ON SUPERVISED LEARNING OF BAYESIAN NETWORK PARAMETERS

Hannes Wettig, Peter Grünwald, Teemu Roos
Petri Myllymäki and Henry Tirri

March 27, 2002

HIIT
TECHNICAL
REPORT
2002–1

ON SUPERVISED LEARNING OF BAYESIAN NETWORK PARAMETERS

Hannes Wettig, Peter Grünwald, Teemu Roos Petri Myllymäki and Henry Tirri

Helsinki Institute for Information Technology HIIT
Tammasaarenkatu 3, Helsinki, Finland
PO BOX 9800
FIN-02015 HUT, Finland
http://www.hiit.fi

HIIT Technical Reports 2002–1
ISSN 1458-9451

On Supervised Learning of Bayesian Network Parameters

Hannes Wettig[1] Peter Grünwald[2] Teemu Roos[1]
Petri Myllymäki[1] Henry Tirri[1]

[1] Complex Systems Computation Group (CoSCo),
Helsinki Institute for Information Technology (HIIT),
P.O. Box 9800, FIN-02015 HUT, Finland.
http://cosco.hiit.fi/, Firstname.Lastname@hiit.FI

[2] Centrum voor Wiskunde en Informatica (CWI),
P.O. Box 94079, NL-1098 SJ Amsterdam, The Netherlands.
http://www.cwi.nl/∼pdg/, pdg@cwi.nl

HIIT Technical Report 2002–1

March 27, 2002

Abstract

Bayesian network models are widely used for supervised prediction tasks such as classification. Usually the parameters of such models are determined using 'unsupervised' methods such as likelihood maximization, as it has not been clear how to find the parameters maximizing the supervised likelihood or posterior globally. In this paper we show how this supervised learning problem can be solved efficiently for a large class of Bayesian network models, including the Naive Bayes (NB) and Tree-augmented NB (TAN) classifiers. We show that there exists an alternative parameterization of these models in which the supervised likelihood becomes concave. From this result it follows that there can be at most one maximum, easily found by local optimization methods.

1 Introduction

In recent years it has been recognized that for supervised prediction tasks such as classification, we should use a supervised learning algorithm such as supervised (conditional) likelihood maximization (Greiner & Zhou, 2001; Ng & Jordan, 2001; Greiner *et al.* , 1997; Kontkanen *et al.* , 2001; Friedman *et al.* , 1997). Nevertheless, in most applications related to this type of task, model parameters are still determined using unsupervised methods such as ordinary likelihood maximization or (ordinary) Bayesian methods. One of the main reasons for this discrepancy is the difficulty in finding the global maximum of the supervised likelihood. In this technical report, we show that this problem can be solved for Bayesian network models, as long as they satisfy a particular additional condition. The condition is satisfied for many existing Bayesian-network based classifiers such as Naive Bayes (NB), TAN (Tree-augmented NB) and 'diagnostic' classifiers (Kontkanen *et al.* , 2001).

We find the maximum supervised likelihood by parameterizing our models in a non-standard manner; roughly speaking, the parameters in our parameterization correspond to logarithms of parameters in the standard Bayesian network parameterization. The new parameterization has the remarkable property that it makes the supervised likelihood a concave function of the parameters. We can therefore find the global maximum supervised likelihood parameters by simple local optimization techniques such as hill climbing. In the experimental part of the paper, we demonstrate the usefulness of our idea by applying it to infer supervised Naive Bayes distributions for a variety of real-world data sets. For most of our data sets, the supervised NB classifiers lead to (sometimes substantially) better predictions than those obtained by the ordinary, 'unsupervised' NB classifiers.

This paper is organized as follows. For ease of exposition, we use the Naive Bayes model as our running example, and first present all our main results in terms of it. We first in Section 2 review the standard (unsupervised) Naive Bayes classifier and its supervised version. Then we show that when this model is parameterized in the usual way, the supervised likelihood is not a concave function of the parameters, which hinders its optimization. In Section 3 we introduce the *L-model*. Although the L-model looks different from supervised NB, in Section 4 we show that the two models in fact represent exactly the same conditional distributions. In Section 5 we show that the supervised likelihood of the data, as a function of the parameters of the L-model, is concave, while the parameter set itself is convex. Section 6 provides alternative interpretations of the L-model. In Section 7 we generalize our results to more general classes of Bayesian network models. In Section 8 we argue that for technical reasons, it is useful to equip our models with such a prior that we effectively maximize the 'supervised Bayesian posterior' rather than the plain supervised likelihood. Finally, in Section 9, we compare our supervised NB to standard NB on a variety

of real-world data sets. An outlook on future research is given in Section 10.

2 The Supervised Naive Bayes Model

Let $(X_0, X_1, \ldots, X_M)$ be a discrete random vector, where each variable X_i takes on values $l \in \{1, \ldots, n_i\}$. The first variable X_0 is called the *class variable*, while the remaining $X_1, \ldots, X_M$ are the *predictor* variables or *attributes*. The (training) data set D consists of N vectors containing $M+1$ entries each: $D = (d_1, \ldots, d_N)$, with $d_j = (d_{j0}, \ldots, d_{jM})$. In the classification task, the goal is to build from the training data D a model that predicts the value of the class variable, given the values of the predictors.

The standard (multinomial) Naive Bayes classifier (NB) (see e.g. (Kontkanen *et al.* , 2000)) consists of parameters $\Theta^S = (\alpha^S, \Phi^S)$, where $\alpha^S = (\alpha^S_1, \ldots, \alpha^S_{n_0})$ and $\Phi^S = (\Phi^S_{kil})$, with $k \in \{1, \ldots, n_0\}$, $i \in \{1, \ldots, M\}$, and $l \in \{1, \ldots, n_i\}$. Here $\alpha^S = P(X_0 \mid \Theta^S)$ is the default distribution over the class, and each $\Phi^S_{ki} = P(X_i \mid X_0 = k, \Theta^S)$ is a distribution over the values of X_i given the class. We restrict our parameters to lie in the set $\boldsymbol{\Theta^S}$ defined as:

$$
\begin{aligned}
\boldsymbol{\alpha}^S &:= \{(\alpha^S_1, \ldots, \alpha^S_k) \mid \sum_{k=1}^{n_0} \alpha^S_k = 1;\ \text{all } \alpha_k > 0\} \\
\boldsymbol{\Phi^S} &:= \{\Phi^S \mid \forall_{\substack{k \in \{1,\ldots,n_0\} \\ i \in \{1,\ldots,M\}}} \sum_{l=1}^{n_i} \Phi^S_{kil} = 1;\ \text{all } \Phi_{kil} > 0\} \\
\boldsymbol{\Theta^S} &:= \{(\alpha^S, \Phi^S) \mid \alpha^S \in \boldsymbol{\alpha^S}; \Phi^S \in \boldsymbol{\Phi^S}\}.
\end{aligned}
$$

Note that $\overline{\boldsymbol{\Theta^S}}$, the *closure* of $\boldsymbol{\Theta^S}$, is the set of all parameter vectors that correspond to some Naive Bayes distribution. $\boldsymbol{\Theta^S}$ itself is the set of all parameter vectors corresponding to a Naive Bayes distribution with only strictly positive probabilities. As we shall see in section 8, without essential loss of generality we may restrict ourselves to parameters in $\boldsymbol{\Theta^S}$.

The (unsupervised) log-likelihood of D given Θ^S is defined as

$$
\begin{aligned}
\log P(D \mid \Theta^S) &= \sum_{j=1}^{N} \log P(d_j \mid \Theta^S), \\
\text{with}\ \ P(d_j \mid \Theta^S) &= \alpha^S_{d_{j0}} \prod_{i=1}^{M} \Phi^S_{d_{j0} i d_{ji}},
\end{aligned}
\tag{1}
$$

where the first equality refers to the *i.i.d.* (independent, identically distributed) assumption inherent to the Naive Bayes model. Eq. (1) can be rewritten as

$$
\log P(D|\Theta^S) = \sum_{k=1}^{n_0} \left(h_k \log \alpha^S_k + \sum_{i=1}^{M} \sum_{l=1}^{n_i} f_{kil} \log \Phi^S_{kil} \right), \tag{2}
$$

where h_k and f_{kil} are data frequency counters: h_k is the number of vectors d_j of class $d_{j0} = k$, and f_{kil} is the number of class k vectors with $d_{ji} = l$.

In the standard NB classifier, for given data D, one infers the maximum likelihood (ML) parameters $\hat{\Theta}^S$ maximizing (2). The inferred parameters $\hat{\Theta}^S$ can then be — and usually are — used for *supervised* prediction tasks: *given* $(X_1 = x_1, \ldots, X_m = x_M)$, one wants to make predictions about the value of X_0. This is done using the conditional distribution of X_0 given $x_1, \ldots, x_M$. For $\Theta^S \in \mathbf{\Theta^S}$, this distribution looks as follows:

$$P(X = k | X_1 = x_1, \ldots, X_M = x_M, \Theta^S) = \frac{\alpha^S_k \prod_{i=1}^M \Phi^S_{kix_i}}{\sum_{k'=1}^{n_0} \alpha^S_{k'} \prod_{i=1}^M \Phi^S_{k'ix_i}}. \tag{3}$$

It has often been argued that because the prediction task is supervised, the score function used to determine the parameters of a model should *also* be supervised, i.e. conditional (Friedman *et al.* , 1997; Greiner *et al.* , 1997; Greiner & Zhou, 2001; Kontkanen *et al.* , 2001; Ng & Jordan, 2001). This leads us to the supervised log-likelihood $S^S(d; \Theta^S)$ defined as follows. Let $d = (k, x_1, \ldots, x_M)$ be a single data vector. Then

$$S^S(d; \Theta^S) := \log P(k \mid x_1, \ldots, x_M, \Theta^S) = \log \frac{\alpha^S_k \prod_{i=1}^M \Phi^S_{kix_i}}{\sum_{k'=1}^{n_0} \alpha^S_{k'} \prod_{i=1}^M \Phi^S_{k'ix_i}}. \tag{4}$$

For a sample $D = (d_1, \ldots, d_N)$, this becomes

$$\begin{aligned} S^S(D; \Theta^S) &:= \sum_{j=1}^N S^S(d_j; \Theta^S) = \sum_{j=1}^N \log \frac{\alpha^S_{d_{j0}} \prod_{i=1}^M \Phi^S_{d_{j0}id_{ji}}}{\sum_{k'=1}^{n_0} \alpha^S_{k'} \prod_{i=1}^M \Phi^S_{k'id_{ji}}} \\ &= \sum_{k=1}^{n_0} \left(h_k \log \alpha^S_k + \sum_{i=1}^M \sum_{l=1}^{n_i} f_{kil} \log \Phi^S_{kil} \right) - \sum_{j=1}^N \log \left(\sum_{k'=1}^{n_0} \alpha^S_{k'} \prod_{i=1}^M \Phi^S_{k'id_{ji}} \right). \end{aligned} \tag{5}$$

In this paper, we are interested in the parameter vectors $\tilde{\alpha}^S$ and $\tilde{\Phi}^S$ maximizing the supervised log-likelihood (5). These are generally very different from the more commonly used ML parameters $\hat{\alpha}^S$ and $\hat{\Phi}^S$, arrived at by maximizing Eq. (2) analytically: while $\hat{\alpha}^S$ and $\hat{\Phi}^S$ are exactly proportional to their corresponding training data frequency vectors, the characterization of $\tilde{\alpha}^S$ and $\tilde{\Phi}^S$ is more complicated (see Section 6).

Since we are *only* interested in the conditional (supervised) likelihood, we will restrict our attention to the set of *conditional* distributions. Formally, we define the *Supervised Naive Bayes* model to be the set of *conditional* distributions of X_0 given $X_1, \ldots, X_M$, defined in Eq. (3):

$$\mathcal{M}^S := \{P(X_0 \mid X_1, \ldots, X_M, \Theta^S) \mid \Theta^S \in \mathbf{\Theta^S}\}.$$

The conditional distributions are extended to N outcomes by independence. For a sample D and parameters Θ^S, this results in the supervised likelihood $S^S(D; \Theta^S)$ given by (5).

Example 1 *($\mathbf{\Theta^S}$-parameterization is not 1-to-1).* Consider a domain with only two binary variables, $X_0 \in \{1,2\}$ and $X_1 \in \{1,2\}$. Let $\Phi^S_{111} = \Phi^S_{211} = b \in (0,1)$. For *all* values of b, the supervised score[1] of any vector (x_0, x_1) is given by

$$P(x_0 \mid x_1, (\alpha^S, \Phi^S)) = \frac{\alpha^S_{x_0} \Phi^S_{x_0 1 x_1}}{\sum_{k'} \alpha^S_{k'} \Phi^S_{k' 1 x_1}} = \alpha^S_{x_0},$$

which is constant wrt. b. This shows that there exist $\Theta^{(1)}, \Theta^{(2)} \in \mathbf{\Theta^S}$ with $\Theta^{(1)} \neq \Theta^{(2)}$, such that $P(\cdot|\Theta^{(1)}) = P(\cdot|\Theta^{(2)})$. While all $\Theta^S \in \mathbf{\Theta^S}$ index a different *unconditional* distribution, some of them index the same *conditional* distribution.

The problem with maximizing the supervised likelihood is that the conventional NB parameterization it is *not* concave. The following simple example shows that the supervised score $S^S(D; \Theta^S)$ may peak more than once along some line, contradicting concavity.

Example 2 *(Non-Concavity of the supervised score).* Consider the domain of the previous example. Let each of the four possible data vectors appear exactly once in the data set D. Set $\alpha^S := (0.1, 0.9)$ and $\Phi^S_{111} := \Phi^S_{112} := 0.5$. Figure 1 shows the plot of the supervised log-likelihood over $\Phi^S_{211} = 1 - \Phi^S_{212}$.

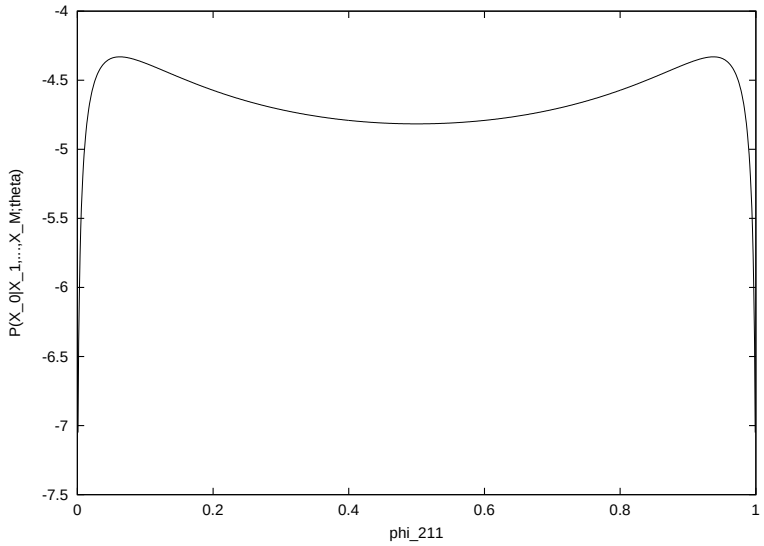

Figure 1: the supervised log-likelihood peaks twice as Φ^S_{211} varies.

Because of this non-concavity, we have to use complicated optimization methods to maximize the supervised score (in contrast to the unsupervised Naive Bayes

[1]We use the word 'score' whenever we want to stress that the log-likelihood is the objective we want to optimize.

case, we cannot solve the problem analytically). Such algorithms may converge slowly due to the non-concavity of the score. One may suspect that they could even get stuck in local maxima, but the tools we develop in the next section allow us to show later on (Proposition 2) that this cannot be so.

3 The Supervised L-Model

We now introduce the model $\mathcal{M}^L$. This is a set of conditional distributions, which, as we shall see, is just supervised NB in disguise, i.e., $\mathcal{M}^L = \mathcal{M}^S$.

Each distribution in $\mathcal{M}^L$ is defined in terms of a parameter vector $\Theta^L = (\alpha^L, \Phi^L)$, with $\alpha^L = (\alpha^L_k)_k$ and $\Phi^L = (\Phi^L_{kil})_{k,i,l}$ indexed as before. The set of all parameter vectors is denoted by $\mathbf{\Theta^L}$. We formally define this set by

$$\begin{aligned} \boldsymbol{\alpha^L} &:= \mathbf{R}^k, \quad \mathbf{\Phi^L} := \mathbf{R^{k\cdot(n_1+\ldots+n_M)}} \\ \text{and} \quad \mathbf{\Theta^L} &:= \{(\alpha^L, \Phi^L) \mid \alpha^L \in \boldsymbol{\alpha^L}; \Phi^L \in \mathbf{\Phi^L}\}. \end{aligned}$$

Each $(\alpha^L, \Phi^L) \in \mathbf{\Theta^L}$ indexes a conditional distribution $P(X_0 \mid X_1, \ldots, X_M, (\alpha^L, \Phi^L))$ as follows. For a data vector $d = (k, x_1, \ldots, x_M)$, let us define

$$\begin{aligned} &P(X_0 = k \mid X_1 = x_1, \ldots, X_M = x_M, (\alpha^L, \Phi^L)) \\ &\qquad := \frac{\exp(\alpha^L_k) \prod_{i=1}^M \exp(\Phi^L_{kix_i})}{\sum_{k'=1}^{n_0} \exp(\alpha^L_{k'}) \prod_{i=1}^M \exp(\Phi^L_{k'ix_i})}. \end{aligned} \tag{6}$$

The distributions $P(X_0 \mid X_1, \ldots, X_M, (\alpha^L, \Phi^L))$ are extended to several outcomes by independence (i.e. taking product distributions). One immediately verifies that, for all $x_1, \ldots, x_M$, $\sum_{k \in \{1,\ldots,n_0\}} P(k|x_1, \ldots, x_m, (\alpha^L, \Phi^L)) = 1$, and that each term in the sum is positive. This confirms that for all $(\alpha^L, \Phi^L) \in \mathbf{\Theta^L}$, and all $x_1, \ldots, x_M$, $P(X_0 \mid x_1, \ldots, x_M, (\alpha^L, \Phi^L))$ given by (6) indeed defines a conditional distribution over X_0.

The supervised log-likelihood corresponding to this conditional distribution is denoted by $S^L(d; \Theta^L)$. It is of course just the log of (6) and hence given by

$$S^L(d; (\alpha^L, \Phi^L)) = \alpha^L_k + \sum_{i=1}^M \Phi^L_{kix_i} - \log \sum_{k'=1}^{n_0} \exp(\alpha^L_{k'} + \sum_{i=1}^M \Phi^L_{k'ix_i}). \tag{7}$$

This is extended to sample $D = (d_1, \ldots, d_N)$ by independence:

$$S^L(D; (\alpha^L, \Phi^L)) = \sum_{j=1}^N S^L(d_j; (\alpha^L, \Phi^L)). \tag{8}$$

We now define the *supervised L-model* $\mathcal{M}_L$ as the set of conditional distributions that are indexed by $\mathbf{\Theta^L}$:

$$\mathcal{M}^L = \{P(X_0 \mid X_1, \ldots, X_M, \Theta^L) \mid \Theta^L \in \mathbf{\Theta^L}\} \tag{9}$$

As for the model $\mathcal{M}^S$ with parameters Θ^S, the mapping from parameters Θ^L to models in $\mathcal{M}^L$ is not one-to-one:

Proposition 1 *Let* $(\alpha^L, \Phi^L) \in \mathbf{\Theta^L}$. *Let* $(\gamma_1, \ldots, \gamma_{n_0})$ *be any vector in* $\mathbf{R}^k$ *and set* $\Psi_{kil} := -M^{-1}\gamma_k$ *for all* k, i, l. *Then* $(\alpha^L + \gamma, \Phi^L + \Psi) \in \mathbf{\Theta^L}$, *and both* (α^L, Φ^L) *and* $(\alpha^L + \gamma, \Phi^L + \Psi)$ *index the same conditional distribution in* $\mathcal{M}^L$.

Proof: Plug $(\alpha^L + \gamma, \Phi^L + \Psi)$ into (7).

□

We now have two supervised (conditional) models: $\mathcal{M}^S$ indexed by $\mathbf{\Theta^S}$, corresponding to the conditional NB distributions; and $\mathcal{M}^L$ indexed by $\mathbf{\Theta^L}$, corresponding to the conditional 'L-distributions'. In the next section we show that these two seemingly different conditional models are in fact equal.

4 The sets $\mathcal{M}^S$ and $\mathcal{M}^L$ are equivalent

To see that $\mathcal{M}^S$ and $\mathcal{M}^L$ are related, define the *log-transformation* $L : \mathbf{\Theta^S} \rightarrow \mathbf{\Theta^L}$ as follows. For a given parameter vector $(\alpha^S, \Phi^S) \in \mathbf{\Theta^S}$, the corresponding transformed parameters $L((\alpha^S, \Phi^S))$ are defined as $L((\alpha^S, \Phi^S)) := (\alpha^L, \Phi^L)$ with (α^L, Φ^L) defined as:

$$\alpha_k^L := \log \alpha_k^S \quad ; \quad \Phi_{kil}^L := \log \Phi_{kil}^S \tag{10}$$

By plugging in (10) into (8) and further into (7), we see that for all $\Theta^S \in \mathbf{\Theta^S}$, $P(X_0 \mid X_1, \ldots, X_m, \Theta^S) = P(X_0 \mid X_1, \ldots, X_m, L(\Theta^S))$. This shows that $\mathcal{M}^S \subseteq \mathcal{M}^L$: each parameter Θ^S indexing a distribution in $\mathcal{M}^S$ is transformed into a parameter Θ^L indexing the *same* conditional distribution in $\mathcal{M}^L$. By this result, one may be tempted to view $\mathbf{\Theta^L}$ simply as a parameterization of $\mathcal{M}^S$ in terms of the logarithms of the original parameters. But it is more complicated than that: in $\mathbf{\Theta^L}$ *all* parameters α_k^L and ϕ_{kil}^L are allowed, not just those that, when exponentiated, can be interpreted as probabilities (i.e. sum to 1 over k and l respectively). Nevertheless we have:

Theorem 1 $\mathcal{M}^S = \mathcal{M}^L$.

Proof: We have already shown that $\mathcal{M}^S \subseteq \mathcal{M}^L$. To show that also $\mathcal{M}^L \subseteq \mathcal{M}^S$, let $(\alpha^L, \Phi^L) \in \mathbf{\Theta^L}$. Let $c \in \mathbf{R}^{1+Mn_0}$ be a vector with components $(c_0, (c_{11}, \ldots, c_{1M}), \ldots, (c_{n_0 1}, \ldots, c_{n_0 M}))$. Define, for $k \in \{1, \ldots, n_0\}$,

$$\Phi_{kil}^{(c)} := \Phi_{kil}^L + c_{ki}, \;\; \alpha_k^{(c)} := \alpha_k^L + c_0 - \sum_{i=1}^{M} c_{ki}. \tag{11}$$

From (7) we infer that, for all $c \in \mathbf{R}^{1+Mn_0}$ and all d,

$$S^L(d; (\alpha^{(c)}, \Phi^{(c)})) = S^L(d; (\alpha^L, \Phi^L)). \tag{12}$$

To see that (12) holds, just substitute its left-hand side into (7) and see that all c_0 and c_{ki} cancel. Now define

$$\begin{aligned} \Phi^S_{kil} &:= \exp(\Phi^{(c)}_{kil}) = \exp(\Phi^L_{kil} + c_{ki}), \\ \alpha^S_k &:= \exp(\alpha^{(c)}_k) = \exp(\alpha^L_k + c_0 - \sum_{i=1}^{M} c_{ki}). \end{aligned} \tag{13}$$

Evidently, for all k and i we can choose c_{ki} such that $\sum_{l=1}^{n_i} \Phi^S_{kil} = 1$, and subsequently c_0 such that $\sum_k \alpha^S_k = 1$. This implies that $(\alpha^S, \Phi^S) \in \mathbf{\Theta^S}$. Substituting (13) into (4), we find that, for all d,

$$S^S(d; (\alpha^S, \Phi^S)) = S^L(d; (\alpha^{(c)}, \Phi^{(c)})).$$

Equation 12 now implies that $\mathcal{M}^S \subseteq \mathcal{M}^L$.

□

Because of the equality proved above, we can think of $\mathbf{\Theta^L}$ as a parameterization of the supervised Naive Bayes model $\mathcal{M}^S$; we call $\mathbf{\Theta^L}$ the *L-parameterization* of $\mathcal{M}^S$.

5 Concavity

We saw that the supervised log-likelihood is not concave for standard supervised NB. Our main theorem shows that, remarkably, it *becomes* concave in the L-parameterization:

Theorem 2 *Let* $\Theta^{(1)}, \Theta^{(2)}, \Theta^L \in \mathbf{\Theta^L}$. *Then:*

(i) For any $\lambda \in [0, 1]$, $\lambda\Theta^{(1)} + (1 - \lambda)\Theta^{(2)} \in \mathbf{\Theta^L}$ *(hence* $\mathbf{\Theta^L}$ *is a convex set).*

(ii) For any sample D *of any length,* $S^L(D; \Theta^L)$ *is a concave (but not strictly concave!) function of* Θ^L.

Proof: Item (i) is immediate. For item (ii) we first introduce some convenient notation. Given a data vector $d = (x_0, \ldots, x_M)$ and parameters (α^L, Φ^L) and $k \in \{1, \ldots, n_0\}$, we write $\beta_{k0}(d)$ for α^L_k and $\beta_{ki}(d)$ for $\Phi^L_{kix_i}$. Whenever d is clear from the context, we omit (d) from $\beta_{ki}(d)$ and simply write β_{ki}. With this notation, the supervised log-likelihood $S^L(d; \Theta^L)$ can be written as

$$\begin{aligned} S^L(d; \Theta^L) &= \sum_{i=0}^{M} \beta_{x_0 i} + g(d; \Theta^L), \\ \text{where} \quad g(d; \Theta^L) &= -\log \sum_{k=1}^{n_0} \exp \sum_{i=0}^{M} \beta_{ki}. \end{aligned} \tag{14}$$

We first show that $S^L(D;\Theta^L)$ is concave as a function of Θ^L. By (8), it suffices to show for any d that $S^L(d;\Theta^L)$ is concave as a function of Θ^L. Thus, we need to show for all $\Theta^{(1)},\Theta^{(2)} \in \mathbf{\Theta^L}$ and all $\lambda \in [0,1]$, that

$$S^L(d;\lambda\Theta^{(1)} + (1-\lambda)\Theta^{(2)}) \geq \lambda S^L(d;\Theta^{(1)}) + (1-\lambda)S^L(d;\Theta^{(2)}). \tag{15}$$

The left-hand side of (15) can be rewritten as

$$\begin{aligned} &S^L(d;\lambda\Theta^{(1)} + (1-\lambda)\Theta^{(2)}) \\ &\qquad = \sum_{i=0}^{M}(\lambda\beta^{(1)}_{x_0 i} + (1-\lambda)\beta^{(2)}_{x_0 i}) + g(d;\lambda\Theta^{(1)} + (1-\lambda)\Theta^{(2)}) \end{aligned} \tag{16}$$

with $g(d;\cdot)$ as in (14). The right-hand side in turn becomes

$$\begin{aligned} &\lambda S^L(d;\Theta^{(1)}) + (1-\lambda)S^L(d;\Theta^{(2)}) \\ &\quad = \lambda\sum_{i=0}^{M}\beta^{(1)}_{x_0 i} + (1-\lambda)\sum_{i=0}^{M}\beta^{(2)}_{x_0 i} + \lambda g(d;\Theta^{(1)}) + (1-\lambda)g(d;\Theta^{(2)}) \\ &\qquad = \sum_{i=0}^{M}(\lambda\beta^{(1)}_{x_0 i} + (1-\lambda)\beta^{(2)}_{x_0 i}) + \lambda g(d;\Theta^{(1)}) + (1-\lambda)g(d;\Theta^{(2)}). \end{aligned} \tag{17}$$

Comparing (16) and (17), we see that their leftmost terms coincide. Substituting these equations into (15), these terms cancel and we see that $S^L(d;\Theta^L)$ is concave if and only if $g(d;\Theta^L)$ is concave.

Hence we need to show that $g(d;\Theta^L)$ is concave over $\mathbf{\Theta^L}$. First note that (a) $g(d;\Theta^L)$ is continuous in Θ^L at all $\Theta^L \in \mathbf{\Theta^L}$; and (b) $\mathbf{\Theta^L}$ is a convex set (item (i) of the theorem). Thus it suffices to prove the following claim for all $\Theta^{(1)},\Theta^{(2)}$:

$$2g\left(d;\frac{\Theta^{(1)}+\Theta^{(2)}}{2}\right) - g(d;\Theta^{(1)}) - g(d;\Theta^{(2)}) \geq 0. \tag{18}$$

Let $b^{(j)}_k = \sum_{i=0}^{M}\beta^{(j)}_{ki}$. The following chain of (in-) equalities shows that (18) indeed holds:

$$\begin{aligned} &2g\left(d;\frac{\Theta^{(1)}+\Theta^{(2)}}{2}\right) - g(d;\Theta^{(1)}) - g(d;\Theta^{(2)}) \\ &= -\log\left(\sum_k \exp\frac{b^{(1)}_k + b^{(2)}_k}{2}\right)^2 + \log\left(\left(\sum_k \exp b^{(1)}_k\right)\left(\sum_k \exp b^{(2)}_k\right)\right) \\ &= -\log\left(\sum_k \exp(b^{(1)}_k + b^{(2)}_k) + 2\sum_{k>k'}\exp\frac{b^{(1)}_k + b^{(2)}_{k'} + b^{(1)}_{k'} + b^{(2)}_k}{2}\right) \\ &\quad + \log\left(\sum_k \exp(b^{(1)}_k + b^{(2)}_k) + \sum_{k>k'}(\exp(b^{(1)}_k + b^{(2)}_{k'}) + \exp(b^{(1)}_{k'} + b^{(2)}_k))\right) \geq 0. \end{aligned}$$

The final inequality holds because

$$\forall_{x,y\in\mathcal{R}} \quad \exp(x)+\exp(y) \geq 2\exp\left(\frac{x+y}{2}\right),$$

which implies what we have used here, namely

$$\log(\exp(x)+\exp(y)+C) \geq \log(2\exp\left(\frac{x+y}{2}\right)+C)$$

for $C > 0$. This shows that $S^L(d;\Theta^L)$ is concave. To see that it is not strictly concave, let $(\alpha^L,\Phi^L) \in \mathbf{\Theta^L}$, and let γ and Ψ as in Proposition 1. For $\lambda \in [0,1]$ define $\Theta_\lambda := \lambda(\alpha^L,\Phi^L) + (1-\lambda)(\alpha^L+\gamma,\Phi^L+\Psi)$. Then clearly $S^L(D;\Theta_\lambda)$ is constant wrt. λ. □

Together, items (i) and (ii) demonstrate that finding the Naive Bayes distribution maximizing the supervised likelihood in the L-parameterization is finding the maximum of a concave function over a convex set. Thus we can use a simple local optimization method such as hill-climbing. The only remaining difficulty is that because concavity is not strict, there will be flat areas in the supervised likelihood surface. In Section 8 we discuss how to handle these.

Here is an important consequence of Theorem 2:

Proposition 2 *The log-likelihood does not have local maxima over the standard parameterization* $\mathbf{\Theta^S}$.

Proof sketch: It is easily shown that the L-transform and the 'S-transform' (Eqs. 12, 13) are continuous. Also, all parameters in $\mathbf{\Theta^S}$ corresponding to the same distribution in $\mathcal{M}^S$ form a connected set; and $S^L(D;\Theta^L)$ is concave. We can exploit these facts to drive the assumption of multiple local maxima in $\mathbf{\Theta^S}$ to contradiction.

We can now make the following two remarks. First, the global maximum will be achieved for a connected set of points rather than a single point. Second, although the log-likelihood can have no local maxima for the standard Naive Bayes parameterization, it is not concave either (i.e. it will have ripples and wrinkles). Greiner and Zhou have used the L-parameterization in (Greiner & Zhou, 2001) and report that "it worked better" [than the standard parameterization]. Our results explain this.

Example 3 *(The concavified surface).* Let us once more look at the domain consisting of only two binary variables, but this time we choose the L-model. Again we set $\alpha^L := (0.1, 0.9)$. Figure 2 gives some clue of how it is possible to concavify the objective, and why it could peak twice in Example 2.

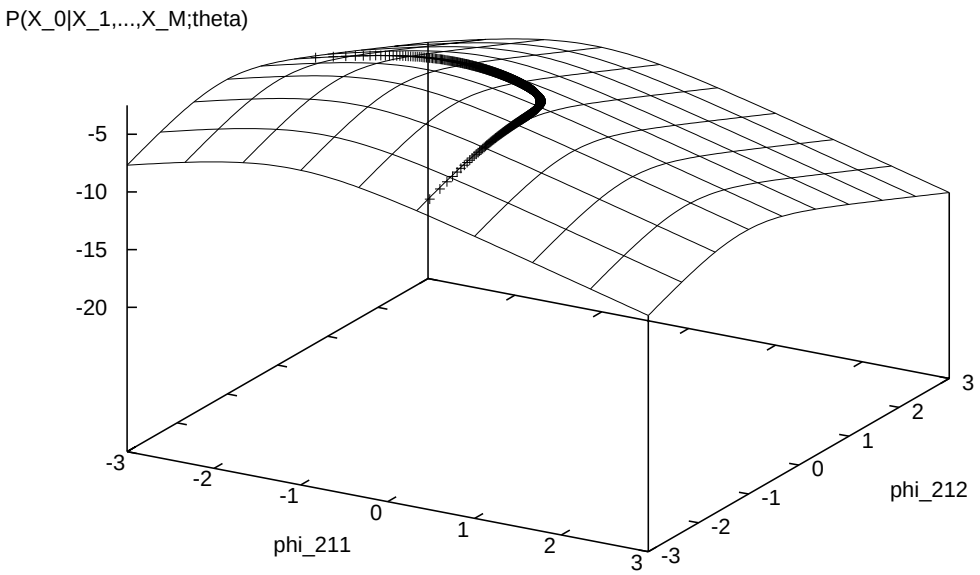

Figure 2: the supervised log-likelihood has become a concave function of Φ^L_{211} and Φ^L_{212}. The pointed line shows the transform of Φ^S_{211} from Figure 1.

6 Alternative Views of the L-Model

The L-parameterization allows us to think of the Naive Bayes classifier as a discriminative (diagnostic) rather than as a generative (sampling) model, see e.g. (Dawid, 1976; Ng & Jordan, 2001). Even though formally identical to supervised Naive Bayes, the L-model can also be interpreted in terms of logistic regression, neural networks and 'recalibrated' models.

Discrete, Supervised Logistic Regression. We can think of the conditional model $\mathcal{M}^L$ as a predictor that combines the information of the attributes using softmax. This is usually done for the continuous or binary case ('linear softmax'; (Heckerman & Meek, 1997; Ng & Jordan, 2001)). Figure 3 gives an interpretation of this, depicting both Naive Bayes and the L-model in their Bayesian network guises. The L-model $\mathcal{M}^L$ does not contain any notion of the unsupervised probabilities. Terms such as $P(X_i|\Theta^L)$ are undefined, and neither are we interested in them, our task is prediction of X_0 *given the* X_i. In this sense, the L-model is *not* a BRC-model in the sense of Heckerman and Meek (Heckerman & Meek, 1997), and we do not have to concern ourselves with *variational dependence*.

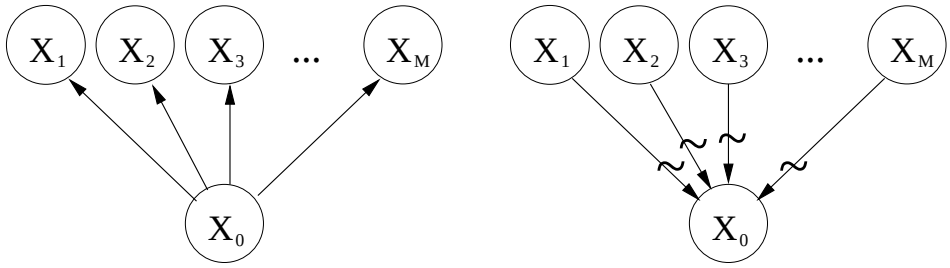

Figure 3: standard Naive Bayes net (left) and L-model (right). The arcs of the network have been reversed and the resulting product distribution has been replaced by softmax (denoted by tildes).

Neural Networks. The conditional distribution (6) is equivalent also to a single-layer (no hidden units) linear feed-forward neural network with logistic sigmoid (softmax) activation function, see e.g. (Bishop, 1995). In this type of a network both inputs and outputs are encoded using the so called 1-of-c encoding with a binary node for each variable–value combination. Thus the logistic activation function is applied to a linear function of the resulting set of indicator variables and the activation value of the output nodes can be interpreted as probabilities of the corresponding class values.

The α_k^L terms which represent the default classification of the $\mathcal{M}^L$ model can be implemented by adding a so called *bias* node, i.e. a node with constant input, to the network. It is sometimes recommended that if a bias node is present one should use a 1-of-c-1 encoding instead of the 1-of-c encoding because the 1-of-c encoding creates a linear dependency on the bias unit (Sarle, 2001). In other words the model is overparametrized. Indeed the same phenomenon is present in our model which is indicated by Proposition 1. In Section 8 we present a solution to the optimization difficulties caused by overparametrization. Our solution which is justified by priors defined over the parameter space is in effect similar to the *weight-decay* method used in neural networks literature.

The parameters of the neural network are usually optimized to maximize the conditional likelihood, or equivalently the so called *cross-entropy*, by local search heuristics such as the gradient descent algorithm. Because of the equivalence of our L-parametrization and single-layer feed-forward neural networks it follows from Theorem 2 that the objective function of the neural network is also concave. However, it does not follow that concavity would be preserved when hidden layers are added to the network.

Calibration. The L-model has the following interesting property: the derivative of $S^L(D;\Theta^L)$ becomes zero if and only if for all k, i, l, the following holds:

$$\begin{aligned} \sum_{j=1}^{N} P(X_0 = k \mid d_{j1}, \ldots, d_{jM}, \Theta^L) &= h_k, \\ \text{and} \sum_{j:d_{ji}=l} P(X_0 = k \mid d_{j1}, \ldots, d_{jM}, \Theta^L) &= f_{kil}. \end{aligned} \tag{19}$$

That is, we have found good parameters for the supervised task exactly when we are 'well-calibrated' wrt. D and all subsets $D_{il} := \{d_j \mid d_{ji} = l\}$ in the sense of (Dawid, 1982). Thus optimizing Θ^L according to S^L means 'recalibrating' ourselves using $\sum_{i=1}^{M} n_i + 1$ calibration tests simultaneously. Here the independence assumption of our model saves us from becoming 'incoherent' as we recalibrate, see (Dawid, 1982).

As a spin-off from this research, we find that we can solve *any* calibration problem of the form

$$\forall_{f \in \mathcal{F}} \sum_{j:f(d_j)=1} P(X_0 = k \mid d_{j1}, \ldots, d_{jM}, \Theta^L) = |\{j : f(d_j) = 1 \wedge d_{j0} = k\}|,$$

where $\mathcal{F}$ is any collection of indicator functions computable from $X_1, \ldots, X_M$ by local optimization methods. In the long run—with an unlimited amount of data available—we should be calibrated with respect to *all* such calibration tests f, see (Turdaliev, 1999). With only limited data availability the calibration tests implicit to the Naive Bayes model (i.e. $\mathcal{F}_{NB} = \{f_{il} : f_{il}(d) = 1 \Leftrightarrow d_i = l\} \cap \{1\}$) seem to be a sensible choice in many cases. Other choices can be made that do not necessarily correspond to a Bayesian model. In order to avoid over-fitting we may, for instance, prune the NB model by demanding the calibration sets to be of certain minimal size c, arriving at $\mathcal{F}_c = \{f_{il} : |D_{il}| \geq c\} \cap \{1\}$. For small data sets the resulting model may consist of considerably fewer parameters (depending on c).

7 More General Bayes Nets

Our main ideas were most easily explained using the Naive Bayes classifier as a running example. But in fact they apply to all Bayesian Network models as long as they satisfy an extra condition as given below. We shall now introduce some more notation needed to describe this generalization thereafter.

Consider a set of random variables $X_0, X_1, \ldots, X_{M'}$ taking values in $\{1, \ldots, n_0\}, \ldots, \{1, \ldots, n_{M'}\}$ respectively. Let $\mathcal{B}$ be a Bayesian network structure over $X_0, \ldots, X_{M'}$, which factorizes $P(X)$ into

$$P(X_0, \ldots, X_{M'}) = \prod_{i=0}^{M'} P(X_i \mid Pa_i),$$

where Pa_i is the parent set of variable X_i in $\mathcal{B}$.

We are interested in predicting some class variable X_m for some $m \in \{0, \ldots, M'\}$ conditioned on all X_i, $i \neq m$. Without loss of generality we may assume that $m = 0$ (i.e. X_0 is the class variable) and that the children of X_0 in $\mathcal{B}$ are $\{X_1, \ldots, X_M\}$ for some $M \leq M'$. For example, if we take $M = M'$ and $\mathcal{B}$ the Naive Bayes structure (leftmost picture in Figure 3), then we are back at our original case. The Bayesian Network model corresponding to $\mathcal{B}$ is the set of all distributions satisfying the conditional independencies encoded in $\mathcal{B}$. It is usually parameterized by vectors $\Theta^{\mathcal{B}}$ with components of the form $\theta^{\mathcal{B}}_{(i,x_i)|q_i}$ defined by

$$\theta^{\mathcal{B}}_{(i,x_i)|q_i} := P(X_i = x_i | Pa_i = q_i),$$

where q_i is any configuration (set of values) for the parents Pa_i of X_i. We let $\mathcal{M}^{\mathcal{B}}$ be the set of *conditional* distributions $P(X_0|X_1, \ldots, X_{M'}, \Theta^{\mathcal{B}})$ corresponding to a distribution $P((X_0, \ldots, X_{M'}) \mid \Theta^{\mathcal{B}})$ satisfying the conditional independencies encoded in $\mathcal{B}$.

We now write $q_i(x)$ to denote the configuration of Pa_i in $\mathcal{B}$ given by the vector $x = (x_0, \ldots, x_{M'})$, and $q_i(k, x)$ for the same configuration given by $(k, x_1, \ldots, x_{M'})$. Then $\mathcal{M}^{\mathcal{B}}$ contains the conditional distributions

$$P(X_0 \mid x_1, \ldots, x_{M'}, \Theta^{\mathcal{B}}) = \frac{\theta^{\mathcal{B}}_{(0,x_0)|q_0(x)} \prod_{i=1}^{M'} \theta^{\mathcal{B}}_{(i,x_i)|q_i(x)}}{\sum_{k'=1}^{n_0} \theta^{\mathcal{B}}_{(0,k')|q_0(x)} \prod_{i=1}^{M'} \theta^{\mathcal{B}}_{(i,x_i)|q_i(k',x)}}, \tag{20}$$

extended to N outcomes by independence. In particular, all $\theta_{(i,x_i)|q_i}$ with $i > M$ (standing for nodes that are neither the class variable nor any of its children) cancel out of the equation, since for these terms it is $q_i(x) = q_i(k, x)$. Thus the only relevant parameters for determining the conditional likelihood are of the form $\theta^{\mathcal{B}}_{(i,x_i)|q_i}$ for all $i \in \{0..M\}$, $x_i \in \{1..n_i\}$ and q_i any configuration of values of Pa_i. We order these parameters lexicographically and define $\mathbf{\Theta}^{\mathcal{B}}$ to be the set of vectors constructed this way, with, for all $i \in \{0, \ldots, M\}$, x_i and q_i, $\theta^{\mathcal{B}}_{(i,x_i)|q_i} > 0$ and $\sum_{x_i=1}^{n_i} \theta^{\mathcal{B}}_{(i,x_i)|q_i} = 1$. $\mathbf{\Theta}^{\mathcal{B}}$ is a generalization of $\mathbf{\Theta}^{\mathbf{S}}$ to arbitrary Bayesian network models.

We now re-define $\mathbf{\Theta}^{\mathbf{L}}$ analogously to its previous definition: for each component $\theta^{\mathcal{B}}_{(i,x_i)|q_i}$ of each vector $\Theta^{\mathcal{B}} \in \mathbf{\Theta}^{\mathcal{B}}$, there is a corresponding component $\theta^{L}_{(i,x_i)|q_i}$ of the vectors $\Theta^{L} \in \mathbf{\Theta}^{\mathbf{L}}$; but the components $\theta^{L}_{(i,x_i)|q_i}$ are in the range $(-\infty, \infty)$ rather than $(0, 1)$. Each $\Theta^{L} \in \mathbf{\Theta}^{\mathbf{L}}$ defines the following conditional distribution:

$$P(X_0 \mid x_1, \ldots, x_{M'}, \Theta^{L}) := \frac{\exp(\theta^{L}_{(0,x_0)|q_0(x)}) \prod_{i=1}^{M} \exp(\theta^{L}_{(i,x_i)|q_i(x)})}{\sum_{k'=1}^{n_0} \exp(\theta^{L}_{(0,k')|q_0(x)}) \prod_{i=1}^{M} \exp(\theta^{L}_{(i,x_i)|q_i(k',x)})}. \tag{21}$$

This gives supervised likelihood $S^{L}(D; \Theta^{L}) = \sum_{j=1}^{N} S^{L}(d_j; \Theta^{L})$ with $S^{L}(d; \Theta^{L})$ equal to the logarithm of (21).

We define $\mathcal{M}^L$ to be the set of *conditional* distributions $P(X_0|X_1,\ldots,X_{M'},\Theta^L)$ for $\Theta^L \in \mathbf{\Theta^L}$. These distributions are extended to N outcomes by independence. We will see below that we can show analogs of Theorems 1 and 2 (and hence optimize the supervised likelihood by hill-climbing) as long as the following condition holds for $\mathcal{B}$:

Condition 1 For all $j = 1..M$, there exists $X_i \in Pa_j \cap \{X_0,\ldots,X_M\}$ such that $Pa_j \subseteq Pa_i \cup \{X_i\}$.

Remark. Condition 1 demands that any two parents of any child of the class X_0 are either connected via an arc in $\mathcal{B}$, or they must both be parents of X_0. In particular, any node having a common child together with X_0 must also be connected to X_0 itself. In other words, all parents Pa_j of a child X_j of X_0 must be 'conditionally fully connected' in $\mathcal{B}$, i.e. fully connected modulo arcs (between parents of X_0) that have no effect on the conditional $P(X_0 \mid Pa_j \setminus \{X_0\})$.

Condition 1 is automatically satisfied by the Naive Bayes (NB) and (as can easily be verified) the TAN (tree-augmented NB) classifiers Friedman *et al.* (1997). It is also automatically satisfied if X_0 only has incoming arcs[2] ('diagnostic' classifiers, see (Kontkanen *et al.* , 2001). For Bayesian network structures for which the condition does not hold, we can always add some arrows to arrive at a structure $\mathcal{B}'$ for which the condition does hold. Therefore, the model $\mathcal{M}^{\mathcal{B}}$ is always a submodel of a larger model $\mathcal{M}^{\mathcal{B}'}$ for which the condition holds. For these reasons, we regard Condition 1 as relatively mild. It allows us to generalize Theorems 1 and 2 as follows:

Theorem 3 *$\mathcal{M}^{\mathcal{B}} \subseteq \mathcal{M}^L$. Moreover, if $\mathcal{B}$ satisfies Condition 1, then $\mathcal{M}^{\mathcal{B}} = \mathcal{M}^L$.*

Theorem 4 *Θ^L (as defined in this section) is convex. $S^L(D;\Theta^L)$ is concave, though not strictly concave.*

The proof of Theorem 4 is entirely analogous to the proof of Theorem 2 and therefore omitted.

Proof of Theorem 3 $\mathcal{M}^{\mathcal{B}} \subseteq \mathcal{M}^L$ is immediate from doing the log-parameter transformation, i.e. setting $\theta^L_{(i,x_i)|q_i} := \log \theta^{\mathcal{B}}_{(i,x_i)|q_i}$ for all i, x_i and q_i.

It remains to show the hard part: under Condition 1, $\mathcal{M}^L \subseteq \mathcal{M}^{\mathcal{B}}$. In the following, we will often speak of the parent configuration q_0 of X_0. In case X_0 has no parents (i.e. $M = M'$), Pa_0 is the empty set and $q_0(x)$ is independent of the values of $x = (x_0,\ldots,x_{M'})$.

[2]It is easy to see that in that case the maximum supervised likelihood may even be determined analytically.

We introduce some more notation. For $j = 1..M$, let p_j be the maximum number in $\{0, \ldots, M\}$ such that $X_{p_j} \in Pa_j$, $Pa_j \subseteq Pa_{p_j} \cup \{X_{p_j}\}$. Such a p_j exists by Condition 1. Let $i = p_j$. Condition 1 implies that $q_j(x)$ is completely determined by the pair $(x_i, q_i(x))$. We can therefore introduce functions Q_j mapping $(x_i, q_i(x))$ to the corresponding $q_j(x)$. We then get that, for every instantiation $x = (x_0, \ldots, x_{M'})$ of all the variables and corresponding parent configurations $q_0(x), \ldots, q_M(x)$, for $j = 1..M$,

$$q_j(x) = Q_j(x_{p_j}, q_{p_j}(x)). \tag{22}$$

Now, for $i = 0..M$ and for each configuration q_i of Pa_i, we introduce a constant $c_{i|q_i}$ and we define, for any $\Theta^L \in \mathbf{\Theta^L}$,

$$\theta^{(c)}_{(i,x_i)|q_i} := \theta^L_{(i,x_i)|q_i} + c_{i|q_i} - \sum_{j:p_j=i} c_{j|Q_j(x_i,q_i)}. \tag{23}$$

The $\theta^{(c)}_{(i,x_i)|q_i}$ constructed this way are combined to a vector $\Theta^{(c)}$ which clearly is a member of $\mathbf{\Theta^L}$.

Stage 1 In this stage of the proof, we show that no matter how we choose the constants $c_{i|q_i}$, for all Θ^L and corresponding $\Theta^{(c)}$ we have $S^L(D; \Theta^{(c)}) = S^L(D; \Theta^L)$.

To see this, consider any data vector $d = (x_0, \ldots, x_{M'})$. d determines configurations $q_0(d), \ldots, q_M(d)$ of the parents of $X_0, \ldots, X_M$. We first show that, for every possible d, no matter how the choose the $c_{i|q_i}$,

$$\sum_{i=0}^{M} \theta^{(c)}_{(i,x_i)|q_i(d)} = \sum_{i=0}^{M} \theta^L_{(i,x_i)|q_i(d)} + c_{0|q_0(d)}. \tag{24}$$

To derive (24) we substitute all terms of $\sum_{i=0}^{M} \theta^{(c)}_{(i,x_i)|q_i(d)}$ by their definition (23). Clearly, for $j = 1..M$, there is exactly one term of the form $c_{j|q_j(d)}$ that appears in the sum with positive sign. Since for each $j \in \{1, \ldots, M\}$ there exists exactly one $i \in \{0, \ldots, M\}$ with $p_j = i$, it must be the case that for $j = 1..M$, a term of the form $c_{j|Q_j(x_i,q_i(d))}$ appears exactly once in the sum with negative sign. By (22) we have $c_{j|Q_j(x_i,q_i(d))} = c_{j|q_j(d)}$. Therefore all terms $c_{j|q_j(d)}$ that appear once with positive sign also appear once with negative sign. It follows that, except for $c_{0|q_0(d)}$, all terms $c_{j|q_j(d)}$ cancel. This establishes (24). By plugging in (24) into Equation 21, it now easily follows that $S^L(D; \Theta^{(c)}) = S^L(D; \Theta^L)$ for any D of any length. This concludes the proof of Stage 1.

Stage 2 Define

$$\theta^{\mathcal{B}}_{(i,x_i)|q_i} = \exp(\theta^{(c)}_{(i,x_i)|q_i}). \tag{25}$$

In this stage we show that we can determine the $c_{i|q_i}$ such that for $i = 0..M$, all x_i and q_i,

$$\sum_{x_i=1}^{n_i} \theta^{\mathcal{B}}_{(i,x_i)|q_i} = 1. \tag{26}$$

We will achieve this by sequentially determining values for $c_{i|q_i}$ in a particular order. We now need some terminology: we say 'c_i *is determined*' if for all configurations q_i of Pa_i, we have already determined $c_{i|q_i}$. We say 'c_i *is undetermined*' if we have determined $c_{i|q_i}$ for *no* configuration q_i of Pa_i. We say 'c_i is *ready to be determined*' if c_i is undetermined and at the same time all c_j with $p_j = i$ have been determined.

We first note that as long as some c_i are undetermined for $i = 0..M$, there must exist $c_{i'}$ that are ready to be determined. To see this, note either c_i itself is ready to be determined (in which case we are done), or there exists $j \in \{1, \ldots M\}$ with $p_j = i$ (and hence $X_i \in Pa_j$) such that c_j is undetermined. If c_j is ready to be determined, we are done. Otherwise, there must exist some k with $X_j \in Pa_k$ such that c_k is undetermined. We can now repeat the argument, and move forward in the Bayesian network structure $\mathcal{B}$ restricted to $\{X_0, \ldots, X_M\}$ until we find a c_l that is ready to be determined. Because $\mathcal{B}$ is acyclic, we must find such a c_l within $M + 1$ steps.

We now describe an algorithm that sequentially assigns values to c_i such that (26) will be satisfied. We start with all c_i undetermined.

WHILE there exists $i \in \{0, \ldots, M\}$ such that c_i is undetermined DO:
{

i. Pick any i such that c_i is ready to be determined (we have just seen that this is possible).

ii. Set, for all configurations q_i of Pa_i, $c_{i|q_i}$ such that $\sum_{x_i=1}^{n_i} \theta^{\mathcal{B}}_{(i,x_i)|q_i} = 1$ holds (clearly this is possible).

}

This algorithm will loop $M + 1$ times and then halt. Step 2 does not affect the values of $c_{j|q_j}$ for any j, q_j such that $c_{j|q_j}$ has already been determined. Therefore, after the algorithm halts, (26) holds. This concludes the proof of Stage 2.

Let $\Theta^L \in \mathbf{\Theta^L}$. For each choice of constants $c_{i|q_i}$ this determines a corresponding vector $\Theta^{(c)}$ with components given by (23). This in turn determines a corresponding vector $\Theta^{\mathcal{B}}$ with components given by (25). In Stage 2 we showed that we can take the $c_{i|q_i}$ such that (26) holds. This is the choice of $c_{i|q_i}$ which we adopt. With this particular choice, $\Theta^{\mathcal{B}}$ indexes a distribution in $\mathcal{M}^{\mathcal{B}}$. By applying the log-transformation to the components of $\Theta^{\mathcal{B}}$ we find that for any D of any length,

$S^{\mathcal{B}}(D;\Theta^{\mathcal{B}}) = S^L(D;\Theta^{(c)})$, where $S^{\mathcal{B}}(D;\Theta^{\mathcal{B}})$ denotes the supervised likelihood of $\Theta^{\mathcal{B}}$ as given by summing the logarithm of (20). The result of Stage 1 now implies that $\Theta^{\mathcal{B}}$ indexes the same conditional distribution as Θ^L. Since $\Theta^L \in \mathbf{\Theta^L}$ was chosen arbitrarily, this shows that $\mathcal{M}^L \subseteq \mathcal{M}^{\mathcal{B}}$.

□

8 The Need for Priors

A Problem In practical applications, sample D will typically have some of its frequency counters $f_{kil} = 0$. In that case, the supervised likelihood $S^S(D;\Theta^L)$ in the ordinary parameterization (1) is maximized for a parameter vector with some of the parameters (conditional or class probabilities) equal to 0. This poses a problem for supervised likelihood optimization within the model $\mathcal{M}^L$: if $S^S(D;\Theta^S)$ is maximized for some (α^S, Φ^S) with $\Phi^S_{kil} = 0$ for some k, i, l, then the supervised likelihood $S^L(D;\Theta^L)$ in $\mathbf{\Theta^L}$ is maximized for some (α^L, Φ^L) with $\Phi^L_{kil} = -\infty$ and S^L will have no maximum over $\mathbf{\Theta^L}$. This makes our optimization task hard to perform.

The same problem can arise in more subtle situations, as illustrated by the following example:

Example 4 *(Divergence of S^L).* Consider a domain of three binary variables X_0, X_1, X_2, with $D = \{(1,1,1),(1,1,2),(1,2,2),(2,1,1),(2,2,2))\}$. $S^L(D;(\alpha,\Phi))$ is maximized (for example, see Example 1) at $\alpha = \Phi_{\cdot 12} = \Phi_{\cdot 22} = (0,0)$ and $\Phi_{\cdot 11} = -\Phi_{\cdot 21} = (b,-b)$ with $b \to \infty$. This can be seen as follows. All vectors with $x_1 = x_2$ have a conditional likelihood of 0.5, which cannot be improved, since there is always a pair of them with contradicting class. Finally observe, that $P(X_0 = 1 \mid X_1 = 1, X_2 = 2, \Theta) \underset{b\to\infty}{\longrightarrow} 1$.

We can avoid such problems by introducing Bayesian parameter priors. We impose a strictly concave prior, which goes to $-\infty$ along with any parameter. We also introduce a set of constraints on the parameters, namely $\sum_k \alpha^L_k = 0$ and for all i, l $\sum_k \Phi^L_{kil} = 0$, thus ensuring the existence of a single maximum of the new objective

$$\begin{aligned} S^+(D;\Theta) :=& \log\left(P(X_0 \mid X_1,\ldots,X_M,\Theta)P(\Theta)\right. \\ =& S^L(D;\Theta) + \log P(\Theta). \end{aligned} \tag{27}$$

over the restricted parameter space.

Note that maximizing $S^+(D;\Theta)$ is equivalent to *Bayesian Maximum A Posteriori (MAP) estimation* based on the conditional model $\mathcal{M}_L$ and prior $P(\Theta)$. We have shown in earlier work that for ordinary, unsupervised Naive Bayes, whenever we are in danger of over-fitting the training data (ie. for small sample sizes), future data predictions can be *greatly* improved by imposing a prior on the parameters and using *Bayesian MAP* or *Bayesian Evidence* rather than ML prediction

(Kontkanen *et al.* , 2000). Supervised NB is inclined to worse over-fitting than unsupervised NB, since it uses the same amount of parameters to model a much smaller domain. In the experiments reported in the next section, we decided to use a strictly technical prior that draws all parameters a little bit closer to zero (i.e. zero-influence), moderating over-fitting. The prior used here is simply the normalized product of all parameters:

$$P(\Theta) := \prod_k \left(\frac{\exp \alpha_k}{\sum_{k'} \exp \alpha_{k'}} \prod_{i,l} \frac{\exp \Phi_{kil}}{\sum_{k''} \exp \Phi_{k''il}} \right). \tag{28}$$

9 Empirical Evaluation

The goal of this empirical study was to illustrate the usefulness of the supervised learning framework presented by using the Naive Bayes classifier as an example predictive model. The globally optimal supervised parameters were obtained by maximizing (27) using a simple hill-climbing algorithm with standard line search. As the test bed, we used 32 real-world data sets from the UCI repository. Where data was continuous, it was discretized as described at `http://www.cs.Helsinki.FI/u/pkontkan/Data/`. The cross-validation method was leave-one-out (loo), avoiding variance due to random splits.

Table 1 lists the data sets used — ordered by size — and both the log-score and the percentage of correct predictions obtained by using standard Naive Bayes (with uniform prior and evidence prediction) and our supervised method. The 'winner scores' are boldfaced.

We observe, that in 26 out 32 cases the supervised method has produced a better log-score. On a few small data sets, it apparently over-fitted the training data more. *On all larger data sets it consistently outperformed standard NB, in several cases by quite a margin. In contrast, for the few smaller data sets where standard NB outperformed supervised NB, it did so by much smaller margins.* This is exactly the type of behavior that we had expected. For completeness we mention, that for the 0/1-loss, the supervised method has won by a score of 18:13. Again it wins on larger data sets in agreement with results in (Ng & Jordan, 2001).

10 Conclusion and Future Work

We showed that by using the parameter transformation described in this paper, one can effectively find the parameters maximizing the global supervised likelihood (or rather, the posterior distribution) of the Naive Bayes model. The empirical results reported suggest that this technique can be used for improving the accuracy of the Naive Bayes classifier in many cases by a considerable amount. Furthermore, we showed that our theoretical result can be extended to more general classes of Bayesian network models including the tree-augmented NB model. In the future we intend to extend our experiments to involve also such

Table 1: Leave-one-out cross-validation results

data set	size	uns. NB	sup. NB
Mushrooms	8124	0.131/95.57	**0.002/100.00**
Page Bl.	5473	0.172/94.74	**0.102/96.29**
Abalone	4177	2.920/23.49	**2.082/25.95**
Segment.	2310	0.181/94.20	**0.118/97.01**
Yeast	1484	1.155/55.59	**1.140/57.75**
German Cr.	1000	0.535/**75.20**	**0.524**/74.30
TicTacToe	958	0.544/69.42	**0.099/98.33**
Vehicle S.	846	1.731/63.95	**0.682/72.22**
Annealing	798	0.161/93.11	**0.053/99.00**
Diabetes	768	0.488/**76.30**	**0.479**/75.78
BC (Wisc.)	699	0.260/**97.42**	**0.105**/96.42
Austr. Cr.	690	0.414/**86.52**	**0.334**/85.94
Balance Sc.	625	0.508/92.16	**0.231/93.60**
C. Voting	435	0.632/90.11	**0.102/96.32**
Mole Fever	425	**0.213/90.35**	0.241/88.71
Dermat.	366	**0.042**/97.81	0.079/97.81
Ionosphere	351	0.361/92.31	**0.171/92.59**
Liver	345	0.643/64.06	**0.629/68.70**
Pr. Tumor	339	1.930/48.97	**1.769/49.26**
Ecoli	336	**0.518**/80.36	0.562/**81.85**
Soybean	307	0.647/85.02	**0.314/90.23**
HD (Cleve)	303	1.221/**58.09**	**1.214**/55.78
HD (Hung.)	294	0.562/**83.33**	**0.444**/82.99
Breast C.	286	0.644/**72.38**	**0.606**/70.98
HD (Stats)	270	0.422/**85.19**	**0.419**/83.33
Thyroid	215	**0.054/98.60**	0.132/94.88
Glass Id.	214	0.913/**70.09**	**0.809**/69.63
Wine	178	**0.056/97.19**	0.169/96.63
Hepatitis	155	0.560/79.35	**0.392/82.58**
Iris Plant	150	**0.169**/94.00	0.265/**94.67**
Lymphogr.	148	0.436/85.81	**0.375/86.49**
Postop.	90	0.840/**67.78**	**0.837**/66.67

more complicated models. We also plan to investigate how to prevent over-fitting with small data samples by using theoretically more elaborate parameter priors than the simple technical prior used in this paper.

Acknowledgements. This research has been supported by the National Technology Agency, and the Academy of Finland. The authors wish to thank Wray Buntine for many useful comments.

References

Bishop, C.M. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.

Dawid, A.P. 1976. Properties of diagnostic data distributions. *Biometrics*, **32**, 647–658.

Dawid, A.P. 1982. The Well-Calibrated Bayesian. *Journal of the American Statistical Association*, **77**, 605–610.

Friedman, N., Geiger, D., & Goldszmidt, M. 1997. Bayesian Network Classifiers. *Machine Learning*, **29**, 131–163.

Greiner, R., & Zhou, W. 2001. *Discriminant Parameter Learning of Belief Net Classifiers*. from http://www.cs.ualberta.ca/∼greiner/.

Greiner, R., Grove, A., & Schuurmans, D. 1997 (August). Learning Bayesian Nets that Perform Well. *In: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI-97).*

Heckerman, D., & Meek, C. 1997. Models and selection criteria for regression and classification. *Pages 223–228 of:* Geiger, D., & Shenoy, P. (eds), *Uncertainty in Arificial Intelligence 13*. Morgan Kaufmann Publishers, San Mateo, CA.

Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., & Grünwald, P. 2000. On Predictive Distributions and Bayesian Networks. *Statistics and Computing*, **10**, 39–54.

Kontkanen, P., Myllymäki, P., & Tirri, H. 2001. Classifier Learning with Supervised Marginal Likelihood. *In:* Breese, J., & Koller, D. (eds), *Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence (UAI'01)*. Morgan Kaufmann Publishers.

Ng, A.Y., & Jordan, M.I. 2001. On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. *Advances in Neural Information Processing Systems*, **14**, 605–610.

Sarle. 2001. *Neural Network FAQ, part 2 of 7: Learning, periodic posting to the Usenet newsgroup comp.ai.neural-nets*. ftp://ftp.sas.com/pub/neural/FAQ.html.

Turdaliev, N. 1999. *Calibration and Bayesian Learning*. http://minneapolisfed.org/research/wp/wp596.ps.