensive tutorial to the related general *Minimum Description Length (MDL)* theory and its application to predictive inference can be found in [15]. The similarities and differences between MDL and another information-theoretical framework, the *Minimum Message Length (MML)* approach [32, 33], are discussed in [1, 16].

The discrete decision problem discussed in this paper, together with the MAP, the evidence and the stochastic complexity predictive distributions for solving this problem are described formally in Section 2. In Section 3, we apply these general results to the special case where the problem domain distributions are assumed to be specified by using *Bayesian network* models (see, e.g, [24, 23]), and show how to define each of the above mentioned predictive distributions for a given Bayesian network structure. This can be seen as an extension of the work reported in [19], where the problem was studied in the more limited Naive Bayes classifier context. Furthermore, in addition to the standard case with uniform prior distribution, we discuss here also the use of Jeffreys' prior for computing the evidence predictive distribution. The case with Jeffreys' prior distribution is particularly interesting as it can be shown that with this prior the evidence predictive distribution approaches the stochastic complexity predictive distribution when the sample size increases [27].

The formulas for computing Jeffreys' prior in the Bayesian network model family case are given in Section 3.5. At this point it should be emphasized that although the *observed* information matrix for Bayesian networks has been presented earlier [30], derivation of the *expected* Fisher information required for computing Jeffreys' prior is one of the novel contributions of this paper. It should also be noted that the derivation of the expected Fisher information presented in [34, 35] is of no relevance here, as it assumes a simple linear dependency model with continuous zero-mean variables (and Gaussian zero-mean noise), while we address the discrete variable case, and allow much more general dependencies represented by a Bayesian network with Multinomial-Dirichlet local dependency models.

Although the general forms for computing the different predictive distributions for Bayesian networks are described in Section 3, there remain some computational problems if the methods are to be used in practice. First of all, in the general setting, determining the predictive distributions requires computing over all possible outcomes of the unset variables, which is clearly a computationally infeasible task if the number of unset variables is high. Secondly, using Jeffreys' prior as formulated here requires computing the marginal distribution for the parents of each node in the Bayesian network. As this problem is known to be NP-hard for multi-connected Bayesian network structures [8], determining Jeffreys' prior may be computationally difficult in practice. However, the standard probabilistic reasoning algorithms (see, e.g., [24, 23, 18, 5]) found in most Bayesian network software packages could in most practical cases be adopted for solving these problems, and determining Jeffreys' prior for a Bayesian network model is hence computationally no harder than actually using the model (for predictive inference). Nevertheless, to simplify our already extensive empirical setup, we decided in the experimental part of the paper to focus on the computationally simple Naive Bayes classifier case, where the predictive inference task in question is a simple classification problem, and the number of possible outcomes is equal to the number of possible classes. The Bayesian network model to be used is in this case a simple tree, and determining Jeffreys' prior is easy, as demonstrated in Section 4.1.

In Section 4.2, the predictive accuracy of the maximum likelihood predictive distribution (the MAP predictive distribution with uniform prior distribution), the evidence predictive distribution (with both uniform and Jeffreys' prior), and the stochastic complexity predictive distribution are evaluated empirically in the Naive Bayes case with publicly available classification data sets. A similar study was reported in [12], but the results were obtained by using only a simple Bernoulli model, and the focus on the experiments was in model selection (hypothesis testing and segmentation), rather than in predictive accuracy. The empirical setup used in this paper follows the methodology presented in [19], but the number of data sets used in this study is considerably higher. In addition, we have used here a more versatile empirical validation regime exploiting several variants of the crossvalidation method for measuring the accuracy of different predictive methods. In our second set of experiments (Section 4.3), we examined how the predictive accuracies of the different approaches depend on the amount of the sample data available. The results are summarized and discussed in Section 4.4.

2. Predictive distributions for discrete decision problems

2.1. The decision problem

In this paper we model our problem domain by a set X of m discrete random variables, $\mathrm{X} = \{X_1, \ldots, X_m\}$, where a random variable X_i can take on any of the values in the set $\mathcal{X}_i = \{x_{i1}, \ldots, x_{in_i}\}$. A *data instantiation* $\mathbf{d} = (x_1, \ldots, x_m)$ is a vector in which all the variables X_i have been assigned a value: by $\mathbf{X} = \mathbf{d}$ we mean that $X_1 = x_1, \ldots, X_m = x_m$, where $x_i \in \mathcal{X}_i$. Let $\mathcal{D}^1$ be the set of all possible data instantiations $\mathbf{d}$: $\mathcal{D}^1 = \mathcal{X}_1 \times \ldots \times \mathcal{X}_m$. A *random sample* $D = (\mathbf{d}_1, \ldots, \mathbf{d}_N)$ is a set of N i.i.d. (independent and identically distributed) data instantiations, where each $\mathbf{d}_j$ is sampled from $\mathcal{P}$, the joint distribution of the variables in X. Hence samples of length N are distributed according to $\mathcal{P}^N$, the N-fold product distribution of $\mathcal{P}$. Whenever this cannot lead to any confusion, we drop in the sequel the superscript N. More precisely, let

$\mathcal{D}^N$ be the set of all samples D of length N. For all $N > 1$ and any $D = (\mathbf{d}_1, \ldots, \mathbf{d}_N) \in \mathcal{D}^N$, $\mathcal{P}^N(D) = \mathcal{P}(D) = \prod_{i=1}^{N} \mathcal{P}(\mathbf{d}_i)$.

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

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

We focus on the following prediction problem: Given the values of some of the variables in $\mathbf{d}$, and some training data D, we want to use the training data to arrive at good predictions for the values of the rest of the variables. More precisely, given D and the values of a subset U (the *clamped* variables) of the variables in X, we wish to predict the values of the variables in the set $\mathrm{V} = \mathrm{X} \setminus \mathrm{U}$ (the *free* variables). We will do our predictions by determining probabilities for each of the possible instantiations of V.

Without loss of generality we may assume that the variables are indexed in such a way that we can write $\mathrm{U} = \{X_1, \ldots, X_k\}$, $\mathrm{V} = \{X_{k+1}, \ldots, X_m\}$. In the sequel, we will use $\mathbf{U} = \mathbf{u}$ as an abbreviation for a partial data instantiation $X_1 = u_1, \ldots, X_k = u_k$ where $u_i \in \mathcal{X}_i$. Similarly, $\mathbf{V} = \mathbf{v}$ stands for $X_{k+1} = v_{k+1}, \ldots, X_m = v_m$ where $v_i \in \mathcal{X}_i$. Hence each data instantiation $\mathbf{d}$ can be written as $\mathbf{d} = (\mathbf{u}, \mathbf{v}) = (u_1, \ldots, u_k, v_{k+1}, \ldots, v_m)$ for some $\mathbf{u}$ and $\mathbf{v}$. Note that $u_{k+1}, \ldots, u_m$ and $v_0, \ldots v_k$ remain undefined in our notation. We can now state our aim more precisely as follows: we wish to compute, for all possible $\mathbf{v}$, the probabilities

$$\mathcal{P}(\mathbf{V} = \mathbf{v} \mid \mathbf{U} = \mathbf{u}, D = (\mathbf{d}_1, \ldots, \mathbf{d}_N)) \tag{2}$$

We will abbreviate (2) to $\mathcal{P}(\mathbf{v} \mid \mathbf{u}, D)$. Using the basic rules of probability theory we can write

$$\mathcal{P}(\mathbf{v}|\mathbf{u}, D) = \frac{\mathcal{P}(\mathbf{u}, \mathbf{v}, D)}{\mathcal{P}(\mathbf{u}, D)} = \frac{\mathcal{P}((\mathbf{u}, \mathbf{v})|D)}{\sum_{\mathbf{v}} \mathcal{P}((\mathbf{u}, \mathbf{v})|D)}, \tag{3}$$

where the summing goes over all possible instantiations of the variables in the set V.

Consequently, we see that the conditional distribution for the variables in V can be computed by using the complete data vector conditional distributions (1) for each of the possible complete vectors $\mathbf{d} = (\mathbf{u}, \mathbf{v})$. The resulting distribution (2) is called the *predictive distribution* of the variables in V. It should be noted that the number of terms in the summation over possible $\mathbf{v}$ grows exponentially with the number of variables in V; therefore, predicting the values of many variables given the values of only a few may be difficult. However, in many cases of practical interest we only want to predict the values of a very small number (in classification problems just one) of the variables, so the set V can be very small.

In practice the "true" problem domain probability distribution $\mathcal{P}$ is not known, and it has to be approximated by using the sample D. We restrict the search for a good approximation of $\mathcal{P}$ to some parametric family $\mathcal{M}$ of probabilistic models. Here each model in $\mathcal{M}$ is a probability distribution over $\mathcal{D}^1$ determined by an instantiation of parameters Θ, i.e., each model defines a probability $P(\mathbf{d}|\Theta)$ for each possible data instantiation $\mathbf{d}$. We can thus identify each model with a particular parameter instantiation Θ, and write $\Theta \in \mathcal{M}$ to denote the model that is determined by Θ. In sections 2.2–2.4 we describe four different approaches for approximating $\mathcal{P}$. The first two of these are the *MAP* and the *evidence* approximations, which are both standard methods of Bayesian statistics [2] and therefore reviewed here only very briefly. The third and fourth approximations we discuss are based on information-theoretic considerations. More specifically, they involve Rissanen's renewed [27] definition of *stochastic complexity*. Since the derivation of these approximations of $\mathcal{P}$ is part of our own theoretical contribution in this paper, we discuss them in considerable detail.

2.2. The Bayesian predictive distributions $\mathcal{P}_{\text{MAP}}$ and $\mathcal{P}_{\text{EV}}$

Given a prior distribution $P(\Theta)$ defined for all $\Theta \in \mathcal{M}$, we can arrive at a posterior distribution $P(\Theta|D)$ by using Bayes' rule:

$$P(\Theta|D) \propto P(D|\Theta)P(\Theta). \tag{4}$$

In the *maximum a posteriori (MAP) probability* approximation, the distribution $\mathcal{P}$ in Equation (1) is replaced by the distribution corresponding to the single model $\hat{\Theta}(D)$ maximizing the posterior distribution (4),

$$\hat{\Theta}(D) = \arg\max_{\Theta} P(\Theta|D).$$

The corresponding predictive distribution is

$$\mathcal{P}_{\text{MAP}}(\mathbf{d} \mid D) = P(\mathbf{d} \mid D, \hat{\Theta}(D)) \stackrel{i.i.d.}{=} P(\mathbf{d} \mid \hat{\Theta}(D)). \tag{5}$$

If we assume the prior distribution $P(\Theta)$ to be uniform, then (4) becomes $P(\Theta|D) \propto P(D|\Theta)$ and the MAP model becomes equal to the *Maximum Likelihood (ML) model* of classical frequentist (non-Bayesian) statistics: the model that maximizes the data likelihood $P(D|\Theta)$. In Section 2.3 we see that the notion of the maximum likelihood model is in the central part in our information-theoretic approach for defining predictive distributions.

If, instead of using a single model $\hat{\Theta}$, we average (integrate) over all the models $\Theta \in \mathcal{M}$, we get a more sophisticated approximation of $\mathcal{P}$. In the Bayesian literature the corresponding integral is called the *evidence* or *marginal likelihood*, and it is given by

$$\mathcal{P}_{\mathrm{EV}}(\mathbf{d}, D) = \int P(\mathbf{d}, D|\Theta)P(\Theta)d\Theta, \tag{6}$$

where the integration goes over all the models Θ in $\mathcal{M}$. The resulting predictive distribution (1) then becomes

$$\mathcal{P}_{\mathrm{EV}}(\mathbf{d} \mid D) = \int P(\mathbf{d} \mid D, \Theta)P(\Theta \mid D)d\Theta. \tag{7}$$

2.3. The information-theoretic predictive distribution $\mathcal{P}_{\mathrm{SC}}$

2.3.1. *The MDL principle and stochastic complexity*

Stochastic Complexity is a central concept in the *Minimum Description Length (MDL) Principle* [26, 27]. According to MDL, the goal of all inductive inference is to *compress* the given data as much as possible, i.e. to describe it using as few bits as possible. This involves the use of a description method or *code*, which is a one–one mapping from datasets to their descriptions. Without loss of generality, these descriptions may be taken to be binary strings [26]. Intuitively, the shorter the description or codelength of a set of D, the more regular or simpler the set D is. Rissanen [25] introduced the stochastic complexity as follows:

> The stochastic complexity of the data set D with respect to the model class $\mathcal{M}$ is the shortest code length of D obtainable when the encoding is done with the help of class $\mathcal{M}$ [25, 27].

Here 'with the help of' has a clear *intuitive* meaning: if there exists a model in $\mathcal{M}$ which captures the regularities in D well, or equivalently *gives a good fit* to D, then the code length of D should be short. However, it turns out to be very hard to define 'with the help of' *in a formal manner*. Indeed, a completely satisfactory formal definition has only been found very recently [27] — before 1996, Rissanen used the evidence (6) for defining the stochastic complexity, but this earlier mathematical definition he now regards merely as an approximation of the new one. We discuss the new definition in some detail.

Note that the informal definition of stochastic complexity (SC) as given above presumes the existence of a code: by definition, the SC of a data set D is the length of the encoding of D where the encoding is done using some special code C^* which gives 'the shortest possible codelengths with respect to $\mathcal{M}$'. In order to introduce a formula for the codelengths obtained using this C^*, we now first have to clarify the connection between probability distributions and codes.

In general, we denote the length (in bits) of the encoding of D when the encoding is done using a code C by $L_C(D)$. All codes considered in MDL are *prefix* codes [26]. From the Kraft inequality (see for example [26] or [10]) it follows that for every (complete) prefix code C, there exists a corresponding probability distribution P such that for all data sets D of given length N (i.e., with N data instantiations), we have $-\log P(D) = L_C(D)$ (throughout this paper, by 'log' we denote logarithm to the base two). Similarly, for every probability distribution P defined over all data sets D of length N there exists a code C such that for all datasets D of length N, we have $L_C(D) = \lceil -\log P(D) \rceil$ (here $\lceil x \rceil$ is the smallest integer greater or equal to x). If we use $-\log P(D)$ instead of $\lceil -\log P(D) \rceil$, our code lengths will always be less than one bit off the mark; we may therefore safely neglect the integer requirement for code lengths [25]. Once we have done this, the two facts above imply that we can interpret any probability distribution over sequences of a given length as a code and vice versa. This correspondence allows us to *identify* codes and probability distributions: every probability distribution P over data sets of length N may equivalently be interpreted as determining a code C such that $L_C(D) = -\log P(D)$ for all D of length N. We see that a short code length corresponds to a high probability and vice versa: whenever $P(D) > P(D')$, we have $-\log P(D) < -\log P(D')$.

If our parametric class of models $\mathcal{M}$ is regular enough (as it will indeed be for all instantiations of $\mathcal{M}$ we consider in this paper), then there exists a *maximum likelihood (ML) estimator* $\tilde{\Theta}$ for every data set D, and we can write:

$$\begin{aligned} \tilde{\Theta}(D) = \arg\max_{\Theta \in \mathcal{M}} P(D|\Theta) &= \arg\min_{\Theta \in \mathcal{M}} -\log P(D|\Theta) \\ &= \arg\min_{\Theta \in \mathcal{M}} L(D|\Theta), \end{aligned} \tag{8}$$

where the last equality indicates the fact that each Θ defines a code such that the code length of D is given by $-\log P(D|\Theta)$. Since this term can be interpreted as a code length, we abbreviate it to $L(D|\Theta)$.

Let us now consider a data set D of arbitrary but fixed length N. The MDL Principle tells us to look for a short encoding of D. The model within class $\mathcal{M}$ that compresses the data most is the ML model $\tilde{\Theta}(D)$, since by (8) it is the model for which $L(D|\Theta)$, the codelength of D when encoded with (the code corresponding to) Θ, is lowest. At first sight it *seems* that we should code our data D using $\tilde{\Theta}(D)$, in which case the MDL Principle would reduce to the maximum likelihood method of classical statistics. However — and this is the crucial observation which makes MDL very different from ML — MDL says that we must code our data using some *fixed* code, which compresses *all* data sets for which there is a good-fitting model in $\mathcal{M}$ [25]. But the code corresponding to $\tilde{\Theta}(D)$, i.e. the code that encodes any D' using $L(D'|\tilde{\Theta}(D)) = -\log P(D'|\tilde{\Theta}(D))$ bits, only gives optimal compression for *some* data sets (among which include D). For most other data sets $D' \neq D$, $\tilde{\Theta}(D)$ will definitely not be optimal: if we had been given such a different data set D' (also of length N) instead of D, then the code corresponding to $\tilde{\Theta}(D')$ rather than $\tilde{\Theta}(D)$ would give us the optimal compression. In general, coding D' using $\tilde{\Theta}(D)$ (i.e. using $L(D'|\tilde{\Theta}(D))$ bits) may be very inefficient.

We repeat the crucial observation: MDL says that we must code our data using some *fixed* code, which compresses *all* data sets that are well modeled by $\mathcal{M}$. We can therefore not use the code based on $\tilde{\Theta}(D)$ if our data happens to be D and the code based on $\tilde{\Theta}(D')$ if our data happens to be D': we would then encode D using a different code than when encoding D'. It would thus be very desirable if we could come up with a code that compresses each possible D as well as the maximum-likelihood, or equivalently, mostly-compressing element in $\mathcal{M}$ for that specific D. In other words, we would like to have a single code C_1 such that $L_{C_1}(D) = L(D|\tilde{\Theta}(D))$ for all possible D. However, such a code cannot exist as soon as our model class contains more than one element, since in general a code can only give short codelengths to a very limited number of data instantiations. Nevertheless, it *is* possible to construct a code C_2 such that

$$L_{C_2}(D) = -\log P(D|\tilde{\Theta}(D)) + K_N = L(D|\tilde{\Theta}(D)) + K_N \tag{9}$$

for all D of length N. Here K_N is a constant that may depend on N but is equal for all D of length N. If, for some $\Theta \in \mathcal{M}$, we say that it fits the data D well, we mean that the probability $P(D|\Theta)$ is high. Note that the code length obtained using C_2 precisely reflects for each D how well D is fitted by the model in the class that fits D best.

Picking C_2 such that the constant K_N is as small as possible yields the most efficient code that satisfies (9). We call the resulting code the *stochastic complexity code* and denote it by C^*. The corresponding minimal K_N is denoted by K_N^* and is called the *model cost* of $\mathcal{M}$. We define the code length of D when encoded using this code to be the *stochastic complexity of D with respect to model class $\mathcal{M}$* which we write as $S(D|\mathcal{M})$:

$$S(D|\mathcal{M}) = L_{C^*}(D) = L(D|\tilde{\Theta}(D)) + K_N^*, \tag{10}$$

where $\tilde{\Theta}(D) \in \mathcal{M}$.

2.3.2. *The stochastic complexity predictive distribution* $\mathcal{P}_{\mathrm{SC}}$

The aforementioned correspondence between probabilities and code lengths implies that there exists a probability distribution $\mathcal{P}_{\mathrm{SC}}^{N}$ such that for all D of length N, $-\log \mathcal{P}_{\mathrm{SC}}^{N}(D) = L_{C^*}(D)$. We call this $\mathcal{P}_{\mathrm{SC}}^{N}$ the *stochastic complexity predictive distribution*. Just like C^* is the code that gives the shortest possible codelength to those data sets for which there exists a good-fitting model in $\mathcal{M}$, $\mathcal{P}_{\mathrm{SC}}^{N}$ is the distribution that gives as much probability as possible to those data sets for which there exists a good-fitting model in $\mathcal{M}$. This motivates the use of $\mathcal{P}_{\mathrm{SC}}$ for prediction.

From (10) we have

$$S(D|\mathcal{M}) = -\log \mathcal{P}_{\mathrm{SC}}^{N}(D) = -\log P(D|\tilde{\Theta}(D)) + K_N^*, \tag{11}$$

where $\tilde{\Theta}(D) \in \mathcal{M}$. Since $\mathcal{P}_{\mathrm{SC}}^{N}(D)$ is a probability distribution, and hence $\sum_{D\in\mathcal{D}^N} \mathcal{P}_{\mathrm{SC}}^{N}(D) = 1$, we see from (11) that $K_N^* = \log \sum_{D\in\mathcal{D}^N} P(D|\tilde{\Theta}(D))$, or, equivalently:

$$\mathcal{P}_{\mathrm{SC}}^{N}(D) = \frac{P(D|\tilde{\Theta}(D))}{\sum_{D\in\mathcal{D}^N} P(D|\tilde{\Theta}(D))} = (F_N)^{-1} \cdot P(D|\tilde{\Theta}(D)), \tag{12}$$

where $F_N = \sum_{D\in\mathcal{D}^N} P(D|\tilde{\Theta}(D))$.

From a practical point of view, using (12) as the predictive distribution may at first sight seem infeasible since computing the normalizing sum F_N involves summing over an exponential number of terms (one for each data instantiation). Nevertheless, it is easy to see that the problem disappears if one computes a predictive distribution for a dataset D of length N in the following straightforward manner:

$$\begin{aligned}
\mathcal{P}_{\mathrm{SC}}^{N+1}(\mathbf{d} \mid D) &= \frac{\mathcal{P}_{\mathrm{SC}}^{N+1}(\mathbf{d}, D)}{\mathcal{P}_{\mathrm{SC}}^{N+1}(D)} = \frac{\mathcal{P}_{\mathrm{SC}}^{N+1}(\mathbf{d}, D)}{\sum_{\mathbf{d}'} \mathcal{P}_{\mathrm{SC}}^{N+1}(\mathbf{d}', D)} \\
&= \frac{P(\mathbf{d}, D \mid \tilde{\Theta}(\mathbf{d}, D)) \cdot F_{N+1}^{-1}}{\sum_{\mathbf{d}'} P(\mathbf{d}', D \mid \tilde{\Theta}(\mathbf{d}', D)) \cdot F_{N+1}^{-1}} \\
&= \frac{P(\mathbf{d}, D \mid \tilde{\Theta}(\mathbf{d}, D))}{\sum_{\mathbf{d}'} P(\mathbf{d}', D \mid \tilde{\Theta}(\mathbf{d}', D))} \\
&\overset{i.i.d}{=} \frac{P(\mathbf{d} \mid \tilde{\Theta}(\mathbf{d}, D)) P(D \mid \tilde{\Theta}(\mathbf{d}, D))}{\sum_{\mathbf{d}'} P(\mathbf{d}' \mid \tilde{\Theta}(\mathbf{d}', D)) P(D \mid \tilde{\Theta}(\mathbf{d}', D))}.
\end{aligned} \tag{13}$$

Although this formula looks somewhat similar to that of the MAP (ML) predictor (5), it should be noted that the probabilities $P(D|\tilde{\Theta}(\mathbf{d}, D))$ do not cancel out here since the maximum likelihood estimator appearing in the denominator of (13) depends on $\mathbf{d}'$ and hence is not a constant. Moreover, the maximum likelihood estimator $\tilde{\Theta}(\mathbf{d}, D)$ is now computed by using the data set $D \cup \mathbf{d}$, not just D.

2.4. Connecting $\mathcal{P}_{\text{EV}}$ and $\mathcal{P}_{\text{SC}}$: the $\mathcal{P}_{\text{EVJ}}$ predictive distribution

2.4.1. $\mathcal{P}_{\text{SC}}$ *is not a random process*

A sequence of probability distributions $\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3 \ldots$, where $\mathcal{P}^i$ is a distribution over $\mathcal{D}^i$, is a *random process* if for all $N > 0$, $D \in \mathcal{D}^N$, we have (see for example [26]):

$$\sum_{\mathbf{d} \in \mathcal{D}^1} \mathcal{P}^{N+1}(D, \mathbf{d}) = \mathcal{P}^N(D). \tag{14}$$

In such a case, the sequence of distributions may be interpreted as one single distribution over the event space of all finite data sets. This means that the data can be interpreted as arriving sequentially. It is very easy to show that the evidence predictive distribution $\mathcal{P}_{\text{EV}}$ has property (14) for all i.i.d. model classes $\mathcal{M}$; that is the reason why we may omit the superscript N and write $\mathcal{P}_{\text{EV}}$ instead of $\mathcal{P}_{\text{EV}}^N$. However, the stochastic complexity predictive distribution $\mathcal{P}_{\text{SC}}$ does *not* have this property. To show this, we give a very simple example. Suppose our set of random variables contains just one element: $\text{X} = \{X_1\}$ and our model class $\mathcal{M}$ contains the i.i.d. Bernoulli models for X_1: $\mathcal{M} = \{P(\cdot|\Theta) \mid 0 \leq \Theta \leq 1\}$ such that $P(X_1 = 1|\Theta) = \Theta$ and for any $\mathbf{d}$ and D, $P(\mathbf{d}, D|\Theta) = P(\mathbf{d}|\Theta)P(D|\Theta)$. We see from equation (12) that

$$\begin{aligned} \mathcal{P}_{\text{SC}}^2(1,1) = & \quad \tfrac{1}{2\cdot 1 + 2\cdot 1/4} & = 0.4000, \text{ while} \\ \sum_{\mathbf{d} \in \{0,1\}} \mathcal{P}_{\text{SC}}^3(1,1,\mathbf{d}) = & \quad \tfrac{1+(2/3)^2(1/3)}{2\cdot 1+6\cdot(2/3)^2(1/3)} & = 0.3974. \end{aligned}$$

The fact that the sequence of distributions $\mathcal{P}_{\text{SC}}^N$ does not define a random process has a problematic consequence. In many practical learning situations, the number of data instantiations the learner receives is not determined beforehand; one usually has an initial training set $D = (\mathbf{d}_1, \ldots, \mathbf{d}_N)$, which is then used to predict the variables in V for one *or more* future data elements $\mathbf{d}_{N+1}, \mathbf{d}_{N+2}, \ldots$. Strictly speaking, the fact that $\mathcal{P}_{\text{SC}}$ does not define a random process causes the predictive distribution $\mathcal{P}_{\text{SC}}^N$ as defined in (13) to be valid only for the situation where we know that the learner will be confronted with just *one* test vector $\mathbf{d}$, after which the prediction process will stop for ever. To see this, let us return to our example and see what happens if, like before, we want to find the predictive distribution for $\mathbf{d}$ given D of length N, but now we also assume that afterwards, a new vector $\mathbf{d}_+$ will arrive.

Rewriting in the same manner as in (13), we get:

$$\begin{aligned}
\mathcal{P}_{\mathrm{SC}}^{N+2}(\mathbf{d} \mid D) &= \frac{\mathcal{P}_{\mathrm{SC}}^{N+2}(\mathbf{d}, D)}{\sum_{\mathbf{d}'} \mathcal{P}_{\mathrm{SC}}^{N+2}(\mathbf{d}', D)} = \frac{\sum_{\mathbf{d}_+} \mathcal{P}_{\mathrm{SC}}^{N+2}(\mathbf{d}, \mathbf{d}_+, D)}{\sum_{\mathbf{d}'} \sum_{\mathbf{d}_+} \mathcal{P}_{\mathrm{SC}}^{N+2}(\mathbf{d}', \mathbf{d}_+, D)} \\
&= \frac{\sum_{\mathbf{d}_+} P(\mathbf{d}, \mathbf{d}_+, D \mid \tilde{\Theta}(\mathbf{d}, \mathbf{d}_+, D))}{\sum_{\mathbf{d}'} \sum_{\mathbf{d}_+} P(\mathbf{d}', \mathbf{d}_+, D \mid \tilde{\Theta}(\mathbf{d}', \mathbf{d}_+, D))}. \qquad (15)
\end{aligned}$$

This is, however, in general *not* equal to (13). The reader may verify this by returning to our little example:

$$\mathcal{P}_{\mathrm{SC}}^{2}(1|1) = \frac{4}{5} = 0.8000 \text{ while } \mathcal{P}_{\mathrm{SC}}^{3}(1|1) = \frac{31}{39} = 0.7949$$

Hence if we assume that more future data will be available at some point, then we have to make different predictions for the first new data vector $\mathbf{d}$! Nevertheless, as we will see below, for large N the difference in the predictive probabilities between $\mathcal{P}_{\mathrm{SC}}^{N+1}(\mathbf{d}|D), \mathcal{P}_{\mathrm{SC}}^{N+2}(\mathbf{d}|D), \ldots$ will tend to zero. For this reason, we decided in our experiments to use $\mathcal{P}_{\mathrm{SC}}$ as defined in (13). It should also be noted that we concentrate in our experiments on the leave-one-out crossvalidation setup, in which case the test set indeed always contains only one test vector $\mathbf{d}$.

2.4.2. *A random process that approximates* $\mathcal{P}_{\mathrm{SC}}$

We see that although the $\mathcal{P}_{\mathrm{SC}}$ predictive distribution has several nice properties from the information-theoretic point of view, it may still not be well-suited for most prediction tasks. On the other hand, the theoretical results on predictive MDL [26] imply that the more (the code corresponding to) a random *process* P compresses the data D, the better we can predict properties of D using the predictive distribution based on P. This means that we should look for the random process that compresses all data D for which there is a good-fitting model in $\mathcal{M}$ as much as possible. Since the probability distribution (not process) which does this is given by $\mathcal{P}_{\mathrm{SC}}$, we may restate our aim as follows: we look for the random process that best approximates $\mathcal{P}_{\mathrm{SC}}$.

Rissanen [27] proved a fundamental theorem which, together with the results in [6, 29] implies the following: under certain reasonable regularity conditions on the model class $\mathcal{M}$, we have

$$-\log \mathcal{P}_{\mathrm{SC}}^{N}(\mathbf{d}_1, \ldots, \mathbf{d}_N) = -\log \mathcal{P}_{\mathrm{EVJ}}(\mathbf{d}_1, \ldots, \mathbf{d}_N) + o(1). \qquad (16)$$

Here, by definition, $\lim_{N\to\infty} o(1) = 0$, and $\mathcal{P}_{\mathrm{EVJ}}$ denotes the evidence distribution as given by (6) with the prior instantiated to the so-called *Jeffreys' prior*. If Jeffreys' prior is proper (as it will turn out to be for the model class of Bayesian networks), then it is given by (see for example [27] or [2]):

$$\pi(\Theta) = \frac{|I(\Theta)|^{1/2}}{\int |I(\eta)|^{1/2} d\eta}. \tag{17}$$

Here $|I(\Theta)|$ is the determinant of the *Fisher (expected) information matrix* $I(\Theta)$. Denoting $\Theta = (\theta_1, \ldots, \theta_r)$, entry (i, j) of matrix $I(\Theta)$ is defined as

$$[I(\Theta)]_{i,j} = -E_\Theta \left[\frac{\partial^2 \log P(\mathbf{X}|\Theta)}{\partial\theta_i \partial\theta_j} \right].$$

Originally, Jeffreys' prior was derived by invariance arguments [2]: the value of $\mathcal{P}_{\text{EVJ}}(D)$ is invariant under one–one transformations of the parameter space. We see here that it also plays a role as the prior which makes the Bayesian evidence (a random process) asymptotically equivalent to the stochastic complexity (not a random process). The results in [7] show that it is the *only* proper prior doing so. On the other hand, before introducing (12), Rissanen used the marginal distribution $\mathcal{P}_{\text{EV}}$ (equation (6)) as the mathematical definition of stochastic complexity. We see that if Jeffreys' prior is used, this still coincides asymptotically with stochastic complexity as given by (12). However, as shown in [27], as soon as another prior is used, then there always exist $\Theta \in \mathcal{M}$ such that with Θ-probability 1,

$$\lim_{N\to\infty} -\log \mathcal{P}_{\text{EV}}(\mathbf{d}_1, \ldots, \mathbf{d}_N) - [-\log \mathcal{P}_{\text{SC}}^N(\mathbf{d}_1, \ldots, \mathbf{d}_N)] = C,$$

with C a constant *greater* than 0. For most priors this constant can be quite large.

Together with the theoretical results obtained in [27] and [25], equation (16) implies that the predictive distribution based on $\mathcal{P}_{\text{EVJ}}$ should yield very accurate predictions. In the next section, we derive an analytic expression for Jeffreys' prior $\pi(\Theta)$ for the case where Θ indexes a Bayesian network, and show how to calculate $\mathcal{P}_{\text{EVJ}}$ for the class of Bayesian networks with a fixed but arbitrary structure.

3. Predictive Distributions for Bayesian Networks

3.1. Bayesian Networks

A Bayesian (belief) network [24, 28] is a graphical high-level representation of a probability distribution over a set of discrete variables. A Bayesian network consists of a *structure* $\mathcal{G}$ and a *parameter set* Θ.

The Bayesian network structure $\mathcal{G}$ is a directed acyclic graph (DAG), where the nodes correspond to the domain variables $X_1, \ldots, X_m$. The graph $\mathcal{G}$ can be represented by a set of $m-1$ *parent variable sets* $\Pi_i \subseteq \{X_{i+1}, \ldots, X_m\}$ where $1 \leq i < m$. For each variable X_i, the parent set Π_i represents the set consisting of the variables for which the corresponding node in the graph $\mathcal{G}$ is a parent (predecessor) of the node corresponding to the variable X_i. For simplicity, we shall henceforth forget about the mapping between the nodes and the random variables, and treat the variables as if they were nodes of the graph $\mathcal{G}$. In addition, the possible configurations of a parent set Π_i are assumed to be stored in an indexed table, which allows us to treat Π_i as a random variable with possible values from a set $\mathcal{X}^i = \{1, \ldots, c_i\}$.

Each Bayesian network topology (parent set) $\mathcal{G}$ defines a set of independence assumptions which allow the joint probability distribution for variables $X_1, \ldots, X_m$ to be written as a product of simple conditional probabilities,

$$(18)\ P(\mathbf{d}) = P(X_1 = x_1, \ldots, X_m = x_m) = \prod_{i=1}^{m} P(X_i = x_i | \Pi_i = \pi_i),$$

where $\pi_i \in \mathcal{X}^i$. In other words, a Bayesian network structure $\mathcal{G} = \{\Pi_1, \ldots, \Pi_{m-1}\}$ represents the class of all probability distributions on variables $X_1, \ldots, X_m$ such that $P(\mathbf{d})$ can, for all $\mathbf{d}$, be written as in (18). It follows that in the Bayesian network model family induced by a graph $\mathcal{G}$, a single distribution P can be uniquely determined by fixing the values of the parameters $\Theta = (\theta^1, \ldots, \theta^m)$, where

$$\theta^i = (\theta^i_{11}, \ldots, \theta^i_{1n_i}, \ldots, \theta^i_{c_i 1}, \ldots, \theta^i_{c_i n_i}),$$

n_i is the number of values of X_i, c_i is the number of possible configurations of Π_i, and

$$\theta^i_{\pi_i x_i} = P(X_i = x_i \mid \Pi_i = \pi_i).$$

In the following we assume an arbitrary but fixed structure $\mathcal{G}$ and we consider the family of corresponding probability distributions $\mathcal{M}_{\mathcal{G}}$, which contains all Θ as defined above excluding points at the boundaries of the parameter space. Formally, $\Theta \in \mathcal{M}_{\mathcal{G}}$ if and only if

1. $\theta^i_{\pi_i x_i} > 0$ and $\theta^i_{\pi_i n_i} = 1 - \sum_{x_i=1}^{n_i-1} \theta^i_{\pi_i x_i}$.
2. All conditional distributions of variables given values for their parent values are multinomial: $X_{i|\pi_i} \sim \mathrm{Multi}(1; \theta^i_{\pi_i 1}, \ldots, \theta^i_{\pi_i n_i})$.

Since the family of Dirichlet densities is *conjugate* (see e.g. [11]) to the family of multinomials, i.e. the functional form of parameter distribution is invariant in the prior-to-posterior transformation, we assume that the prior distributions of the parameters are from this family[1]. More precisely, let $(\theta^i_{\pi_i 1}, \ldots, \theta^i_{\pi_i n_i}) \sim \mathrm{Di}(\mu^i_{\pi_i 1}, \ldots, \mu^i_{\pi_i n_i})$, where $(\mu^i_{\pi_i 1}, \ldots, \mu^i_{\pi_i n_i})$ are the *hyperparameters* of the corresponding distributions. Assuming that the parameter vectors $(\theta^i_{\pi_i 1}, \ldots, \theta^i_{\pi_i n_i})$ are independent, the joint prior distribution of all the parameters Θ is

$$\prod_{i=1}^{m} \prod_{\pi_i=1}^{c_i} \mathrm{Di}(\mu^i_{\pi_i 1}, \ldots, \mu^i_{\pi_i n_i}).$$

Having now defined the prior distribution, the predictive distributions $\mathcal{P}_{\mathrm{MAP}}$ (5) and $\mathcal{P}_{\mathrm{EV}}$ (7) can be written more explicitly, as will be shown in the next two sections. The general stochastic complexity predictive distribution $\mathcal{P}_{\mathrm{SC}}$ (13) is instantiated for the Bayesian network case in Section 3.4. For being able to determine the $\mathcal{P}_{\mathrm{EVJ}}$ predictive distribution, in Section 3.5 we show how to compute Jeffreys' prior for a given Bayesian network model. In Section 4 we see that, for the subclass of Bayesian networks used in our experiments, Jeffreys' prior, as needed in computing the predictive distribution $\mathcal{P}_{\mathrm{EVJ}}$, is indeed of the proper conjugate (Dirichlet) form.

3.2. The $\mathcal{P}_{\mathrm{MAP}}$ predictive distribution for Bayesian Networks

For any $\mathbf{d}_j \in D$, let d_{ji} denote the value of X_i in $\mathbf{d}_j$, and $d_{j[\Pi_i]}$ the value of Π_i in $\mathbf{d}_j$. In addition, let us introduce the following two indicator variables $z^i_{jx_i}$ and $z^i_{j\pi_i}$:

$$z^i_{jx_i} = \begin{cases} 1, & \text{if } d_{ji} = x_i, \\ 0, & \text{otherwise.} \end{cases}, \text{ and } z^i_{j\pi_i} = \begin{cases} 1, & \text{if } d_{j[\Pi_i]} = \pi_i, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

For an unindexed test vector $\mathbf{d}$ we simply omit the subscript j and write $z^i_{x_i}$ and $z^i_{\pi_i}$.

As already noted in Section 2.2, the MAP predictive distribution can be determined by computing the likelihood of a test vector $\mathbf{d}_t = (\mathbf{u}, \mathbf{v})$, $t = N + 1$:

$$\mathcal{P}_{\mathrm{MAP}}(\mathbf{d}_t \mid D) = P(\mathbf{d}_t \mid \hat{\Theta}(D)) = \prod_{i=1}^{m} \prod_{\pi_i=1}^{c_i} \prod_{x_i=1}^{n_i} (\hat{\theta}^i_{\pi_i x_i})^{z^i_{tx_i} z^i_{t\pi_i}},$$

[1] Additional justifications for using Dirichlet priors are given in [14], where it is shown that under certain reasonable assumptions, Dirichlet is the only prior that can be used without violating the assumptions made.

where $z^i_{tx_i}$ and $z^i_{t\pi_i}$ are indicator variables as defined above and $\hat{\theta}^i_{\pi_i x_i}$ is given by (see, for example, [17])

$$\hat{\theta}^i_{\pi_i x_i} = \frac{f^i_{\pi_i x_i} + \mu^i_{\pi_i x_i} - 1}{\sum_{l=1}^{n_i}\left(f^i_{\pi_i l} + \mu^i_{\pi_i l}\right) - n_i}.$$

Here $f^i_{\pi_i x_i}$ are the sufficient statistics of the training data D: $f^i_{\pi_i x_i}$ is the number of data vectors where variable X_i has value x_i and the parents of X_i have configuration π_i:

$$f^i_{\pi_i x_i} = \sum_{j=1}^{N} z^i_{jx_i} z^i_{j\pi_i},$$

where $z^i_{jx_i}$ and $z^i_{j\pi_i}$ are defined as in (19). With the uniform prior all the hyperparameters $\mu^i_{\pi_i x_i}$ are set to 1, in which case we get the standard maximum likelihood estimator,

$$\tilde{\theta}^i_{\pi_i x_i} = \frac{f^i_{\pi_i x_i}}{\sum_{l=1}^{n_i} f^i_{\pi_i l}}.$$

3.3. The $\mathcal{P}_{\mathrm{EV}}$ predictive distribution for Bayesian Networks

The evidence predictive distribution (7) is defined as an integral over the parameter space. As shown in [9, 17], with Bayesian networks this integral can be solved analytically, yielding

$$\mathcal{P}_{\mathrm{EV}}(\mathbf{d}_t \mid D) = \prod_{i=1}^{m} \prod_{\pi_i=1}^{c_i} \prod_{x_i=1}^{n_i} (\bar{\theta}^i_{\pi_i x_i})^{z^i_{tx_i} z^i_{t\pi_i}}, \tag{20}$$

where

$$\bar{\theta}^i_{\pi_i x_i} = \frac{f^i_{\pi_i x_i} + \mu^i_{\pi_i x_i}}{\sum_{l=1}^{n_i}\left(f^i_{\pi_i l} + \mu^i_{\pi_i l}\right)}.$$

From (20) we see that similarly to the $\mathcal{P}_{\mathrm{MAP}}$ predictive distribution case, the resulting predictive distribution can be regarded as a likelihood of the test vector $\mathbf{d}_t$, but now taken at the mean of the posterior rather than at the mode. This result is somewhat surprising, since it means that being able to determine the expectations of the parameters as described above, the corresponding single Bayesian network model represents all the possible models representable by the same network structure, in the sense that it produces the same predictive distribution that would be obtained by integrating over all the different parameter instantiations.

3.4. The $\mathcal{P}_{\mathrm{SC}}$ predictive distribution for Bayesian Networks

The stochastic complexity predictive distribution is proportional to the likelihood of the combined data set $D^+ = D \cup \mathbf{d}_t$ at the maximum likelihood point:

$$\mathcal{P}_{\mathrm{SC}}(\mathbf{d}_t \mid D) \propto P(D^+ \mid \tilde{\Theta}(D^+)) = \prod_{i=1}^{m} \prod_{\pi_i=1}^{c_i} \prod_{x_i=1}^{n_i} (\tilde{\theta}^i_{\pi_i x_i})^{(f^i_{\pi_i x_i})^+},$$

where

$$\tilde{\theta}^i_{\pi_i x_i} = \frac{(f^i_{\pi_i x_i})^+}{\sum_{l=1}^{n_i} (f^i_{\pi_i l})^+},$$

and $(f^i_{\pi_i l})^+$ are the sufficient statistics of D^+. Consequently, the predictive distribution can also in this case be regarded as a likelihood of the test vector $\mathbf{d}_t$, but the maximum likelihood estimator is now computed from the extended data set, consisting of the original data set together with the test vector itself.

3.5. The $\mathcal{P}_{\mathrm{EVJ}}$ predictive distribution for Bayesian Networks

As can be seen from (17), Jeffreys' prior is proportional to the square root of the determinant of the Fisher information matrix. Since the Fisher information matrix for N data vectors can be obtained from the information matrix for only one vector simply by multiplying all the elements by N, it is sufficient to consider only the simple case with only one data vector. The log-likelihood of a data vector $\mathbf{d}$ can be written as

(21)
$$\log P(\mathbf{d} \mid \Theta) = \sum_{i=1}^{m} \sum_{\pi_i=1}^{c_i} z^i_{\pi_i} \left(\sum_{x_i=1}^{n_i} \left(z^i_{x_i} \log \theta^i_{\pi_i x_i} \right) + z^i_{n_i} \log \theta^i_{\pi_i n_i} \right).$$

where $z^i_{\pi_i}$ and $z^i_{x_i}$ are defined as in (19).

Let us consider the element $(\theta^{i_1}_{q_{i_1} l_1}, \theta^{i_2}_{q_{i_2} l_2})$ of the second derivative (Hessian) matrix of (21), where $l_1, l_2 \in \{x_{i1}, \ldots, x_{in_i}\}$. If either the variable indices i_1, i_2 or the parent configurations q_{i_1}, q_{i_2} are different, then clearly the second derivative is zero, and thus also the corresponding element of the information matrix is zero. It follows that the only non-zero elements of the information matrix are in submatrices where both parameters in question have the same variable and configuration index. Let us now consider one of these submatrices $I^i_{\pi_i}(\Theta)$, where i is the variable index and π_i the parent configuration. We need two type of second derivatives: first for the case when the value indices l_1 and l_2

are different, and secondly for case when they are the same. After some simple calculus we get

$$(22)\qquad -\frac{\partial^2 \log P(\mathbf{d} \mid \Theta)}{\partial\theta^i_{\pi_i l_1}\partial\theta^i_{\pi_i l_2}} = \begin{cases} \frac{z^i_{\pi_i} z^i_{n_i}}{(\theta^i_{\pi_i n_i})^2}, & \text{if } l_1 \neq l_2, \\ \\ \frac{z^i_{\pi_i} z^i_{l}}{(\theta^i_{\pi_i l})^2} + \frac{z^i_{\pi_i} z^i_{n_i}}{(\theta^i_{\pi_i n_i})^2}, & \text{if } l_1 = l_2 = l. \end{cases}$$

The elements of the Fisher information matrix are now the expectations of (22) over the set of all possible data vectors $\mathcal{D}$. For the case where $l_1 \neq l_2$ we get

$$\begin{aligned} \mathrm{E}_\Theta\left[\frac{-\partial^2 \log P(\mathbf{d} \mid \Theta)}{\partial\theta^i_{\pi_i l_1}\partial\theta^i_{\pi_i l_2}}\right] &= \sum_{\mathbf{d}\in\mathcal{D}} P(\mathbf{d} \mid \Theta)\frac{z^i_{\pi_i} z^i_{n_i}}{(\theta^i_{\pi_i n_i})^2} \\ &= \frac{1}{(\theta^i_{\pi_i n_i})^2}\sum_{\{\mathbf{d}\in\mathcal{D} \mid \Pi_i=\pi_i, X_i=n_i\}} P(\mathbf{d} \mid \Theta) \\ (23)\qquad &= \frac{P(\Pi_i = \pi_i, X_i = n_i \mid \Theta)}{(\theta^i_{\pi_i n_i})^2} = \frac{P(\Pi_i = \pi_i \mid \Theta)}{\theta^i_{\pi_i n_i}}. \end{aligned}$$

Similarly, with $l_1 = l_2 = l$ we get

$$(24)\quad \mathrm{E}_\Theta\left[\frac{-\partial^2 \log P(\mathbf{d} \mid \Theta)}{\partial(\theta^i_{\pi_i l})^2}\right] = \frac{P(\Pi_i = \pi_i \mid \Theta)}{\theta^i_{\pi_i l}} + \frac{P(\Pi_i = \pi_i \mid \Theta)}{\theta^i_{\pi_i n_i}}.$$

Multiplying these elements by N, we get the Fisher information matrix

(25)

$$I^i_{\pi_i}(\Theta) = \begin{pmatrix} N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i 1}} + \frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) & N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) & \cdots & N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) \\ N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) & N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i 2}} + \frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) & \cdots & N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) \\ \vdots & \vdots & \ddots & \vdots \\ N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) & N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) & \cdots & N(\frac{P^i_{\pi_i}}{\theta^i_{\pi_i, n_i - 1}} + \frac{P^i_{\pi_i}}{\theta^i_{\pi_i n_i}}) \end{pmatrix},$$

where $P^i_{\pi_i} = P(\Pi_i = \pi_i|\Theta)$. Notice that $I^i_{\pi_i}(\Theta)$ is an $(n_i - 1) \times (n_i - 1)$ matrix since $\theta^i_{\pi_i n_i}$ is completely determined by $\theta^i_{\pi_i 1}, \ldots, \theta^i_{\pi_i, n_i - 1}$. It is relatively easy to show (see [3]) that the determinant of (25) is given by

$$(26)\qquad |I^i_{\pi_i}(\Theta)| = \frac{(N \cdot P^i_{\pi_i})^{n_i - 1}}{\prod_{l=1}^{n_i}\theta^i_{\pi_i l}}.$$

The whole Fisher information matrix $I(\Theta)$ is a block diagonal matrix, where the blocks are the submatrices $I^i_{\pi_i}(\Theta)$. The determinant of a block diagonal matrix is product of the determinants of the blocks, and thus

$$|I(\Theta)| = \prod_{i=1}^{m} \prod_{\pi_i=1}^{c_i} \frac{(N \cdot P^i_{\pi_i})^{n_i-1}}{\prod_{l=1}^{n_i} \theta^i_{\pi_i l}}. \tag{27}$$

Finally, as noted in Section 2.4, $\pi(\Theta) \propto \sqrt{|I(\Theta)|}$, so we get

$$\begin{aligned}\pi(\Theta) &\propto \prod_{i=1}^{m} \prod_{\pi_i=1}^{c_i} \left((N \cdot P^i_{\pi_i})^{\frac{n_i-1}{2}} \prod_{l=1}^{n_i} (\theta^i_{\pi_i l})^{-\frac{1}{2}} \right) \\ &\propto \prod_{i=1}^{m} \prod_{\pi_i=1}^{c_i} \left((P^i_{\pi_i})^{\frac{n_i-1}{2}} \prod_{l=1}^{n_i} (\theta^i_{\pi_i l})^{-\frac{1}{2}} \right).\end{aligned}$$

However, computing Jeffreys' prior as formulated above requires computing for each variable the marginal distribution of its parents. Unfortunately, for multi-connected Bayesian networks, this problem is known to be NP-hard [8]. In the experiments reported in Section 4, we used a simple tree-structured Bayesian network, in which case Jeffreys' prior is of a proper conjugate form, and it can be computed efficiently.

4. Empirical Results

4.1. Experimental setup

In the following, we concentrate on the standard classification problem, where the task is to predict the value of a single *classification variable*, given the values of all the other variables. Consequently, the set of free variables consists of only one variable, and the set of clamped variables contains all the other variables — in other words, by using the notation given in Section 2, U $= \{X_1, \ldots, X_{m-1}\}$ and V $= \{X_m\}$, and we wish to compute probabilities of the form $\mathcal{P}(X_m = x_m \mid X_1 = x_1, \ldots, X_{m-1} = x_{m-1}, D)$.

In the Naive Bayes classifier case, the variables in U are assumed to be independent, given the value of variable X_m. Consequently, we can regard the Naive Bayes model as a simple tree-structured Bayesian network, where variable X_m forms the root of the tree, and variables $X_1, \ldots, X_{m-1}$ are represented by the leaves. In this case, the Jeffreys' prior formula (17) reduces to

$$\begin{aligned}\pi(\Theta) &\propto \prod_{k=1}^{K} (\theta^m_k)^{-\frac{1}{2}} \prod_{i=1}^{m-1} \prod_{k'=1}^{K} \left((\theta^m_{k'})^{\frac{n_i-1}{2}} \prod_{l=1}^{n_i} (\theta^i_{k'l})^{-\frac{1}{2}} \right) \\ &= \prod_{k=1}^{K} (\theta^m_k)^{\frac{1}{2}(\sum_{i=1}^{m-1}(n_i-1)-1)} \prod_{i=1}^{m-1} \prod_{k'=1}^{K} \prod_{l=1}^{n_i} (\theta^i_{k'l})^{-\frac{1}{2}}, \end{aligned} \tag{28}$$

where K denotes the number of the values of the root variable X_m. Consequently, the prior distribution can be represented as a product of Dirichlet distributions,

$$\Theta \sim \mathrm{Di}(\frac{1}{2}(\sum_{i=1}^{m-1}(n_i-1)+1),\ldots,\frac{1}{2}(\sum_{i=1}^{m-1}(n_i-1)+1))\prod_{i=1}^{m-1}\prod_{k=1}^{K}\mathrm{Di}(\frac{1}{2},\ldots,\frac{1}{2}).$$

For our experiments with the Naive Bayes classifier, eight public domain classification data sets of varying size were used (the data sets can be obtained from the UCI data repository [4]). Table I describes the size (N), the number of attributes (m), and the number of classes (K) for each of these data sets.

Table I. The datasets used in the experiments.

Dataset	Data vectors	Attributes	Classes	CV folds
Heart Disease (HD)	270	14	2	9
Iris (IR)	150	5	3	5
Lymphography (LY)	148	19	4	5
Australian (AU)	690	15	2	10
Breast Cancer (BC)	286	10	2	11
Diabetes (DB)	768	9	2	12
Glass (GL)	214	10	6	7
Hepatitis (HE)	150	20	2	5

For comparing the predictive accuracy of different predictive distributions, we used two different utility functions: the *log-score* and the *0/1-score*. The log-score of a predictive distribution $\mathcal{P}(X_m|\mathbf{u},D)$ is defined as $-\log\mathcal{P}(X_m=k|\mathbf{u},D)$ where k denotes the actual ("true") classification, i.e., the correct value of X_m. When using logarithm of base two, the log-score has the following coding interpretation: If one encodes the data using the code corresponding to $\mathcal{P}(X_m|\mathbf{u},D)$, then $\log\mathcal{P}(X_m=k|\mathbf{u},D)$ is equal to the number of bits one needs to describe the actual outcome k. On the other hand, if we imagine the correct predictive distribution to give in this case probability one to the actual outcome, and probability zero to other outcomes, we see that the log-score is equivalent to the cross-entropy or Kullback-Leibler distance between the above defined degenerate distribution and the predictive distribution produced.

For the 0/1-score, we simply first determine the k for which the probability $\mathcal{P}(X_m=k|\mathbf{u},D)$ is maximized, and the 0/1-score is then defined to be 1, if the actual outcome indeed was k, otherwise it is defined to be 0.

Two separate sets of experiments were performed on each data set by using the following predictive inference methods with the Naive Bayes model described above:

- ML: The $\mathcal{P}_{\mathrm{MAP}}$ predictive distribution (5) with uniform prior (equivalent to the predictive distribution with the maximum likelihood model).
- EV: The $\mathcal{P}_{\mathrm{EV}}$ predictive distribution (7) with uniform prior.
- SC: The $\mathcal{P}_{\mathrm{SC}}$ predictive distribution (13).
- EVJ: The $\mathcal{P}_{\mathrm{EV}}$ predictive distribution (7) with Jeffreys' prior (28).

In the first set of experiments (Section 4.2) we measured the crossvalidated prediction performance by using the two utility functions described above. In our second set of experiments (Section 4.3), we studied how the prediction quality of our various approaches depends on the size of the training set D.

4.2. Crossvalidation results

In our crossvalidation experiments, we initially used each of the datasets with the same number of folds as in the major experimental comparison performed by the Statlog project [22] (the number of folds used in each case can be found in Table I). However, it should be noted that although the result of one crossvalidation run is an average of n numbers, where n is the number of folds used, the result depends of course on how the n folds are selected from the sample data. To see how much the results vary with different fold partitionings, we performed 100 independent crossvalidation runs where the data was randomly partitioned into n folds, and computed the minimum, the average, and the maximum of the crossvalidated prediction accuracies obtained. As can be seen in Figures 1 (in the log-score case) and 2 (in the 0/1-score case), the crossvalidation results can vary quite a lot depending on the specific fold partitioning used.

Though the differences in crossvalidation results between different prediction methods are small, we see that for the log-score, evidence with uniform prior performs consistently better than the other methods, followed very closely by the evidence with Jeffreys' prior. The ML approach produces the worst results. For the 0/1-score, the picture is not as clear-cut, and the differences are quite small. However, it should be noted that in this case the ML approach produces the best results

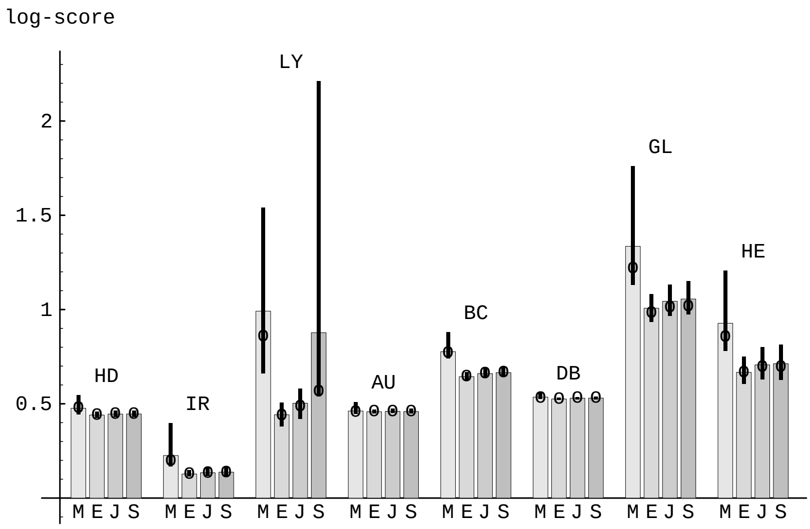

Figure 1. The minimum (lower end of the black line), the average (grey bar), and the maximum (upper end of the black line) of the crossvalidated log-scores obtained by 100 independent crossvalidation runs. The corresponding leave-one-out crossvalidation results are marked with small circles. The y-axis represents the log-score, so the smaller the score, the better. The prediction methods used are ML (denoted here by M), EV (E), EVJ (J), and SC (S).

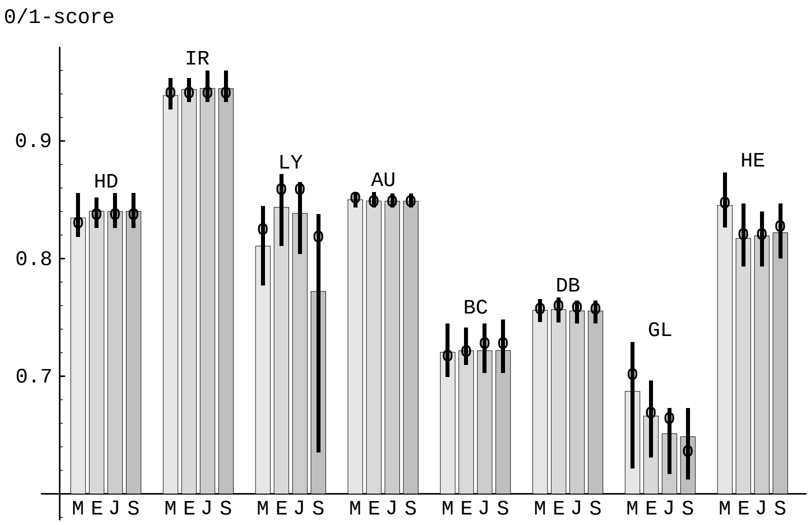

Figure 2. The minimum (lower end of the black line), the average (grey bar), and the maximum (upper end of the black line) of the crossvalidated 0/1-scores obtained by 100 independent crossvalidation runs. The corresponding leave-one-out crossvalidation results are marked with small circles. In this picture higher score is better. The prediction methods used are ML (denoted here by M), EV (E), EVJ (J), and SC (S).

with two of the data sets (GL and HE). One explanation for this fact may be that for the much coarser 0/1-score, it is in many cases not important exactly what probability we attach to a class value being k; all probability distributions over the class values for which k gets the maximum probability will lead to the same prediction. Thus it can very well happen that, while the ML prediction captures less well the regularities underlying the data (and hence performs worse with respect to log-score), it still captures them well enough to give maximum probability to the class value that should indeed receive maximum probability.

In addition to comparison purposes between the different predictive distributions, the results in Figures 1 and 2 are interesting as they show surprisingly good performance of the Naive Bayes model when compared to the results reported in the machine learning literature (for references, see e.g., [31]). This fact becomes especially clear if we look at the maximal results reported here, which in many cases is justifiable as many of the results reported in the literature seem to be obtained in this way, although this fact may not be explicitly stated.

The high variance of the results obtained indicate that one single n-fold crossvalidation run can not be used as a reliable measure for comparing various predictive inference methods, unless the same specific fold partitioning is used in all cases. If, however, a number of independent runs is performed, then some statistical measure, such as the average, can be used for this purpose. Alternatively, the leave-one-out results seem to follow the behavior of the averaged crossvalidation results quite accurately. For this reason, we decided to restrict ourselves to leave-one-out crossvalidation in the next section.

4.3. Results with varying amount of training data

To see how the prediction quality of our various approaches depends on the size of the training set D, we performed a set of experiments using only small fractions of the training data available. In these experiments, leave-one-out crossvalidation was used, but at each step, only the k first vectors from the training set were used in order to predict the test vector that was "left out", and this procedure was repeated for $k = 1, \ldots, N-1$. As this setup is dependent on the ordering of the data vectors, the whole leave-one-out crossvalidation cycle was then repeated 100 times with 100 randomly generated permutations of the dataset. The averaged (over the 100 leave-one-out crossvalidation runs) results are plotted as a function of k in Figures 3–6. These statistics of the behavior of different predictive distributions as a function of increasing amount of training data should now give us some idea as to the typical behavior of our prediction methods. It should be noted that since with small sample sizes, the

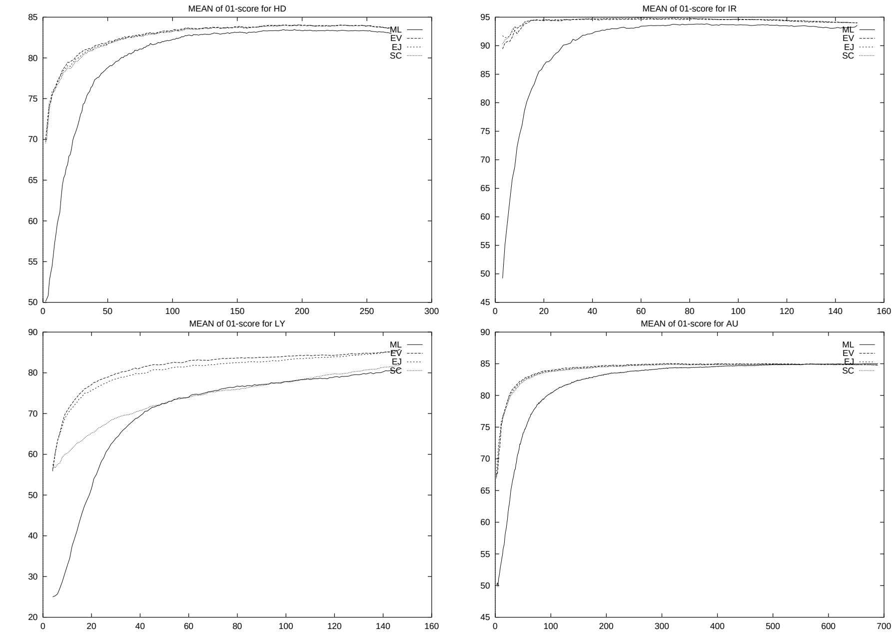

Figure 3. Average leave-one-out 0/1-scores obtained with different predictive distributions for the HD, IR, LY and AU dataset cases as a function of the number of the training examples used.

ML method will sometimes yield infinitely bad log-score, in order to prevent scaling problems when presenting the results graphically, the *p-score* (the probability of the correct class, instead of its logarithm) was used in these tests as the alternative score for the 0/1-score, and not the log-score.

All in all, the results with all the eight datasets used show very similar behavior: the evidence-based EV and EVJ approaches perform surprisingly well even in cases where the training data consists of only a few data vectors, which shows that the data sets used here are quite redundant, and when properly used, only a very small sample of these data sets is needed for constructing good models. In the following section, we analyze three further interesting aspects of the results.

4.4. Discussion

It is a well-known fact that, for small sample sizes, the ML predictor is too dependent on the observed data and does not take into account that future data *may* turn out to be different. Our results support this observation and show that compared to the other methods, the ML predictive distribution appears to be much more sensitive to the amount of data available. This phenomenon can be explained by the fact that

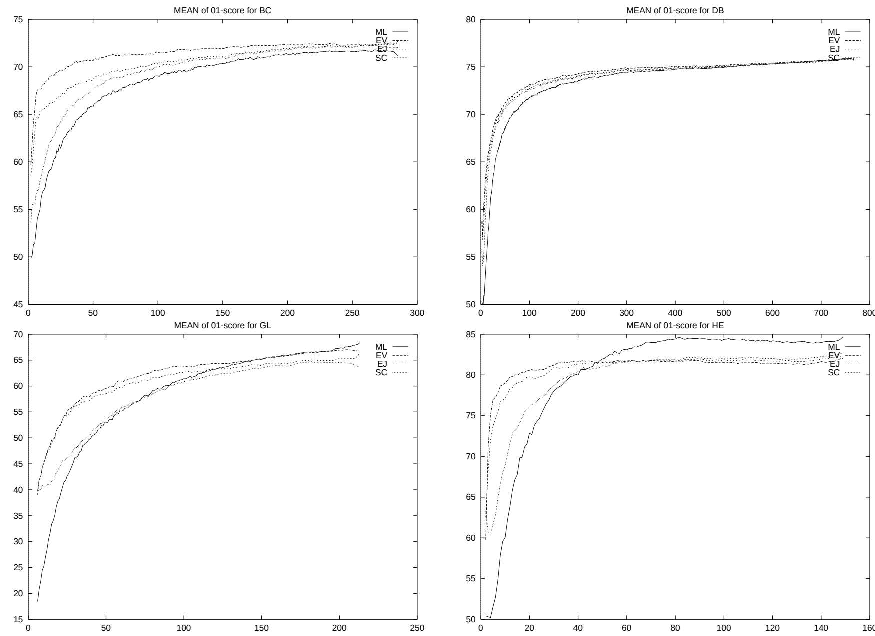

Figure 4. Average leave-one-out 0/1-scores obtained with different predictive distributions for the BC, DB, GL and HE dataset cases as a function of the number of the training examples used.

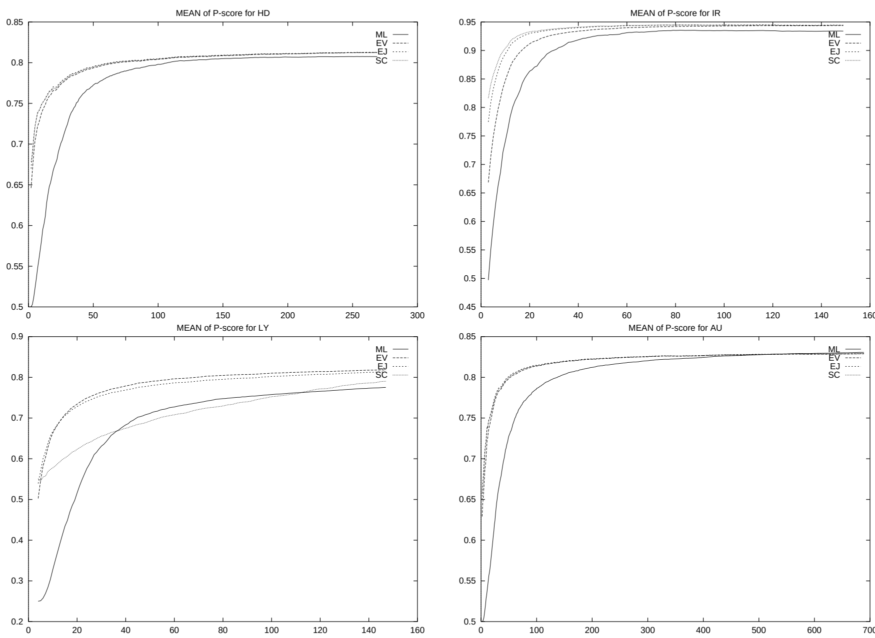

Figure 5. Average leave-one-out p-scores obtained with different predictive distributions for the HD, IR, LY and AU dataset cases as a function of the number of the training examples used.

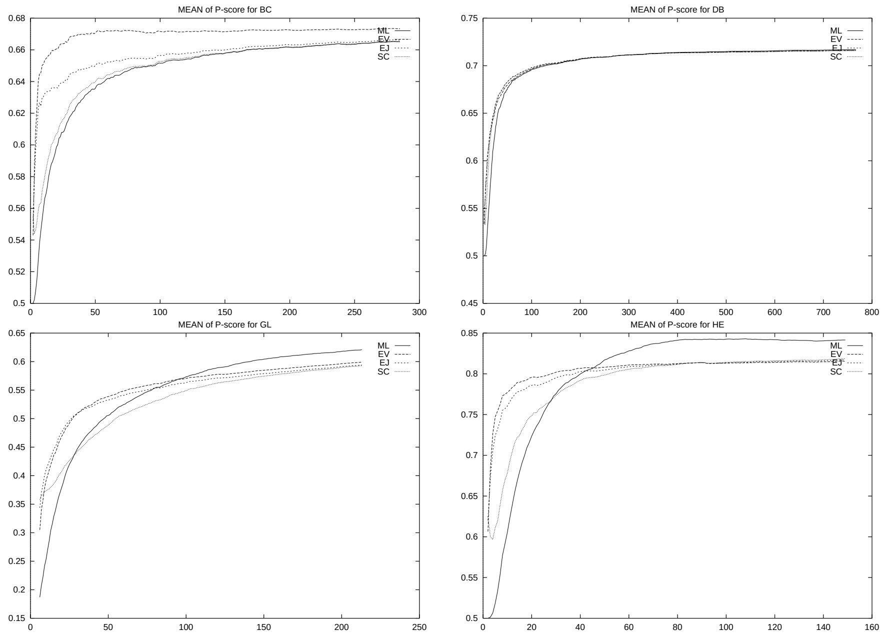

Figure 6. Average leave-one-out p-scores obtained with different predictive distributions for the BC, DB, GL and HE dataset cases as a function of the number of the training examples used.

the EV and EVJ approaches are more conservative methods as they base their predictions on expected values of the parameters, or actually, on integrating over all the parameter values, while ML makes more "eager" predictions based on the maximum likelihood estimator.

Let us consider a very simple example to illustrate this point. Suppose our data consists of a string of ones and zeros generated by some i.i.d Bernoulli-process $p = P(X = 1)$. If we have seen an initial string consisting of one '1', and no zeros, then the ML predictor will determine that the probability of the second symbol being a '1' is unity. However, using the EV prediction, this probability is $\frac{2}{3}$. If the next data item turns out to be a '0', then the log-score of the ML predictor will be $-\infty$ while that of the EV will be $\log 2 - \log 3$. The behavior of the SC and EVJ methods lie somewhere in between that of ML and EV. In our Bernoulli example, the probability of the second symbol being a '1' would be $\frac{3}{4}$ for EVJ and $\frac{4}{5}$ for SC.

Theoretically, for large sample sizes EVJ and SC should lead to the same predictions, since $\mathcal{P}_{\mathrm{EVJ}}$ and $\mathcal{P}_{\mathrm{SC}}$ become asymptotically identical. It can be seen from Figures 3–6 that this phenomenon indeed occurs for most of the data sets used. For small sample sizes however, EVJ usually performs somewhat better than SC. We believe that this is caused by the fact that $\mathcal{P}_{\mathrm{SC}}$ as we use it here is defined for the *unsupervised* case, where the goal is to give maximally high probability for full unseen vectors $(x_1, \ldots, x_m)$, while the methods were tested in the *supervised* case, where only one variable (the class variable x_m) was actually predicted. An exact optimization of the SC method for the supervised classification case would require the use of a *conditional* maximum likelihood estimator of data D. Using the notation of Section 2.1, this is the model Θ in model class $\mathcal{M}$ that maximizes the *conditional* probability $P(D_V|D_U, \Theta)$ where D_V is the set of the N instantiations of the free variables in V and D_U is the set of N instantiations of the clamped variables in U. This modification is by no means straightforward and is left as a goal for future work.

The results presented by Rissanen [26, 27] imply that application of the MDL framework should lead to quite accurate predictions. However, in the experiments performed, predictions based on the standard Bayesian EV approach (marginal likelihood predictive distribution with uniform prior) usually gave slightly better results than predictions based on the EVJ approach motivated by MDL. We conjecture that the reasons for this are twofold. First, it should be remembered that $-\log \mathcal{P}_{\mathrm{EVJ}}(D)$ is only an asymptotic approximation of the actual stochastic complexity (10). The small sample size behavior of this approximation is theoretically not well understood, and hence the small size performance of this approximation tells nothing about the validity of the MDL approach per se. Second, it is important to note that the fundamental goal behind the MDL approach is to optimize the 'worst-case' behavior of a predictive distribution: *whatever* the data D is, $-\log \mathcal{P}_{\mathrm{EVJ}}(D)$ will be about equally close to $-\log P(D|\tilde{\Theta}(D))$, the code length obtained by using the best-fitting model in the class for D. For the model class of Bernoulli processes, Rissanen [27] showed that this leads to a considerable gain (in the sense of smaller number of bits needed to code the data and hence better predictions in the log-score sense) over $\mathcal{P}_{\mathrm{EV}}$ with the uniform prior, if the data sets used are highly 'skewed'. By skewed data sets we mean here data for which the ML estimator lies near the boundary of the parameter space $\mathcal{M}$.

Intuitively, the above phenomenon is easy to accept when we recall that $\mathcal{P}_{\mathrm{EVJ}}$ employs Jeffreys' prior which yields a slight preference to models near the boundary points of the parameter space, while $\mathcal{P}_{\mathrm{EV}}$ assumes a uniform prior over the model parameters. Consequently, while $\mathcal{P}_{\mathrm{EVJ}}$ can be expected to give smaller code lengths for skewed data sets than $\mathcal{P}_{\mathrm{EV}}$, the price to pay is that for non-skewed, more 'average' data sets, using $\mathcal{P}_{\mathrm{EVJ}}$ can lead to slightly inferior predictive performance. As our experimental results show, the data sets used in this study are not very skewed, since already a small amount of data was enough for obtaining as good predictive performance as with all the data. For this reason, it seems reasonable to assume that the datasets used were just not skewed enough for EVJ to outperform EV. Our preliminary studies [20] on this subject support the theoretical arguments, but all in all, this interesting hypothesis produces an important research problem that needs to be studied in more detail.

5. Conclusion and Future Work

We have described how to obtain predictive distributions by using different approximations for the problem domain probability distribution, given some sample data. Of the alternatives considered here, the predictive distribution exploiting Rissanen's new formulation of stochastic complexity is of particular interest as it has some nice theoretical properties, which suggest that the corresponding probability distribution is in some cases more accurate than the marginal likelihood distribution. A straightforward application of Rissanen's MDL formalism lead us to the SC predictive distribution. However, as discussed earlier, this predictive distribution does not form a random process which makes its use troublesome. For this reason, we first described how the evidence predictive distribution with Jeffreys' prior can be regarded as an approximation of the MDL-optimal predictive distribution, and then showed how to compute Jeffreys' priors in the Bayesian network model family case.

In the experimental part of the paper, the predictive accuracy of the ML predictive distribution (the MAP predictive distribution with uniform prior distribution), the evidence predictive distribution (with both the non-informative uniform prior, and Jeffreys' prior suggested by the MDL theory), and the straightforward SC predictive distribution were evaluated empirically by using publicly available classification data sets. For computational reasons, the specific model used in the tests was the structurally simple Naive Bayes model. In the experiments performed, in the 0/1-score sense there was no clear winner. In the log-score sense, the evidence-based predictive distributions EV and EVJ outperformed the other methods, especially in cases where the amount of training data was small. The best results were obtained by using the uniform prior. One reason for the evidence predictive distribution performing slightly worse with Jeffreys' prior may be caused by the fact that Jeffreys' prior is in a sense a "worst-case" prior, as we indicated in the last section.

The somewhat inferior performance of the SC method with small training sets can be partly explained by the non-randomness of the resulting process. In addition, the SC predictive distribution was defined so that the method would give a maximally high probability for full unseen vectors, while the methods were tested in the restricted supervised case, where only one variable (the class variable) was actually predicted. All in all, it should be emphasized that the fact that the two approaches motivated by the MDL theory (EVJ and SC) did not produce better results than the standard Bayesian EV approach, marginal likelihood prediction with uniform prior, does not imply that there is anything wrong with the MDL approach per se; rather, they illustrate the difficulty of applying the theoretically elegant MDL framework in practice in a formally solid manner.

The results with decreasing amount of training data show that the evidence-based approaches are quite robust in the sense that they predict surprisingly well even with small training sets. What is more, it should also be noted that the actual classification accuracies obtained with the Naive Bayes model are surprisingly high, when compared to the results obtained by alternative models. The latter fact has actually been noted before by several authors (see for example [21, 13]); the results reported here extend these previous works as they show that when the evidence predictive distribution is used, very good performance may already be achieved for quite small sample sizes. This suggests that in many natural domains, using the Naive Bayes model may not be so naive after all, and that the empirical results presented here also bear relevance for the general case with multi-connected Bayesian networks.

Acknowledgements

This research has been supported by the ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT). The CoSCo group has also been supported by the Technology Development Center TEKES and the Academy of Finland, and P. Grünwald by a TALENT–grant of the Netherlands Organization for Scientific Research (NWO). Part of this research was done while Grünwald was at CWI, Kruislaan 413, 1098 SJ The Netherlands. The Breast cancer and the Lymphography domains were obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Slovenia. Thanks go to M. Zwitter and M. Soklič for providing the data.

References

1. Baxter, R. and J. Oliver: 1994, 'MDL and MML: Similarities and Differences'. Technical Report 207, Department of Computer Science, Monash University.

2. Berger, J.: 1985, *Statistical Decision Theory and Bayesian Analysis.* New York: Springer-Verlag.
3. Bernardo, J. and A. Smith: 1994, *Bayesian theory.* John Wiley.
4. Blake, C., E. Keogh, and C. Merz: 1998, 'UCI repository of machine learning databases'. URL: http://www.ics.uci.edu/~mlearn/MLRepository.html.
5. Castillo, E., J. Gutiérrez, and A. Hadi: 1997, *Expert Systems and Probabilistic Network Models*, Monographs in Computer Science. New York, NY: Springer-Verlag.
6. Clarke, B. and A. Barron: 1990, 'Information-Theoretic Asymptotics of Bayes Methods'. *IEEE Transactions on Information Theory* **36**(3), 453–471.
7. Clarke, B. and A. Barron: 1994, 'Jeffrey's Prior is asymptotically least favorable under entropy risk'. *Journal of Statistical Planning and Inference* **41**, 37–60.
8. Cooper, G.: 1990, 'The computational complexity of probabilistic inference using Bayesian belief networks'. *Artificial Intelligence* **42**(2–3), 393–405.
9. Cooper, G. and E. Herskovits: 1992, 'A Bayesian Method for the Induction of Probabilistic Networks from Data'. *Machine Learning* **9**, 309–347.
10. Cover, T. and J. Thomas: 1991, *Elements of Information Theory.* New York, NY: John Wiley & Sons.
11. DeGroot, M.: 1970, *Optimal statistical decisions.* McGraw-Hill.
12. Dom, B.: 1995, 'MDL estimation with small sample sizes including an application to the problem of segmenting binary strings using Bernoulli models'. Technical Report RJ 9997 (89085), IBM Research Division, Almaden Research Center.
13. Friedman, N. and Goldszmidt: 1996, 'Building Classifiers using Bayesian Networks'. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence.* Portland, Oregon, pp. 1277–1284.
14. Geiger, D. and D. Heckerman: 1994, 'A Characterization of the Dirichlet Distribution Through Global and Local Independence'. Technical Report MSR-TR-94-16, Microsoft Research.
15. Grünwald, P.: 1998, 'The Minimum Description Length Principle and Reasoning under Uncertainty'. Ph.D. thesis, CWI, ILLC Dissertation Series 1998-03.
16. Grünwald, P., P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri: 1998, 'Minimum Encoding Approaches for Predictive Modeling'. In: G. Cooper and S. Moral (eds.): *Proceedings of the 14th International Conference on Uncertainty in Artificial Intelligence (UAI'98).* Madison, WI.
17. Heckerman, D., D. Geiger, and D. Chickering: 1995, 'Learning Bayesian Networks: The Combination of Knowledge and Statistical Data'. *Machine Learning* **20**(3), 197–243.
18. Jensen, F.: 1996, *An Introduction to Bayesian Networks.* London: UCL Press.
19. Kontkanen, P., P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald: 1997, 'Comparing Predictive Inference Methods for Discrete Domains'. In: *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics.* Ft. Lauderdale, Florida, pp. 311–318.
20. Kontkanen, P., P. Myllymäki, T. Silander, H. Tirri, and K. Valtonen: 1999, 'Exploring the Robustness of Bayesian and Information-Theoretic Methods for Predictive Inference'. In: *Proceedings of Uncertainty'99: The Seventh International Workshop on Artificial Intelligence and Statistics.* Fort Lauderdale, FL.

21. Langley, P. and S. Sage: 1994, 'Induction of Selective Bayesian Classifiers'. In: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence.* Seattle, Oregon, pp. 399–406.
22. Michie, D., D. Spiegelhalter, and C. Taylor (eds.): 1994, *Machine Learning, Neural and Statistical Classification.* London: Ellis Horwood.
23. Neapolitan, R.: 1990, *Probabilistic Reasoning in Expert Systems.* New York, NY: John Wiley & Sons.
24. Pearl, J.: 1988, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Mateo, CA.
25. Rissanen, J.: 1987, 'Stochastic Complexity'. *Journal of the Royal Statistical Society* **49**(3), 223–239 and 252–265.
26. Rissanen, J.: 1989, *Stochastic Complexity in Statistical Inquiry.* New Jersey: World Scientific Publishing Company.
27. Rissanen, J.: 1996, 'Fisher Information and Stochastic Complexity'. *IEEE Transactions on Information Theory* **42**(1), 40–47.
28. Shachter, R.: 1988, 'Probabilistic Inference and Influence Diagrams'. *Operations Research* **36**(4), 589–604.
29. Tageuchi, J. and A. Barron: 1998, 'Asymptotically minimax regret by Bayes mixtures'. In: *1998 IEEE International Symposium on Information Theory (Cambridge, MA, August 1998).*
30. Thiesson, B.: 1995, 'Score and information for recursive exponential models with incomplete data'. Technical Report R-95-2020, Aalborg University, Institute for Electronic Systems, Department of Mathematics and Computer Science.
31. Tirri, H., P. Kontkanen, and P. Myllymäki: 1996, 'Probabilistic Instance-Based Learning'. In: L. Saitta (ed.): *Machine Learning: Proceedings of the Thirteenth International Conference (ICML'96).* pp. 507–515.
32. Wallace, C. and D. Boulton: 1968, 'An information measure for classification'. *Computer Journal* **11**, 185–194.
33. Wallace, C. and P. Freeman: 1987, 'Estimation and Inference by Compact Coding'. *Journal of the Royal Statistical Society* **49**(3), 240–265.
34. Wallace, C., K. Korb, and H. Dai: 1996a, 'Causal Discovery via MML'. Technical Report 96/254, Department of Computer Science, Monash University.
35. Wallace, C., K. Korb, and H. Dai: 1996b, 'Causal Discovery via MML'. In: L. Saitta (ed.): *Machine Learning: Proceedings of the Thirteenth International Conference (ICML'96).* pp. 516–524.