Learning Locally Minimax Optimal Bayesian Networks

Tomi Silander

*A*STAR Institute of High Performance Computing, Singapore*

Teemu Roos and Petri Myllymäki

Helsinki Institute for Information Technology HIIT, Finland

Abstract

We consider the problem of learning Bayesian network models in a non-informative setting, where the only available information is a set of observational data, and no background knowledge is available. The problem can be divided into two different subtasks: learning the structure of the network (a set of independence relations), and learning the parameters of the model (that fix the probability distribution from the set of all distributions consistent with the chosen structure). There are not many theoretical frameworks that consistently handle both these problems together, the Bayesian framework being an exception. In this paper we propose an alternative, information-theoretic framework which sidesteps some of the technical problems facing the Bayesian approach. The framework is based on the minimax-optimal Normalized Maximum Likelihood (NML) distribution, which is motivated by the Minimum Description Length (MDL) principle. The resulting model selection criterion is consistent, and it provides a way to construct highly predictive Bayesian network models. Our empirical tests show that the proposed method compares favorably with alternative approaches in both model selection and prediction tasks.

1 Introduction

Bayesian networks [1,2] are one of the most popular model classes for multivariate data. Learning a Bayesian network from data reveals the probabilistic structure of the domain and provides a tool for predicting future observations. Under certain restrictions and assumptions, Bayesian networks even allow principled speculations about the causal mechanisms of the domain, and provide estimates about effects of interventions [3].

Traditionally, learning of Bayesian networks has been divided in two separate tasks: learning the structure of the network that represents conditional independence relations, and learning the parameters that specify the joint probability distribution, see [4]. The methods for learning the structure are usually based on either conditional independence tests [5,6], or some scoring function such as *a posteriori* probability or description length, see [7]. These methods are not totally separate and there are also some hybrid methods [8,9].

Methods based on conditional independence tests are sensitive to choice of significance levels. Furthermore, since they are based on interpretation of Bayesian network structures as sets of independence assumptions, they do not usually offer a natural way to learn the parameters for the structure.

The popular Bayesian BDeu [10] criterion for learning Bayesian network structures has recently been reported to be very sensitive to the choice of prior hyperparameters [11,12]. On the other hand, some alternative model selection criteria, like the Akaike information criterion (AIC) [13] and the Bayesian information criterion (BIC) [14], are derived through asymptotics, and their behavior is suboptimal for finite sample sizes, nor do they suggest a particular way to learn the parameters for Bayesian networks. To our knowledge, apart from the methods presented in this paper, the Bayesian approach is one of the very few frameworks that offer a theoretically coherent solution to both structure and parameter learning.

For large networks, the study of different scoring criteria is hindered by the fact that learning the network structure is NP-hard for all popular scoring criteria [15], even if these criteria have a convenient characteristic of decomposability, which allows incremental scoring in heuristic local search [16]. However, owing to recent advances in exact structure learning [17,18], it is feasible to find the optimal network for decomposable scores when the number of variables is about 30 or less. This makes it possible to study the behavior of different scoring criteria for problems of realistic size without the uncertainty stemming from heuristic search.

In this paper we introduce a new decomposable scoring criterion for learning Bayesian network structures, the *factorized normalized maximum likelihood* (fNML). This score features no tunable parameters, and thus avoids the sensitivity problems of Bayesian scores. We show that the new criterion is asymptotically consistent. Unlike AIC and BIC, it is derived in closed form for finite sample sizes, and it has a probabilistic interpretation as a distribution which has a certain minimax optimality property.

We also use the predictive form of the normalized maximum likelihood (NML) model [19] to find well predicting parameters given the learned network structure. This new method for learning the parameters, which we call the *factor-*

ized sequential normalized maximum likelihood (fsNML), is a natural extension of the fNML model selection criterion for predictive purposes. In order to demonstrate the theoretical validity of fsNML, we give a non-asymptotic upper-bound on the logarithmic loss (or code length) of the fsNML predictions relative to the optimal parameters – for a fixed graph structure, the fsNML predictions are never (for any data-set) much worse than those obtained by optimizing the parameters with hindsight. Both the fNML and fsNML methods are motivated by the Minimum Description Length (MDL) principle, see [20,7].

The rest of the paper is structured as follows. In Section 2, we first introduce Bayesian networks and the notation needed later. In Section 3, we first briefly review the most popular decomposable scores, after which we are ready to introduce the fNML criterion for structure learning. In Section 4 we turn our focus to the parameter learning and introduce our sNML based solution. We then describe the empirical experiments and their results in Section 5. The paper ends with discussion in Section 6 and a short conclusions in Section 7. Proofs for the central results can be found in appendices at the end of the paper.

2 Bayesian Networks

We assume the reader to be familiar with Bayesian networks (for a tutorial, see [4]), and only introduce the notation needed later in this paper.

A Bayesian network defines a joint probability distribution for an m-dimensional multivariate data vector $X = (X_1, \ldots, X_m)$. We will only consider cases in which all the variables are discrete, so that variable X_i may have r_i different values which, without loss of generality, may be denoted $\{1, \ldots, r_i\}$.

A Bayesian network consists of a directed acyclic graph G and a set of conditional probability distributions. We specify the DAG with a vector $G = (G_1, \ldots, G_m)$ of parent sets so that $G_i \subset \{X_1, \ldots, X_m\}$ denotes the parents of variable X_i, i.e., the variables from which there is an arc to X_i. Each parent set G_i has q_i ($q_i = \prod_{X_p \in G_i} r_p$) possible values that are the possible value combinations of the variables belonging to G_i. We assume a non-ambiguous enumeration of these values and denote the event that G_i holds the j^{th} value combination simply by $G_i = j$.

The local Markov property for Bayesian networks states that each variable is independent of its non-descendants given its parents. Formally, this is equiv-

alent to the following factorization of the joint distribution:

$$P(x \mid G) = \prod_{i=1}^{m} P(x_i \mid G_i). \tag{1}$$

The conditional probability distributions $P(X_i \mid G_i)$ are determined by a set of parameters, Θ, via the equation

$$P(X_i = k \mid G_i = j, \Theta) = \theta_{ijk},$$

where k is a value of X_i, and j is a value configuration of the parent set G_i. We denote the set of parameters associated with variable X_i by Θ_i and define $\Theta_{ij} = (\Theta_{ij1}, \ldots, \Theta_{ijr_i})$.

For learning Bayesian network structures we assume a data D of N complete independent and identically distributed (i.i.d.) instantiations of the vector X, i.e., an $N \times m$ data matrix without missing values. It turns out to be useful to introduce a notation for certain parts of this data matrix. We often want to select rows of the data matrix by certain criteria. We then write the selection criterion as a superscript of the data matrix D. For example, $D^{G_i=j}$ denotes those rows of D where the variables of G_i have the j^{th} value combination. If we further want to select certain columns of these rows, we denote the columns by subscripting D with a corresponding variable set. As a shorthand, we write $D_{\{X_i\}} = D_i$. For example, $D_i^{G_i=j}$ selects the i^{th} column of the rows $D^{G_i=j}$.

Since the rows of D are assumed to be i.i.d., the probability of a data matrix can be calculated just by taking the product of the row probabilities. Combining equal terms yields

$$P(D \mid G, \Theta) = \prod_{i=1}^{m} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}, \tag{2}$$

where N_{ijk} denotes number of rows in $D^{X_i=k, G_i=j}$. We also define a vector $\vec{N}_{ij} = (N_{ij1}, \ldots, N_{ijr_i})$ and a sum $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

For a given structure G, we define the *maximized likelihood*,

$$\hat{P}(D \mid G) = \sup_{\Theta} P(D \mid G, \Theta). \tag{3}$$

Note that $\hat{P}(D \mid G)$ does not define a probability distribution for the data since the maximizing parameters depend on the data D at which the likelihood is evaluated, and hence the sum over all data-sets is generally greater than one. It is not difficult to show that the maximizing parameters in (3) are simply the relative frequencies found in data: $\hat{\theta}_{ijk} = N_{ijk}/N_{ij}$, where N_{ij} denotes the number of rows in $D^{G_i=j}$; in case $N_{ij} = 0$, we define $\hat{\theta}_{ijk} = 1/r_i$. We often drop the dependency on G when it is clear from the context.

3 Model Selection

As said in the introduction, methods for learning the structure of a Bayesian network based on data can be (with only a little violence) divided into those based on independence tests and those based on scores. Here we focus on the score-based approach.

A *scoring function* is simply a function of the structure G and observed data D which evaluates different structures according to their goodness in the light of the data D; the higher the score, the better the structure. A scoring function $\textsc{Score}(G, D)$ for learning a Bayesian network structure is called *decomposable*, if and only if it can be expressed as a sum of local scores

$$\textsc{Score}(G, D) = \sum_{i=1}^{m} S(D_i, D_{G_i}), \tag{4}$$

for all G and D.

Many popular scoring functions avoid overfitting by balancing the fit to the data with the complexity of the model. A common form of this idea can be expressed as

$$\textsc{Score}(G, D) = \log \hat{P}(D \mid G) - \Delta(D, G), \tag{5}$$

where $\Delta(D, G)$ is a complexity penalty.

The maximized likelihood $\hat{P}(D \mid G)$ (Eq. (3)) factorizes by the network structure, and for the decomposable scores discussed in this paper, the complexity penalty can also be factorized. Hence, we can write the penalized scores in the factorized form (4), with the local scores given by

$$S(D_i, D_{G_i}) = \log \hat{P}(D_i \mid D_{G_i}) - \Delta_i(D_i, D_{G_i}). \tag{6}$$

Different scores differ in how the local penalty $\Delta_i(D_i, D_{G_i})$ is determined.

3.1 AIC and BIC

The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are two popular decomposable scores for learning Bayesian network structures. The local penalty terms for these scores are

$$\Delta_i^{\mathrm{BIC}} = \frac{q_i(r_i - 1)}{2} \ln N, \quad \text{and} \quad \Delta_i^{\mathrm{AIC}} = q_i(r_i - 1),$$

where $q_i(r_i - 1)$ is the number of (free) parameters required to specify the conditional distribution of X_i given its parents. If we denote by $k_G = \sum_{i=1}^{m} q_i(r_i - 1)$

the total number of free parameters for structure G, the overall scores become

$$\mathrm{BIC}(G, D) = \hat{P}(D \mid G) - \frac{k_G}{2} \ln N,$$
$$\mathrm{AIC}(G, D) = \hat{P}(D \mid G) - k_G,$$

respectively.

Both of these complexities are independent of the actual data, and only depend on the arities r_i of random variables and the structure of the Bayesian network. These scores do not have any additional user-defined parameters; in this sense they are as objective as the fNML score we propose later.

3.2 Bayesian Dirichlet scores

Bayesian Dirichlet (BD) scores assume that the parameter vectors Θ_{ij} are independent of each other and distributed according to Dirichlet distributions with some hyper-parameter vector $\vec{\alpha}_{ij} = (\alpha_{ij1}, \ldots, \alpha_{ijr_i}) \in \mathbb{R}^{r_i}$. We let $\vec{\alpha} \in \mathbb{R}^{k'_G}$, where $k'_G = \sum_{i=1}^{m} q_i r_i$ is the total number of hyper-parameters, denote the concatenated vector of all the hyper-parameters. The local BD score is given by

$$\begin{aligned} S_{\mathrm{BD}}(D_i, D_{G_i}, \vec{\alpha}) &= \log P(D_i \mid D_{G_i}, \vec{\alpha}) = \sum_{j=1}^{q_i} \log P\left(D_i^{G_i=j} \mid D_{G_i}^{G_i=j}, \vec{\alpha}_{ij}\right) \\ &= \sum_{j=1}^{q_i} \log \int P(D_i^{G_i=j} \mid D_{G_i}^{G_i=j}, \Theta_{ij}) \,\mathrm{Dir}(\Theta_{ij} \,;\, \vec{\alpha}_{ij}) \, d\Theta_{ij} \qquad (7) \\ &= \sum_{j=1}^{q_i} \log \left(\frac{\mathrm{Beta}(\vec{\alpha}_{ij} + \vec{N}_{ij})}{\mathrm{Beta}(\vec{\alpha}_{ij})} \right), \end{aligned}$$

where $\mathrm{Dir}(\Theta_{ij} \,;\, \vec{\alpha}_{ij})$ denotes the Dirichlet density, and Beta is the multinomial Beta function

$$\mathrm{Beta}(\alpha_1, \ldots, \alpha_K) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k'=1}^{K} \alpha_{k'})}.$$

With all $\alpha_{ijk} = 1$, we get a K2-score [21], and with $\alpha_{ijk} = \alpha/(q_i r_i)$ we get a family of BDeu scores popular for giving equal scores to different Bayesian network structures that encode the same independence assumptions. BDeu scores depend only on a single parameter, the *equivalent sample size* α. Recent studies on the role of this parameter show that network learning under BDeu is very sensitive to this parameter [11,12].

For comparison, we can write the BD-score as a penalized maximized likeli-

hood with penalty

$$\Delta_i^{BD}(D_i, D_{G_i}) = \sum_{i=i}^{q_i} \log \left(\frac{\hat{P}(D_i^{G_i=j} \mid D_{G_i}^{G_i=j})}{P(D_i^{G_i=j} \mid D_{G_i}^{G_i=j}, \vec{\alpha}_{ij})} \right).$$

This penalty is always non-negative since the maximized likelihood is always at least as great as any convex combination of the individual likelihoods (see Eq. (7)). The BD penalty is data-dependent and it is controlled by the hyperparameters α_{ijk}. The asymptotic behavior is well studied [7]. However, when learning Bayesian networks, the data parts $D_i^{G_i=j}$ are often very small, which makes the asymptotic results less useful.

3.3 Factorized NML

The *factorized normalized maximum likelihood* (fNML) score is based on the *normalized maximum likelihood* (NML) distribution [22,23]. The NML distribution for the model class $\mathcal{M}$ (which may or may not be a Bayesian network) is the unique distribution solving the minimax problem

$$\min_{Q} \max_{D'} \frac{\hat{P}(D' \mid \mathcal{M})}{Q(D' \mid \mathcal{M})}, \tag{8}$$

where Q ranges over all distributions.

As originally shown by Shtarkov [22], the solution of the above minimax problem is given by

$$P_{\mathrm{NML}}(D \mid \mathcal{M}) = \frac{\hat{P}(D \mid \mathcal{M})}{\sum_{D'} \hat{P}(D' \mid \mathcal{M})}, \tag{9}$$

where the normalization is over all data sets D' of the same size $N = |D|$. The log of the normalizing factor is called *parametric complexity* or *regret* [1]. The NML distribution is a central concept in modern minimum description length (MDL) methods, see [7,20].

Evaluation of the normalizing sum is often hard due to exponential number of terms in the sum. Currently, there are tractable formulas for only a handful of models; for examples, see [7]. In the case of a single r-ary multinomial variable and the sample size N, the normalizing sum is given by

$$\mathcal{C}_N^r = \sum_{k_1+k_2+\ldots+k_r=N} \frac{N!}{k_1!\, k_2! \cdots k_r!} \prod_{j=1}^{r} \left(\frac{k_j}{N} \right)^{k_j}, \tag{10}$$

[1] In general the term regret is used to describe the loss to the post-hoc optimal model, i.e., $regret(P, D, \mathcal{M}) := \log P(D \mid \mathcal{M}) - \log \hat{P}(D \mid \mathcal{M})$.

where the sum goes over all non-negative integer vectors $(k_j)_{j=1}^r$ that sum to N. A linear-time algorithm for the computation of $\mathcal{C}_N^r$ was introduced recently in [24].

Given a data set D, the NML model selection criterion proposes to choose the model $\mathcal{M}$ for which the $P_{\mathrm{NML}}(D \mid \mathcal{M})$ is largest. After taking the logarithm the score is in a form of penalized log-likelihood,

$$\log P_{\mathrm{NML}}(D \mid \mathcal{M}) = \log \hat{P}(D \mid \mathcal{M}) - \log \sum_{D'} \hat{P}(D' \mid \mathcal{M});$$

the complexity penalty can be interpreted as a measure of how well the model can fit datasets D' of size N *on the average*.

Because of the score equivalence of the maximum likelihood score, the NML score is score equivalent as well. However, it can be shown *not* to be decomposable. Sacrificing the score equivalence, we propose a decomposable version of this score, which penalizes the complexity locally similarly to the other decomposable scores. Specifically, we propose the local score

$$S_{\mathrm{fNML}}(D_i, D_{G_i}) = \log P_{\mathrm{NML}}(D_i \mid D_{G_i}) = \log \left(\frac{\hat{P}(D_i \mid D_{G_i})}{\sum_{D_i'} \hat{P}(D_i' \mid D_{G_i})} \right), \qquad (11)$$

where the normalizing sum goes over all the possible D_i-column vectors of length N, i.e., $D_i' \in \{1, \ldots, r_i\}^N$.

Since equation (11) defines a (log-) conditional distribution for the data column D_i, adding these local scores together yields a total score that defines a distribution for the whole data. In this sense fNML can be seen as an alternative way to define the marginal likelihood (or *evidence*) for the data

$$\log P_{\mathrm{fNML}}(D \mid G) = \sum_{i=1}^{m} \log P_{\mathrm{NML}}(D_i \mid D_{G_i}).$$

At the same time, combining the local scores yields an enumerator that equals the factorization of the maximum likelihood, thus the whole score can be seen as a penalized maximum log-likelihood with local (data-dependent) penalties

$$\Delta_i^{\mathrm{fNML}}(D_{G_i}) = \log \sum_{D_i'} \hat{P}(D_i' \mid D_{G_i}). \qquad (12)$$

The following observation follows from the factorization of the maximum likelihood by the parent configurations, and it is crucial for efficient calculation of the local penalty term.

Theorem 1. The local penalty of fNML can be expressed in terms of multi-

nomial normalizing constants

$$\Delta_i^{\mathrm{fNML}}(D_{G_i}) = \sum_{j=1}^{q_i} \log \mathcal{C}_{N_{ij}}^{r_i},$$

where $\mathcal{C}_{N_{ij}}^{r_i}$ is the normalizing constant of NML for an r_i-ary multinomial model with sample size N_{ij}.

Proof. The penalty is defined as the sum of maximized likelihoods over all possible column vectors D_i':

$$\sum_{D_i'} \hat{P}(D_i' \mid D_{G_i}) = \sum_{D_i'} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \left(\frac{N_{ijk}'}{N_{ij}} \right)^{N_{ijk}'},$$

see Eq. (2), where the maximum likelihood parameters $\hat{\theta}_{ijk} = N_{ijk}'/N_{ij}$ are substituted for θ_{ijk}, and N_{ijk}' denotes the number of times the parent configuration j co-occurs in D_{G_i} together with the occurrence of value k in D_i'. In the sum, rows (terms) with parent configuration j are independent of all the other rows with some other configuration, j', and hence we can switch the order of the product and summation to get [2]

$$\sum_{D_i'} \hat{P}(D_i' \mid D_{G_i}) = \prod_{j=1}^{q_i} \sum_{D_i'^{D_{G_i}=j}} \prod_{k=1}^{r_i} \left(\frac{N_{ijk}'}{N_{ij}} \right)^{N_{ijk}'} = \prod_{j=1}^{q_i} \mathcal{C}_{N_{ij}}^{r_i}.$$

Taking the logarithm on both sides concludes the proof. □

Theorem 1 makes it possible to implement the calculation of the fNML model selection criterion as efficiently as other decomposable selection criteria for Bayesian networks [25]. However, it should be noted that the model search problem remains difficult as the parent assignment problem (i.e., choosing the best parent set for a variable) is known to be NP-hard with all popular scores, including BDeu, AIC, BIC, NML and fNML [26] .

To conclude this section we show that asymptotically, and under mild regularity conditions, the fNML score belongs to the (large) class of BIC-like scores that are consistent. Other scores in this class include most Bayesian and MDL criteria. The regularity conditions required for BIC-like behavior typically exempt a measure zero set of generating parameters, such as the boundaries of the parameter simplex. The following theorem gives sufficient conditions on the penalty term that guarantee consistency for exponential family models.

[2] Recall that the notation $D_i'^{D_{G_i}=j}$ refers to the i'th column of the rows where the parent configuration is given by j.

The theorem requires that the number of candidate models is finite, which is always true in the case of Bayesian networks when the number of nodes is limited.

Theorem 2. For (curved) exponential families, if data is generated by an i.i.d. distribution p, and the penalty term is given by [3] $\frac{1}{2}\,k \log N + \mathcal{O}_p(1)$, where k is the number of parameters then, asymptotically, the model containing p that has the least number of parameters will be chosen with p-probability tending to one as the sample size N grows.

Proof. The proof is very similar to the proof of Prop. 1.2 of [27] (see also Remark 1.2 therein); the main difference is that while in [27], the penalty term is defined by a fixed sequence that is the same for all models (except of course for the factor k), in our case the penalty terms are random and may depend on the model. The proof consists to two parts. Assume first that a model G_1 with k_{G_1} parameters *does not* contain the true distribution p, and that another model, G_2, with k_{G_2} parameters *does* contain p. Then we find that there is an $\epsilon > 0$ such that

$$\log \hat{P}(D \mid G_1) + N\frac{\epsilon}{2} < \log \hat{P}(D \mid G_2)$$

with p-probability tending to one, as $N \to \infty$ [27]. Hence any penalty term that grows sublinearly in N (such as $\frac{1}{2}\,k \log N$) is eventually dominated by the $N\epsilon/2$ difference in the log-likelihoods, and the correct model G_2 is chosen. Secondly, assume that contrary to the first part of the proof, both models G_1 and G_2 contain the true distribution p, but we have $k_{G_1} > k_{G_2}$. Then, following again the proof of Prop. 1.2 in [27], we find that

$$\left|\log \hat{P}(D \mid G_1) - \log \hat{P}(D \mid G_2)\right| = \mathcal{O}_p(1),$$

i.e., the difference between the maximized log-likelihoods is bounded in the limit in probability. Hence, the difference between the penalty terms, which is of order $\frac{1}{2}(k_{G_1} - k_{G_2}) \log n$, dominates, and the simpler of the two models is chosen eventually with p-probability tending to one. From these two cases, it follows that among a finite set of candidates the simplest of the models containing p is eventually chosen. □

Since Bayesian networks are curved exponential families [28,29], it now remains to prove that the penalty term of fNML satisfies this property.

[3] The notation $f(N) = \mathcal{O}_p(1)$ indicates that the left-hand side is bounded in the limit in probability, i.e., that for any $\epsilon > 0$, there is a constant $M > 0$, such that eventually $\Pr[|f(N)| > M] < \epsilon$ as $N \to \infty$.

Theorem 3 (Asymptotically fNML behaves like BIC)**.** Assuming that the maximum likelihood parameters are asymptotically bounded away from the boundaries of the parameter simplex, the local penalty of fNML behaves as

$$\Delta_i^{\mathrm{fNML}}(D_{G_i}) = \frac{q_i(r_i - 1)}{2} \log N + \mathcal{O}(1),$$

almost surely, where the $\mathcal{O}(1)$ term is bounded by a constant independent of N.

Proof. By Thm. 1, the local penalty is a sum of logarithms of multinomial normalizing constants, $\log \mathcal{C}_{N_{ij}}^{r_i}$. The logarithms of the constants follow, in turn, by Thm. 1 in [23], under suitable conditions on the model class, the asymptotic form $\frac{k}{2} \log \frac{N_{ij}}{2\pi} + \ln \int \sqrt{|I(\theta)|}\, d\theta + o(1)$, where $k = r_i - 1$ is the number of parameters, and $I(\theta)$ is the Fisher information matrix. The required conditions hold for the multinomial model, and further, the value of the Fisher information integral is known and finite; for both these results, see e.g. [30]. Hence, we get for the normalizing constants the approximation [4]

$$\log \mathcal{C}_{N_{ij}}^{r_i} = \frac{r_i - 1}{2} \log N_{ij} + \mathcal{O}(1). \tag{13}$$

Under the assumption that the maximum likelihood parameters are bounded away from the boundaries, the strong law of large numbers implies that the counts N_{ij} grow linearly in the total sample size N almost surely, i.e., $N_{ij}/N = \eta + o(1)$ a.s. for some $0 < \eta < 1$. Taking logarithms on both sides yields

$$\log N_{ij} = \log N + \mathcal{O}(1) \quad \text{a.s.} \tag{14}$$

Plugging (14) into (13), and adding together the q_i terms yields the result. □

The total fNML penalty becomes then

$$\begin{aligned} \Delta^{\mathrm{fNML}}(D) &= \sum_{i=1}^{m} \Delta_i^{\mathrm{fNML}}(D_{G_i}) \\ &= \sum_{i=1}^{m} \frac{q_i(r_i - 1)}{2} \log N + \mathcal{O}(1) = \frac{1}{2}\, k \log N + \mathcal{O}(1) \quad \text{a.s.}, \end{aligned} \tag{15}$$

where $q_i(r_i - 1)$ is the number of parameters (associated with the ith variable). The almost sure convergence in (15) implies the convergence in probability required in Thm. 2, and hence, fNML is consistent.

[4] Note that here the convergence happens surely, without any probabilistic qualifications since the normalizing constant $\mathcal{C}_N^{r_i}$ is not a random variable. (The counter N_{ij} is random, but in (13) the statement holds for an increasing sequence of N_{ij}.)

4 Prediction

The scoring methods described in the previous section can be used for selecting the best Bayesian network structure. However, much of the appeal of the Bayesian networks rests on the fact that *with the parameter values instantiated*, they define a joint probability distribution that can be used for probabilistic inference. For that reason, the structure selection is usually followed by a parameter learning phase. Next we will first review the standard Bayesian solution, and then in Section 4.2 introduce our new information-theoretic parameter learning scheme.

4.1 Bayesian Parameter Selection

In general, the Bayesian answer for learning the parameters amounts to inferring their posterior probability distribution. Consequently, the answer to determining the predictive probability

$$P(d_{\text{new}} \mid D, G) = \int P(d_{\text{new}} \mid \theta, G) P(\theta \mid D, G) d\theta$$

avoids selecting any particular parameter values. The actual calculation of the integral can be hard, but with the assumptions behind the BDeu score, the task becomes trivial since the predictive posterior probability of a new vector coincides with its probability calculated using the a posteriori expected parameter values

$$\tilde{\theta}_{ijk}^{BD} = \frac{N_{ijk} + \alpha_{ijk}}{\sum_{k'=1}^{r_i} [N_{ijk'} + \alpha_{ijk'}]}.$$

This choice of parameters can be further backed up by a prequential model selection principle: since the BDeu score is just a marginal likelihood $P(D \mid G, \alpha)$, it can be expressed as a product of predictive distributions

$$P(D \mid G, \alpha) = \prod_{n=1}^{N} P(d_n \mid D^{n-1}, \alpha) = \prod_{n=1}^{N} P(d_n \mid \tilde{\theta}(D^{n-1}, \alpha)),$$

where $D^{n-1} = (d_1, \ldots, d_{n-1})$ denotes the first $n-1$ rows of D. Since we have selected the structure that has the strongest predictive record when using the expected parameter values, it is very natural to continue using the expected parameter values after the selection.

4.2 Sequential NML Parameter Selection

Having proposed a non-Bayesian method for structure learning, it would be intellectually dissatisfactory to fall back to the Bayesian solution in the parameter learning task — in particular, as the Bayesian solution again depends on the hyperparameters. Hence, in accordance with the information-theoretic approach we introduce a solution to the parameter learning task based on a minimax criterion.

The so called *sequential NML* model [31,19] is similar in spirit to the factorized NML model in the sense that the idea is to obtain a joint likelihood as a product of locally minimax (regret) optimal models. In sNML, the normalization is done separately for each observation (vector) in a sequence:

$$P_{\mathrm{sNML}}(D \mid \mathcal{M}) = \prod_{n=1}^{N} \frac{\hat{P}(d_n, D^{n-1} \mid \mathcal{M})}{\sum_{d'} \hat{P}(d', D^{n-1} \mid \mathcal{M})}, \tag{16}$$

where $\mathcal{M}$ is the model class with which the maximized likelihoods are defined. For Bayesian networks family, for instance, the $\mathcal{M}$ would be a network structure G. In the following, we will mainly discuss the multinomial case, where each d_n is a single categorical datum — in Sec. 4.3, the Bayesian network case will be reduced to a collection of multiple multinomials.

One advantage of a row-by-row normalization is that it immediately leads to a natural prediction method: having seen a data-matrix of size $(N-1) \times m$, we can use the locally minimax optimal model for the N'th observation vector, obtained as the N'th factor in the product (16), as a predictive distribution.

That sNML gives a good predictive method can be demonstrated by showing that predicting with it never yields much worse a result than predicting the data while taking advantage of knowledge of the post-hoc optimal parameter value(s).

For a simple Bernoulli model implies a neat bound on the regret of sNML.

Theorem 4 ([32])**.** For the Bernoulli model [5], a result by Takimoto and Warmuth [32] , the regret $R_{\mathrm{sNML}}(D, N, 2)$ of any binary sequence D of length N is upper-bounded by

$$R_{\mathrm{sNML}}(D, N, 2) := \log \hat{P}(D) - \log P_{\mathrm{sNML}}(D) \leq \frac{1}{2} \log(N+1) + \frac{1}{2}.$$

[5] The number 2 in $R_{\mathrm{sNML}}(D, N, 2)$ denotes that there are two categories in the data.

This is better than, for instance, what can be obtained by either the Laplace predictor, i.e., mixture with uniform prior, or the Krichevsky-Trofimov prediction, i.e., mixture with Dirichlet$(1/2, \ldots, 1/2)$ prior, see [32].

For a categorical datum with K different values, the following bound can be obtained.

Theorem 5. For any categorical (discrete) data D of length N, the regret of the sNML model is upper-bounded by

$$R_{\mathrm{sNML}}(D, N, K) \leq \frac{1}{K} \sum_{k=1}^{K-1} N \log \frac{N+k}{N} + k \log \frac{N+k}{k}.$$

We give an elementary proof of this statement in Appendix C. A relaxed version of the bound is as follows:

$$R_{\mathrm{sNML}}(D, N, K) \leq (K-1) \left[\frac{K-1}{K} \log \left(\frac{N}{K-1} + 1 \right) + \frac{1}{2} \right];$$

for $K = 2$, this agrees with the binary case above.

In theory, using sNML for determining a predictive distribution $P(d \mid D, G)$ would be straightforward. Furthermore, since the fNML was introduced as a computationally feasible version of the NML, we would still want to use a prediction scheme based on NML, thus the sNML would be a natural choice. In practice, however, using sNML for Bayesian networks faces two major problems. Firstly, it is not computationally feasible to calculate the normalizing term (at least in the naïve way), since the number of possible values of a single data vector may be prohibitively large. Secondly, we set ourselves to learn the parameters for the selected Bayesian network, and it turns out that the predictive distribution $P_{\mathrm{sNML}}(d \mid D, G)$ cannot necessarily be obtained with any parametrization of the structure G (see Appendix A for a counter-example). In the Bayesian case, the predictive probability can be obtained with the expected parameter values, but for NML we have no such luck.

4.3 Factorized Sequential NML

$P_{\mathrm{sNML}}(d \mid D, G)$ did not directly offer us a method for determining the model parameters. On the other hand, Bayesian expected parameters can be interpreted as predictive probabilities for a one-dimensional categorical datum:

$$\theta_{ijk}^{BD} = P(d_{\mathrm{new},i} = k \mid D_i^{G_i=j}, G, \alpha_{ij}),$$

where $d_{\mathrm{new},i}$ denotes the value of the ith variable in the predicted vector. In analogy to this, we propose to use the corresponding sNML predictive

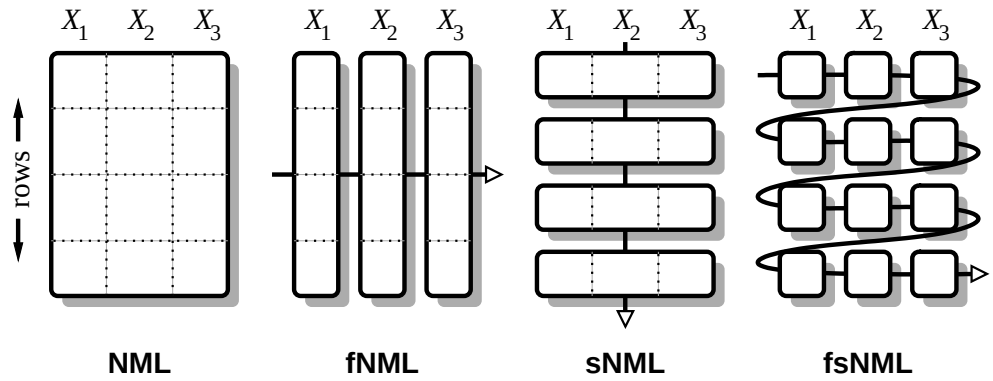


Fig. 1. A schematic illustration of alternative ways to obtain minimax optimal models by normalizing the maximized likelihood $\hat{P}(D \mid G)$. *left to right:* In NML, the normalization is done over the whole data matrix in one go. In factorized NML (fNML), each column is normalized separately. In sequential NML (sNML), each row is normalized separately. In factorized-sequential NML (fsNML), the normalization is done entry-by-entry, in either the row or column order (the result is the same either way).

probability distribution to set the parameters, i.e,

$$\theta_{ijk}^{\text{fsNML}} = P_{\text{sNML}}(d_{\text{new},i} = k \mid D_i^{G_i=j}, G).$$

We call this approach *factorized sequential NML* (see Figure 1). For categorical data this yields a spiced-up version of the Laplace's rule of succession

$$\theta_{ijk} = \frac{e(N_{ijk})(N_{ijk}+1)}{\sum_{k'=1}^{r_i} e(N_{ijk'})(N_{ijk'}+1)},$$

where $e(0) = 1$, and otherwise $e(n) = (\frac{n+1}{n})^n \to e$ as n grows.

This selection of parameters defines a joint probability distribution in a similar spirit as P_{fNML}:

$$P_{\text{fsNML}}(D \mid G) = \prod_{i=1}^{m} \prod_{j=1}^{q_i} P_{\text{sNML}}(D_i^{G_i=j}),$$

where the probability $P_{\text{sNML}}(D_i^{G_i=j})$ is given by (16) for univariate categorical data with r_i different values. In contrast with NML, where normalization is done over the whole data matrix in a single, huge summation, or sNML, where normalization is done over data vectors of length m, the normalization in fsNML is very simple since it can be carried out a single entry at a time.

Theorem 6. Given a Bayesian network structure G, the regret of the fsNML

distribution for any $N \times m$ data matrix D is upper-bounded by

$$\begin{aligned} R_{\text{fsNML}}(D, N, G) &:= \log \hat{P}(D \mid G) - \log P_{\text{fsNML}}(D \mid G) \\ &\leq \sum_{i=1}^{m} q_i \bar{R}_{\text{sNML}}\left(N/q_i, r_i\right), \end{aligned}$$

where q_i and r_i denote the number of parent configurations and the arity of variable X_i, respectively, and $\bar{R}_{\text{sNML}}(\frac{N}{q_i}, r_i)$ is the worse case univariate regret $[\bar{R}_{\text{sNML}}(N', K) = \max_{D'} R_{\text{sNML}}(D', N', K)]$ bounded by Thm. 5.

Proof. Since both $\hat{P}(D \mid G)$ and $P_{\text{fsNML}}(D \mid G)$ factorize similarly, we have

$$\begin{aligned} R_{\text{fsNML}}(D, N, G) &= \log \hat{P}(D \mid G) - \log P_{\text{fsNML}}(D \mid G) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{q_i} \left[\log \hat{P}(D_i^{G_i=j}) - \log P_{\text{sNML}}(D_i^{G_i=j})\right] \\ &= \sum_{i=1}^{m} \sum_{j=1}^{q_i} R_{\text{sNML}}(D_i^{G_i=j}, N_{ij}, r_i) \\ &\leq \sum_{i=1}^{m} \sum_{j=1}^{q_i} \bar{R}_{\text{sNML}}(N_{ij}, r_i). \end{aligned}$$

The proof of Thm. 5 in Appendix C actually shows that the bound for $\bar{R}_{\text{sNML}}(N_{ij}, r_i)$ is tight. The $\bar{R}_{\text{sNML}}$ is convex, and since we have $\sum_j N_{ij} = N$, the maximum of the innermost sum above occurs when all the N_{ij} are equal to N/q_i, thus we have

$$\begin{aligned} R_{\text{fsNML}}(D, N, G) &\leq \sum_{i=1}^{m} \sum_{j=1}^{q_i} \bar{R}_{\text{sNML}}(N_{ij}, r_i) \\ &\leq \sum_{i=1}^{m} q_i \bar{R}_{\text{sNML}}(\frac{N}{q_i}, r_i). \end{aligned}$$

□

5 Experiments

It is not obvious how to compare different criteria for learning Bayesian network structures. If the data is generated from a Bayesian network, one might say that the task is to recover the data generating network, but if the generating network is complex, and the sample size is small, it may be more rational to pick a simpler model than the "correct" one. This simplicity requirement is often backed up by arguments about the prediction, or *generalization*, capability of the model. However, it is not always clear how the network structure should be used for prediction.

We divide our experiments in two parts. First, we estimate how well different criteria manage to identify the network structure, when a fixed structure is used to generate artificial data. In the second part, we evaluate the predictive accuracy of the learned networks by complementing the structural learning methods with the corresponding method for learning the parameters. In this case, we use real data from the UCI repository [33].

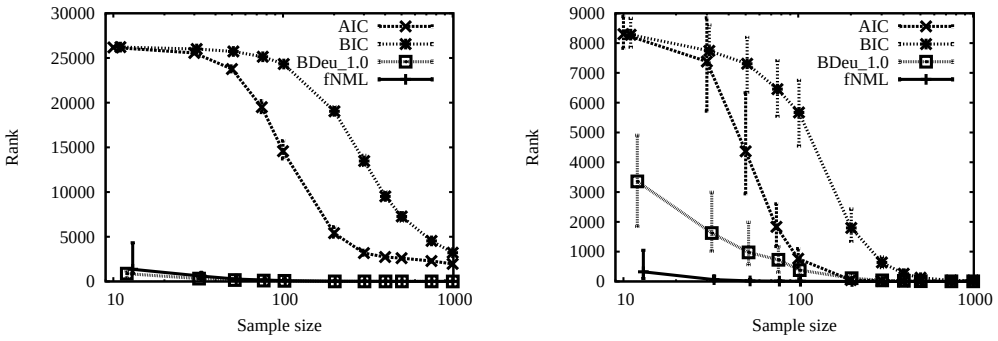
5.1 Model Selection

We first generated data from different networks with five nodes, and then studied how the generating network structures were ranked among all the possible networks by different scoring criteria.

We generated 100 different 5-node Bayesian network structures with 4 edges and another 100 structures with 7 edges. The variables were randomly assigned to have between two to four values ($r_i \in \{2, 3, 4\}$). For each network, we generated parameters by two different schemes. The first scheme exactly matched the assumptions of the BDeu score with $\alpha = 1.0$, i.e., the parameters were distributed according to $\theta_{ij} \sim Dir(\frac{1}{r_i q_j}, \ldots, \frac{1}{r_i q_j})$. The other scheme was to generate the parameters independently from a Dirichlet distribution $\theta_{ij} \sim \mathrm{Dir}(1/2, \ldots, 1/2)$. This distribution was selected for two reasons: first, compared to the uniform distribution, the $\mathrm{Dir}(1/2, \ldots, 1/2)$ prior puts more mass near the boundaries of the parameter space and therefore, makes the generating structure more identifiable, and secondly, it has a special role in information theory as the "least favorable" prior of the multinomial model, see e.g. [34]. From the minimax regret point of view, it is reasonable to assume that the mixture with $\mathrm{Dir}(1/2, \ldots, 1/2)$ prior is similar to the NML (and especially fNML) distribution, see [35].

For each network (structure + parameters), we generated 100 data sets of 1000 data vectors, and studied how different scoring criteria ranked the structure of the generating network among all the 5-node networks as a function of (sub)sample size.

Not surprisingly, the results indicate that when parameter generation mechanism matches the assumptions of the BDeu(1.0) score, BDeu(1.0) usually also ranks the generating structure higher than the other scores (Figure 2(a)). However, fNML behaves very similarly. The density of the network (4 vs. 7 edges) is not a very significant factor. If anything, the similar behavior of fNML and BDeu(1.0) is more pronounced in networks with 7 edges (not shown in the figure). For the parameter-free scores AIC and BIC, the underfitting tendency of BIC can be clearly detected whereas AIC tends to rank the generating network higher. Qualitatively these two scores seem to behave similarly to each



(a) BDeu(1.0) scheme (b) Dir$(1/2, \ldots, 1/2)$ scheme

Fig. 2. The rank of the true structure (lower is better) for different scoring criteria as a function of sample size when the parameters for a 5-node, 7-edge network were generated by the BDeu(1.0) and Dir$(1/2, \ldots, 1/2)$ schemes. The lines show the median over 100 repetitions, error bars indicate upper and lower quartiles.

other.

Switching the parameter generation scheme to independent Dirichlets with $\alpha_{ijk} = \frac{1}{2}$ usually also switches the ranking ability of fNML and BDeu, while the behavior of AIC and BIC stays mostly unaffected. For example, the results of Figure 2(b) were obtained using the same network *structures* as Figure 2(a). Only the parameter generation scheme was changed from BDeu(1.0) to Dir$(1/2, \ldots, 1/2)$. For dense networks fNML often appears as a clear winner.

5.2 Prediction

In order to evaluate the predictive accuracy of the methods, we selected 20 UCI data sets with less than 20 variables, so that we can use exact structure learning algorithms [18] that eliminate the uncertainty due to the heuristic search for the best structure. We then compared our method, the fNML-based structure learning and fsNML parametrization, with the state-of-the-art Bayesian method, the BDeu score and expectation parameters [6]. The equivalent sample size hyperparameter α for the Bayesian learning was set to 1.0. We also included a Bayesian score BD1/2, where both the structure learning and the parameter learning were conducted by setting all hyperparameters $\alpha_{ijk} = 1/2$.

The comparison was done by creating 100 random train and test splits (50%–50%) of each data set, and then using each training data set for learning three Bayesian networks, one with each method. The Bayesian networks were then used to determine the predictive probability $P(d_{\text{new}} \mid G, \Theta)$ for each vector in

[6] We have omitted AIC and BIC from these experiments, since it is not clear how the network structures selected by them should be used for prediction

the test data.

Table 1
Summary of the prediction experiment.

Data	N	m	$\bar{r}_i$	fNML	BDeu1.0	BD1/2
abalone	4177	9	3.0	2.350 ± 0.019	$\mathbf{2.346 \pm 0.020}$	2.370 ± 0.020
adult	32561	15	7.9	2.620 ± 0.015	$\mathbf{2.588 \pm 0.014}$	2.647 ± 0.014
balance	625	5	4.6	$\mathbf{4.347 \pm 0.018}$	4.437 ± 0.039	4.385 ± 0.039
bc	286	10	4.3	$\mathbf{3.991 \pm 0.076}$	4.429 ± 0.103	4.016 ± 0.103
bc wisc	699	11	2.9	$\mathbf{3.493 \pm 0.022}$	3.542 ± 0.025	3.503 ± 0.025
diabetes	768	9	2.9	8.987 ± 0.006	9.207 ± 0.318	$\mathbf{8.962 \pm 0.318}$
ecoli	336	8	3.4	$\mathbf{2.219 \pm 0.001}$	2.220 ± 0.001	2.222 ± 0.001
glass	214	11	3.3	$\mathbf{9.636 \pm 0.095}$	10.586 ± 0.090	9.697 ± 0.090
heart cl	303	14	3.1	$\mathbf{4.697 \pm 0.044}$	4.827 ± 0.148	4.822 ± 0.148
heart hu	294	14	2.6	8.687 ± 0.099	9.105 ± 0.034	$\mathbf{8.678 \pm 0.034}$
heart st	270	14	2.9	$\mathbf{9.241 \pm 0.085}$	9.877 ± 0.079	9.273 ± 0.079
iris	150	5	3.0	$\mathbf{3.718 \pm 0.002}$	3.746 ± 0.002	3.722 ± 0.002
liver	345	7	2.9	$\mathbf{4.539 \pm 0.015}$	4.607 ± 0.016	4.540 ± 0.016
page blks	5473	11	3.2	$\mathbf{8.407 \pm 0.111}$	8.917 ± 0.218	8.577 ± 0.218
post op	90	9	2.9	1.679 ± 0.000	$\mathbf{1.677 \pm 0.000}$	1.680 ± 0.000
shuttle	58000	10	3.0	$\mathbf{5.095 \pm 0.005}$	5.122 ± 0.006	5.107 ± 0.006
thyroid	215	6	3.0	$\mathbf{6.940 \pm 0.003}$	7.035 ± 0.017	6.941 ± 0.017
tic tac	958	10	2.9	$\mathbf{7.197 \pm 0.048}$	7.472 ± 0.045	7.209 ± 0.045
wine	178	14	3.0	$\mathbf{5.314 \pm 0.081}$	6.277 ± 0.318	5.413 ± 0.318
yeast	1484	9	3.7	$\mathbf{9.906 \pm 0.001}$	9.990 ± 0.001	9.912 ± 0.001

The results of the predictive experiment are presented in Table 1. For each data set, the table lists the number of data vectors N, the number of variables m, the average number of values per variable ($\bar{r}_i$), and for each method the average and the $1.96\times$ standard deviation of hundred numbers (one for each train-and-test split of the data), each of which is the average of the negative logarithms of the predictive probabilities $P(d_{\text{new}} \mid D)$ obtained by the method.

For example, the marking 9.636 ± 0.095 for the data set *glass* and the method fNML was obtained as the average and $1.96\times$ standard deviation of hundred

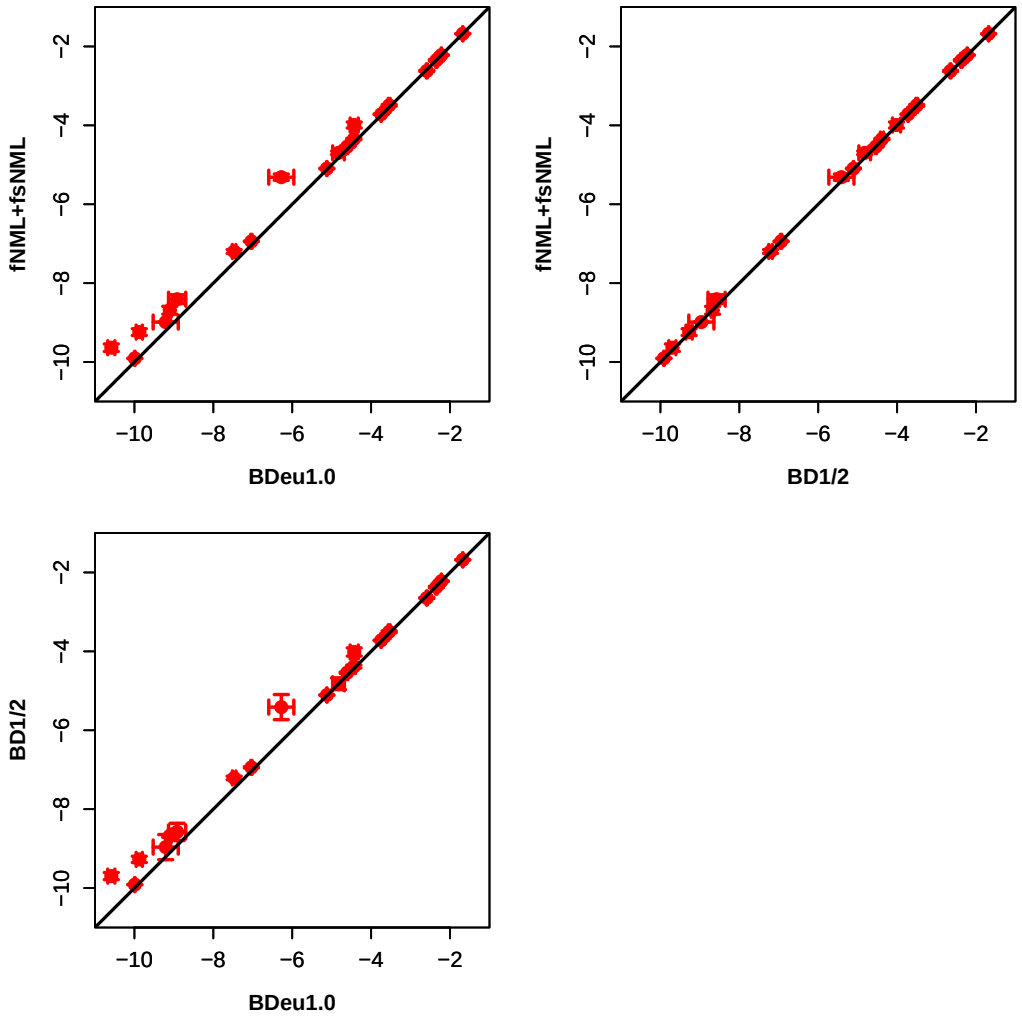


Fig. 3. Visualization of the prediction experiment. Each graph show the predictive accuracies obtained with two methods, indicated by the horizontal and vertical labels, in terms of average log-likelihood per data vector (greater values are better). Error-bars show $\pm 1.96\times$ standard deviation over 100 random train-test splits. Points above the diagonal line represent cases where the method shown on the vertical axis performs better.

numbers $(s_1, s_2, \ldots, s_{100})$, where each s_i was calculated by using the i^{th} random partition of the $glass$ data $(glass_i^{\text{train}}, glass_i^{\text{test}})$

$$s_i = \frac{1}{|glass_i^{\text{test}}|} \sum_{d \in glass_i^{\text{test}}} -\log P_{\text{fNML}}(d \mid glass_i^{\text{train}}).$$

The predictive distribution P_{fNML} was obtained by selecting the optimal structure using the fNML selection criterion, and then parametrizing the selected structure by using the fsNML parameters.

In 15 data sets (out of 20) the NML-based method predicted better than the

other methods, and never did it predict much worse. In almost all cases, the difference between fNML+fsNML and the BD1/2 method is very small. The results are shown graphically in Fig. 3. It also worth noticing that the good performance of the fNML+fsNML did not come at an expense of increased variance: 11 times (out of 20) our NML based method had a smaller variance across the train-and-test splits than other methods, and only 5 times the variance was larger, the other 4 times ending in a tie.

6 Discussion — On Likelihood Equivalence and (Again) Priors

Based on our results, it seems that minimax criteria lead to methods that are competitive both in terms of structure learning and prediction, and, moreover, robust with respect to changes in the parameter distribution. While this may have been expected, the results also raise several questions.

For instance, it is curious that while our starting point, the regular NML model, is likelihood equivalent — graphs encoding the same conditional independence assertions get the same score — the practical fNML and fsNML criteria are *not* likelihood equivalent, and neither are the Bayesian Dirichlet scores where the hyper-parameters are constant, such as the BD1/2 criterion. These non-likelihood equivalent methods seem to outperform the BDeu criterion. In fact, this appears to be the case even when the equivalent sample size parameter is optimized with hind-sight [25]. *Does this suggest that likelihood equivalence is not necessarily a desirable property?* In the current literature, it is almost universally accepted even if some authors have drawn conclusions similar to ours [9,36]. Or is it just the BDeu score (which was thought to be the state-of-the-art) that is inferior? More extensive experiments with other scores, including the NML score in cases where it is computationally feasible, will hopefully help to resolve the question.

Perhaps already pointing towards an answer, our initial experiments (not reported here) show that even if we drop the requirement of likelihood equivalence, and set all the the Dirichlet parameters α_{ijk} to some constants, model selection is still very sensitive to *which constants* we choose. In this light it seems likely that the problem may be less due to likelihood equivalence than to the already blamed parameter sensitivity. While in theory, it can be argued that priors don't matter too much, except for very small sample sizes, Bayesian networks learned from multidimensional data can easily have parameters with very little, if any, support in the data. *In practice, we have almost always very little data*, and therefore, *priors matter.*

It is sometimes asked whether the NML corresponds to some special prior for the parameters. Strictly speaking the answer is no: the fact that NML does

not define a stochastic process is a proof of this, see e.g. [31]. However, there are priors that are asymptotically "almost" minimax, and hence, necessarily very similar to NML which is exactly minimax, see [35,7]. In particular, the $\mathrm{Dir}(1/2, \ldots, 1/2)$ prior is an important example in the multinomial case. In fact, our results confirm that the BD1/2 method, which is based on this prior, has similar robustness properties as our fNML/fsNML methods. So it is not our claim that Bayesian methods are somehow at fault, but that *it is essential to have a principled way to protect oneself against unreasonable priors.*

As for future work, it can be said that the proposed methods of fNML and fsNML have been derived more by the logic of practicality rather than logic of necessity. A natural, alternative approach for finding the parameters given a Bayesian network structure is based on the idea of normalizing the joint fNML likelihood $P_{\mathrm{fNML}}(d_{\mathrm{new}}, D \mid G)$ to get a predictive distribution. However, it turns out that the resulting distribution may lie outside the set of distributions representable with G, thus no such parametrization is in general possible (Appendix B). There is, however, another candidate that we are planning to study: the minimax optimal predictive distribution conforming to G:

$$P^G_{\mathrm{sNML}}(d_{\mathrm{new}} \mid D) = \operatorname*{argmin}_{q(\cdot) \in G} \max_D \frac{\hat{P}(d_{\mathrm{new}}, D)}{\sum_{d'} \hat{P}(d', D)},$$

where the minimization is over all distributions satisfying the conditional independence assumptions encoded by G. By definition, this distribution can be obtained by parametrizing G. However, it turns out that the minimax condition alone does not yield a unique distribution, but further requirements are needed. Even when P^G_{sNML} is unique, it may be different from the P_{fsNML} which can be easily proved by finding a counter-example: for instance, a graph with two binary connected binary variables, X_0 and X_1, and $D = (d_1) = ((0, 0))$ will prove the point. Currently, there are efficient methods for solving the parameters that yield $P^G_{\mathrm{sNML}}(d)$ for only certain restricted network structures.

7 Conclusions

We have introduced a new probabilistic scoring criterion, the factorized normalized maximum likelihood, for learning Bayesian network structures from data when no background information is available. The score is decomposable, which makes it easy to incorporate it to existing search heuristics and exact structure learning algorithms. We also introduced an associated method for determining the Bayesian network parameters. The theoretical analysis of the methods shows that they lead to consistent model selection and predictions that are never much worse than those obtained by optimizing the parameters with hindsight. Together the methods provides a computationally efficient,

completely objective and parameter-free approach for learning Bayesian networks, which applicable to both small and large data-sets.

Initial empirical tests are promising. We are particularly pleased with the good predictive capabilities of the models learned with our approach: in many cases the predictive accuracy was much better than with the standard BDeu score, and never was it much worse. We argue that the comparative advantage of the new methods over BDeu is due to the strong sensitivity of the latter with respect to the parameter prior, a problem which our non-Bayesian methods avoid. While there are also several open questions for future research, the current results show that the proposed approach offers a theoretically well-founded, robust method for learning Bayesian networks.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[2] F. Jensen, An Introduction to Bayesian Networks, UCL Press, London, 1996.

[3] J. Pearl, Causality: Models, Reasoning and Inference, Cambridge University Press, 2000.

[4] D. Heckerman, A tutorial on learning with Bayesian networks, Tech. Rep. MSR-TR-95-06, Microsoft Research, Advanced Technology Division, One Microsoft Way, Redmond, WA 98052 (1996).

[5] P. Spirtes, C. Glymour, R. Scheines (Eds.), Causation, Prediction and Search, Springer-Verlag, 1993.

[6] J. Cheng, R. Greiner, J. Kelly, D. Bell, W. Liu, Learning bayesian networks from data: An information-theory based approach, Artificial Intelligence J. 137 (1-2) (2002) 43–90.

[7] P. Grünwald, The Minimum Description Length Principle, MIT Press, 2007.

[8] R. G. Cowell, Conditions under which conditional independence and scoring methods lead to identical selection of bayesian network models, in: J. S. Breese, D. Koller (Eds.), UAI, Morgan Kaufmann, 2001, pp. 91–97.

[9] L. M. de Campos, A scoring function for learning bayesian networks based on mutual information and conditional independence tests, Journal of Machine Learning Research 7 (2006) 2149–2187.

[10] W. Buntine, Theory refinement on Bayesian networks, in: B. D'Ambrosio, P. Smets, P. Bonissone (Eds.), Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, 1991, pp. 52–60.

[11] H. Steck, T. S. Jaakkola, On the dirichlet prior and bayesian regularization, in: Advances in Neural Information Processing Systems 15, MIT Press, Vancouver, Canada, 2002, pp. 697–704.

[12] T. Silander, P. Kontkanen, P. Myllymäki, On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter, in: R. Parr, L. van der Gaag (Eds.), Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2007, pp. 360–367.

[13] H. Akaike, Information theory and an extension of the maximum likelihood principle, in: B. Petrox, F. Caski (Eds.), Proceedings of the Second International Symposium on Information Theory, Akademiai Kiado, Budapest, 1973, pp. 267–281.

[14] G. Schwarz, Estimating the dimension of a model, Annals of Statistics 6 (1978) 461–464.

[15] D. Chickering, Learning Bayesian networks is NP-Complete, in: D. Fisher, H. Lenz (Eds.), Learning from Data: Artificial Intelligence and Statistics V, Springer-Verlag, 1996, pp. 121–130.

[16] D. Heckerman, D. Geiger, D. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Machine Learning 20 (3) (1995) 197–243.

[17] M. Koivisto, K. Sood, Exact Bayesian structure discovery in Bayesian networks, Journal of Machine Learning Research 5 (2004) 549–573.

[18] T. Silander, P. Myllymäki, A simple approach for finding the globally optimal Bayesian network structure, in: R. Dechter, T. Richardson (Eds.), Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2006, pp. 445–452.

[19] T. Roos, J. Rissanen, On sequentially normalized maximum likelihood models, in: Workshop on Information Theoretic Methods in Science and Engineering (WITMSE-08), Tampere, Finland, 2008.

[20] J. Rissanen, Information and Complexity in Statistical Modeling, Springer, 2007.

[21] G. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (1992) 309–347.

[22] Y. Shtarkov, Universal sequential coding of single messages, Problems of Information Transmission 23 (1987) 3–17.

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

[24] P. Kontkanen, P. Myllymäki, A linear-time algorithm for computing the multinomial stochastic complexity, Information Processing Letters 103 (6) (2007) 227–233.

[25] T. Silander, T. Roos, P. Kontkanen, P. Myllymäki, Factorized normalized maximum likelihood criterion for learning bayesian network structures, in: Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM-08), Hirtshals, Denmark, 2008, pp. 257–264.

[26] M. Koivisto, Parent assignment is hard for the MDL, AIC, and NML costs, in: Proceedings of the 19th Annual Conference on Learning Theory (COLT-06), 2006, pp. 289–303.

[27] D. Haughton, On the choice of a model to fit data from an exponential family, Annals of Statistics 16 (1) (1988) 342–355.

[28] D. Geiger, D. Heckerman, H. King, C. Meek, Stratified exponential families: graphical models and model selection, Annals of Statistics 29 (2001) 505–529.

[29] D. Chickering, Optimal structure identification with greedy search, Journal of Machine Learning Research 3 (2002) 507–554.

[30] P. Grünwald, Minimum description length tutorial, in: P. Grünwald, I. Myung, M. Pitt (Eds.), Advances in Minimum Description Length: Theory and Applications, The MIT Press, 2006, pp. 23–79.

[31] J. Rissanen, T. Roos, Conditional NML models, in: Proceedings of the Information Theory and Applications Workshop (ITA-07), San Diego, CA, 2007.

[32] E. Takimoto, M. Warmuth, The last-step minimax algorithm, in: Proc. 11th International Conference on Algorithmic Learning Theory, 2000, pp. 279–290.

[33] S. Hettich, C. Blake, C. Merz, UCI repository of machine learning databases, University of California, Irvine, Dept. of Information and Computer Sciences, http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998).

[34] N. Merhav, M. Feder, Universal prediction, IEEE Transactions on Information Theory 44 (6) (1998) 2124–2147.

[35] J. Takeuchi, A. Barron, Asymptotically minimax regret by Bayes mixtures, in: 1998 IEEE International Symposium on Information Theory, Cambridge, MA, 1998.

[36] S. Yang, K.-C. Chang, Comparison of score metrics for bayesian network learning, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 32 (3) (2002) 419–428.

Appendix A

The following example shows that the joint probability distribution

$$P_{\mathrm{sNML}}(d|D,G) = \frac{P(d,D|G,\hat{\theta}(D,d))}{\sum_{d'} P(d',D|G,\hat{\theta}(D,d))}$$

cannot necessarily be presented with any parametrization of the network G.

Let G be a simple v-structure $G = (\{\}, \{X_1, X_3\}, \{\})$, and let the data D consist of just a single 3-dimensional binary-vector $[(0, 0, 0)]$. A direct calculation of $P_{\text{sNML}}(d \mid D, G)$ yields a probability distribution

$P(d\|D)$	$\frac{8}{19}$	$\frac{2}{19}$	$\frac{2}{19}$	$\frac{2}{19}$	$\frac{2}{19}$	$\frac{1}{38}$	$\frac{2}{19}$	$\frac{1}{38}$
d	000	001	010	011	100	101	110	111

In this joint probability distribution $P(X_1, X_3) \neq P(X_1)P(X_3)$:

$P(x_1, x_3\|D)$	$\frac{10}{19}$	$\frac{4}{19}$	$\frac{4}{19}$	$\frac{1}{19}$
x_1x_3	00	01	10	11

,

$P(x_1\|D)P(x_3\|D)$	$\frac{196}{361}$	$\frac{70}{361}$	$\frac{70}{361}$	$\frac{25}{361}$
x_1x_3	00	01	10	11

.

However, all the parametrizations of the structure G yield distributions where X_1 and X_3 are marginally independent, i.e., $P(X_1, X_3) = P(X_1)P(X_3)$.

Appendix B

The following example shows that the joint probability distribution achieved by normalizing P_{fNML},

$$P_{\text{sfNML}}(d \mid D, G) = \frac{P_{\text{fNML}}(d, D \mid G)}{\sum_{d'} P_{\text{fNML}}(d', D \mid G)},$$

cannot necessarily be presented with any parametrization of the network G.

Let G be a simple v-structure $G = (\{\}, \{X_1, X_3\}, \{\})$, and let the data D consist of two 3-dimensional binary-vectors $[(0, 0, 0), (0, 0, 0)]$. A direct calculation of P_{sfNML} yields a probability distribution in which $P(X_1, X_3) \neq P(X_1)P(X_3)$

$P(d \mid D)$	$\frac{32805}{49729}$	$\frac{2808}{49729}$	$\frac{4860}{49729}$	$\frac{2808}{49729}$	$\frac{2808}{49729}$	$\frac{416}{49729}$	$\frac{2808}{49729}$	$\frac{416}{49729}$
d	000	001	010	011	100	101	110	111

.

Appendix C: Proof of Theorem 5

We derive a regret bound for the categorical data of size N with K categories. We start by reviewing the probability distribution of interest

$$P_{\mathrm{sNML}}(D) = \prod_{n=1}^{N} \frac{\widehat{P}(d_n, D^{n-1})}{\sum_{d'} \widehat{P}(d', D^{n-1})},$$

where we have denoted with D^{n-1} the first $n-1$ data items of the sequence D, and with $\widehat{P}(X)$ the maximum likelihood of the data X, $\widehat{P}(X) = P(X|\widehat{\theta}(X))$. We denote with k_{n-1} the number of times the value k appears in D^{n-1}.

To anticipate the comparison of the P_{sNML} with the $\widehat{P}$, we write the $\widehat{P}$ in the form

$$\widehat{P}(D) = \prod_{n=1}^{N} \frac{\widehat{P}(d_n, D^{n-1})}{\widehat{P}(D^{n-1})}.$$

Now we compare the ratio

$$\begin{aligned}
Q(D) &= \frac{\widehat{P}(D)}{P_{\mathrm{sNML}}(D)} \\
&= \prod_{n=1}^{N} \frac{\widehat{P}(d_n, D^{n-1}) \sum_{d'} \widehat{P}(d', D^{n-1})}{\widehat{P}(D^{n-1})\widehat{P}(d_n, D^{n-1})} \\
&= \prod_{n=1}^{N} \frac{\sum_{d'} \widehat{P}(d', D^{n-1})}{\widehat{P}(D^{n-1})} \\
&= \prod_{n=1}^{N} \frac{\sum_{d'} \prod_{k=1}^{K} (\frac{k_{n-1}+[d'=k]}{n})^{(k_{n-1}+[d'=k])}}{\prod_{k=1}^{K} (\frac{k_{n-1}}{n-1})^{k_{n-1}}} \\
&= \prod_{n=1}^{N} \frac{\frac{1}{n^n} \sum_{d'} \prod_{k=1}^{K} (k_{n-1} + [d'=k])^{(k_{n-1}+[d'=k])}}{\frac{1}{(n-1)^{n-1}} \prod_{k=1}^{K} k_{n-1}^{k_{n-1}}} \\
&= \prod_{n=1}^{N} \frac{(n-1)^{n-1}}{n^n} \sum_{k=1}^{K} \frac{(k_{n-1}+1)^{k_{n-1}+1}}{{k_{n-1}}^{k_{n-1}}} \\
&= \prod_{n=1}^{N} \frac{(n-1)^{n-1}}{n^n} \sum_{k=1}^{K} (k_{n-1}+1) e(k_{n-1}),
\end{aligned}$$

where we have used the function $e(x) = (\frac{x+1}{x})^x$ that approaches the real number e from below ($e(0) = 1$) when x grows. The sum within the product obtains it largest when all the k_{n-1} are equal. Therefore we can bound the ratio by

$$
\begin{aligned}
Q(D) &\leq \prod_{n=1}^{N} \frac{(n-1)^{n-1}}{n^n} \sum_{k=1}^{K} (\frac{n-1}{K} + 1) e(\frac{n-1}{K}) \\
&= \prod_{n=1}^{N} \frac{(n-1)^{n-1}}{n^n} (n+K-1) e(\frac{n-1}{K}) \\
&= \prod_{n=1}^{N} \frac{(n-1)^{n-1}}{n^n} (n+K-1) (\frac{n+K-1}{n-1})^{\frac{n-1}{K}} \\
&= \prod_{n=1}^{N} \frac{(n-1)^{(\frac{K-1}{K})(n-1)} (n+K-1)^{\frac{n+K-1}{K}}}{n^n} \\
&= \prod_{n=1}^{N} \frac{(n-1)^{(\frac{K-1}{K})(n-1)}}{n^{\frac{K-1}{K}n}} \frac{(n+K-1)^{\frac{n+K-1}{K}}}{n^{\frac{n}{K}}} \\
&= \frac{1}{N^{N\frac{K-1}{K}}} \frac{\prod_{k=1}^{K-1} (N+k)^{\frac{N+k}{K}}}{\prod_{k=1}^{K-1} k^{\frac{k}{K}}} \\
&= \prod_{k=1}^{K-1} (\frac{N+k}{N})^{\frac{N}{K}} (\frac{N+k}{k})^{\frac{k}{K}}.
\end{aligned}
$$

By taking the logarithm we get a bound for the regret

$$
\begin{aligned}
R(N,K) &= \max_{D} \ln(Q(D)) \\
&\leq \frac{1}{K} \sum_{k=1}^{K-1} \left[\ln(\frac{N+k}{N})^N + \ln(\frac{N+k}{k})^k \right].
\end{aligned}
$$

This concludes the proof.

By noticing that $(\frac{N+k}{N})^N \leq e^k$ and that $(\frac{N+k}{k})^k \leq (\frac{N+K-1}{K-1})^{K-1}$ we get the relaxed version

$$
R(N,K) \leq (K-1) \left[\frac{K-1}{K} \ln\left(\frac{N}{K-1} + 1 \right) + \frac{1}{2} \right].
$$