Pp. 72–83 in *STeP 2002 – Intelligence, The Art of Natural and Artificial and Artificial: Proceedings of the 10th Finnish Artificial Intelligence Conference*, edited by P. Ala-Siuru and S. Kaski. Finnish Artificial Intelligence Society, 2002.

SUPERVISED NAIVE BAYES PARAMETERS

Hannes Wettig**, Peter Grünwald*$^\circ$*, Teemu Roos***,*
Petri Myllymäki* *and Henry Tirri**

* Complex Systems Computation Group (CoSCo)
Helsinki Institute for Information Technology (HIIT)
University of Helsinki and Helsinki University of Technology
P.O. Box 9800, FIN-02015 HUT, Finland
$^\circ$ CWI, P.O. Box 94079, NL-1098 SJ Amsterdam, The Netherlands.

Bayesian network models are widely used for supervised prediction tasks such as classification. The Naive Bayes (NB) classifier in particular has been successfully applied in many fields. Usually its parameters are determined using 'unsupervised' methods such as likelihood maximization. This can lead to seriously biased prediction, since the independence assumptions made by the NB model rarely ever hold. It has not been clear though, how to find parameters maximizing the supervised likelihood or posterior globally. In this paper we show, how this supervised learning problem can be solved efficiently. We introduce an alternative parametrization 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. We present test results that show this is feasible and highly beneficial.

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 [6, 10, 5, 9, 4]. Nevertheless, in most applications related to this type of task, model parameters are still determined using unsupervised methods such as ordinary likelihood maximization. One of the main reasons for this discrepancy is the difficulty in finding the global maximum of the supervised likelihood. In this paper we show how this problem can be solved for the Naive Bayes (NB) classifier.

We find the supervised maximum likelihood parameters by parametrizing the NB model in a different manner; we take the logarithms of the original parameters and drop the sum-to-one constraints. The new parametrization 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. We first in Section 2 review the standard (unsupervised) Naive Bayes classifier and its supervised version. Then we show that when this model is parametrized 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. The supervised likelihood of the data, as a function of the parameters of the L-model, becomes concave, while the parameter set itself is convex. Section 5 provides alternative interpretations of the L-model. In Section 6 we argue that for technical reasons, it is useful to equip our models with a prior as to effectively maximize the 'supervised Bayesian posterior' rather than the plain supervised likelihood. Finally, in Section 7, 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 8.

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 $l \in \{1, \ldots, n_i\}$. Without loss of generality be X_0 the *class variable* (the one we want to predict), 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. [8]) 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 $\alpha^S = P(X_0|\Theta^S)$ is the default distribution over the class, and each $\Phi^S_{ki} = P(X_i|X_0 = k, \Theta^S)$ is a distribution over the values of X_i given the class. We restrict our parameters to lie in the set $\mathbf{\Theta^S}$ defined as:

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

Note that $\overline{\mathbf{\Theta^S}}$, the *closure* of $\mathbf{\Theta^S}$, is the set of all parameter vectors that correspond to some Naive Bayes distribution. $\mathbf{\Theta^S}$ itself is the set of all parameter vectors corresponding to a Naive Bayes distribution with only strictly positive probabilities. Without essential loss of generality we may restrict ourselves to parameters in $\mathbf{\Theta^S}$, as we shall see in Section 6.

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

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

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 $d_{ji} = l$.

With the standard NB classifier, for given data D, one infers the maximum likelihood (ML) parameters $\hat{\Theta}^S$ by 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:

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

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 [4, 5, 6, 9, 10]. 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

$$S^S(d; \Theta^S) := \log P(k | x_1, \ldots, x_M, \Theta^S) = \log \frac{\alpha_k^S \prod_{i=1}^M \Phi_{kix_i}^S}{\sum_{k'=1}^{n_0} \alpha_{k'}^S \prod_{i=1}^M \Phi_{k'ix_i}^S}. \tag{4}$$

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

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

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 5).

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):

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

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

Example 1 *($\mathbf{\Theta^S}$-parametrization 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 $\Phi_{111}^S = \Phi_{211}^S = b \in (0, 1)$. For *all* values of b, the supervised score[1] of any vector (x_0, x_1) is given by

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

[1] We use the word ‘score’ in order to stress that the log-likelihood is the objective to be optimized.

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 $P(X_0|X_1, \Theta^{(1)}) = P(X_0|X_1, \Theta^{(2)})$. While all $\Theta^S \in \mathbf{\Theta^S}$ index a different *joint* distribution, some of them index the same *conditional* distribution.

The problem with maximizing the supervised likelihood is that in the conventional NB parametrization 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 $\Phi^S_{211} = 1 - \Phi^S_{212}$ as it peaks twice.

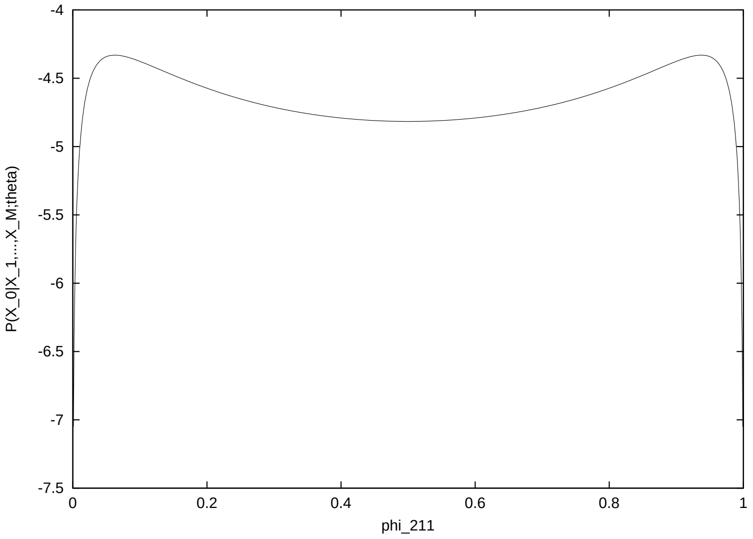

Figure 1: the supervised log-likelihood peaks twice as Φ^S_{211} varies.

Because of this non-concavity, we have to use complicated optimization methods to maximize the supervised score (in contrast to the unsupervised NB case, we cannot solve the problem analytically). Such algorithms may converge slowly due to the non-concavity of the score.

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

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

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

$$P(X_0 = k|X_1 = x_1, \ldots, X_M = x_M, (\alpha^L, \Phi^L)) := \frac{\exp(\alpha_k^L) \prod_{i=1}^M \exp(\Phi_{kix_i}^L)}{\sum_{k'=1}^{n_0} \exp(\alpha_{k'}^L) \prod_{i=1}^M \exp(\Phi_{k'ix_i}^L)}. \tag{6}$$

The distributions $P(X_0|X_1, \ldots, X_M, (\alpha^L, \Phi^L))$ are extended to several outcomes by independence (i.e. taking product distributions). One immediately verifies that, for all $x_1, \ldots, x_M$ it is

$$\sum_{k \in \{1,\ldots,n_0\}} P(k|x_1, \ldots, x_m, (\alpha^L, \Phi^L)) = 1;$$

and that each term in the sum is positive. This confirms that $P(X_0|x_1, \ldots, x_M, (\alpha^L, \Phi^L))$ given by (6) indeed defines a conditional distribution over X_0 for all $(\alpha^L, \Phi^L) \in \mathbf{\Theta^L}$, and all $x_1, \ldots, x_M$.

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

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

extended to a 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 $\mathbf{\Theta^L}$:

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

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 following we show that these two seemingly different conditional models are in fact equal.

4 EQUALITY OF $\mathcal{M}^S$ AND $\mathcal{M}^L$; CONCAVITY

To see that $\mathcal{M}^S$ and $\mathcal{M}^L$ are related, define the *log-transformation* $L : \mathbf{\Theta^S} \to \mathbf{\Theta^L}$ 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) given by:

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

By plugging (10) into (8) and further into (7), we see that for all $\Theta^S \in \mathbf{\Theta^S}$ it is

$$P(X_0|X_1, \ldots, X_m, \Theta^S) = P(X_0|X_1, \ldots, X_m, L(\Theta^S)).$$

This shows that $\mathcal{M}^S \subseteq \mathcal{M}^L$: each parameter set Θ^S indexing a distribution in $\mathcal{M}^S$ is transformed into a parameter set 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 parametrization 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 $(c_0, (c_{11}, \ldots, c_{1M}), \ldots, (c_{n_01}, \ldots, c_{n_0M}))$. Define for $k \in \{1, \ldots, n_0\}$

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

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

$$\begin{aligned} \Phi_{kil}^S &:= \exp(\Phi_{kil}^{(c)}) = \exp(\Phi_{kil}^L + c_{ki}), \\ \alpha_k^S &:= \exp(\alpha_k^{(c)}) = \exp(\alpha_k^L + c_0 - \sum_{i=1}^{M} c_{ki}). \end{aligned} \tag{13}$$

Evidently, for all k and i we can choose c_{ki} such that $\sum_{l=1}^{n_i} \Phi_{kil}^S = 1$, and subsequently c_0 such that $\sum_k \alpha_k^S = 1$. 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 parametrization of the supervised Naive Bayes model $\mathcal{M}^S$; we call $\mathbf{\Theta^L}$ the *L-parametrization* of $\mathcal{M}^S$.

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

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 proof of item (ii) see Section 5 and our technical report [13].

Together, items (i) and (ii) demonstrate that finding the NB distribution maximizing the supervised likelihood in the L-parametrization 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.

Remark The log-likelihood does not have local maxima over the standard parametrization $\mathbf{\Theta^S}$ (see [13]), but neither is it concave (i.e. it will have ripples and wrinkles). Greiner and Zhou have used the L-parametrization in [6] and report that "it worked better" [than the standard parametrization]. 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. We had $\alpha^S := (0.1, 0.9)$, now we set $\alpha^L := L(\alpha^S) = (\log 0.1, \log 0.9)$. Figure 2 gives some clue of how it is possible to concavify the objective, and why it could peak twice in Example 2.

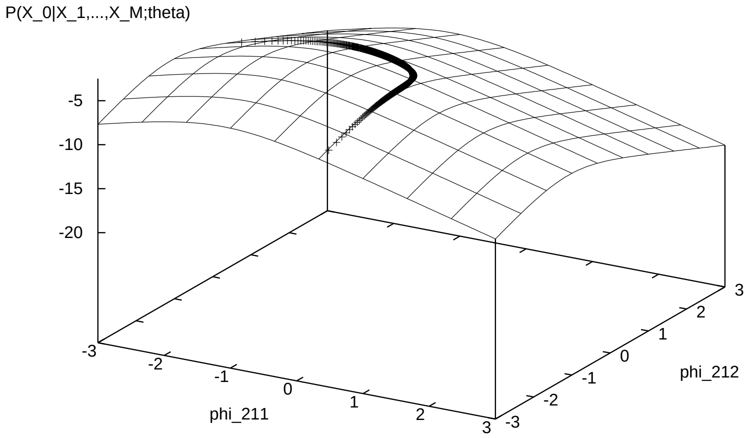

Figure 2: the supervised log-likelihood has become a concave function of the new parameters Φ^L_{211} and Φ^L_{212}; the pointed line shows the transform of Φ^S_{211} in Figure 1.

The L-parametrization allows us to think of the Naive Bayes classifier as a discriminative (diagnostic) rather than as a generative (sampling) model, see e.g. [2, 10]. 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'; [7, 10]). Figure 3 gives an interpretation of this, depicting both Naive Bayes and the L-model in their Bayesian network guises.

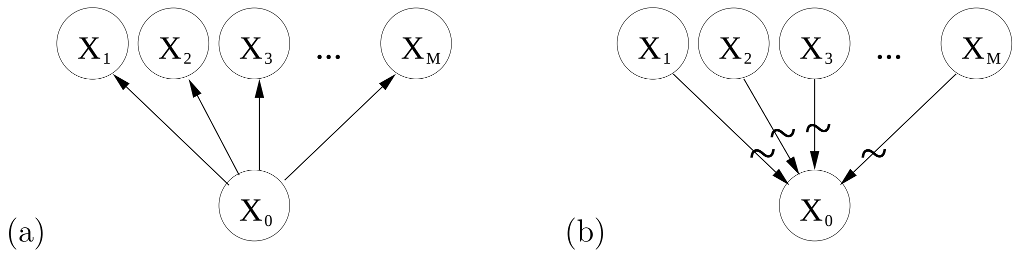

Figure 3: standard NB net (left) and L-model (right). All arcs have been reversed and the resulting product distribution has been replaced by softmax (denoted by tildes).

Technically, to get a logistic regression model from the L-model, one would create one (binary) regressor variable for each possible value of an attribute and one (binary) output variable for each value of the class.

Looking at the L-model in this light also proves concavity of $S^L(D;\Theta^L)$ (Theorem 2 (ii)), since the supervised log-likelihood is known to be concave for logistic regression models (see e.g. [11], p.234).

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. [1]. 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. The parameters of the neural network are usually optimized to maximize the conditional likelihood, or equivalently the so called *cross-entropy.* It follows from Theorem 2 that the objective function of the neural network is also concave.

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:

$$\sum_{j=1}^{N} P(X_0 = k|d_{j1},\ldots,d_{jM},\Theta^L) = h_k, \text{ and } \sum_{j:d_{ji}=l} P(X_0 = k|d_{j1},\ldots,d_{jM},\Theta^L) = f_{kil}. \tag{14}$$

That is, we have found good parameters for the supervised task exactly when we are 'well-calibrated' wrt. D and all subsets $D_{il} := \{d_j|d_{ji} = l\}$ in the sense of [3]. Thus optimizing Θ^L according to S^L means 'recalibrating' ourselves using $\sum_{i=1}^{M} n_i + 1$ calibration tests simultaneously. Here the i.i.d. assumption of our model saves us from becoming 'incoherent' as we recalibrate, see [3].

As a spin-off we find, that using the L-model we can solve *any* calibration problem of the form

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

– 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 [12]. With only limited data availability the calibration tests implicit to the Naive Bayes model (i.e. $\mathcal{F}_{NB} = \{f_{il} : f_{il}(d) = 1 \Leftrightarrow d_i = l\} \cup \{1\}$) seem to be a sensible choice in many cases. Other choices can be made that do not need to correspond to any Bayesian model at all. 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 $\mathcal{F}_c = \{f_{il} : |D_{il}| \geq c\} \cup \{1\}$. For small data sets the resulting model may consist of considerably fewer parameters (depending on c).

6 THE NEED FOR A PRIOR

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 at (α^S,Φ^S) with $\Phi^S_{kil} = 0$ for some k, i, l, then the supervised likelihood $S^L(D;\Theta^L)$ in $\overline{\mathbf{\Theta^L}}$ is maximized at 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 (cf. 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 $P(X_0 = 1 \mid X_1 = 1, X_2 = 2, \Theta) \xrightarrow[b\to\infty]{} 1$.

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 $\sum_k \alpha_k^L = 0$ and for all i, l $\sum_k \Phi_{kil}^L = 0$, thus ensuring the existence of a single maximum of the new objective

$$S^+(D; \Theta) := \log\left(P(X_0 \mid X_1, \ldots, X_M, \Theta) P(\Theta)\right) = S^L(D; \Theta) + \log P(\Theta). \tag{15}$$

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 (i.e. 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 [8]. 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:

$$P(\Theta) := \prod_k \left(\frac{\exp \alpha_k}{\sum_{k'} \exp \alpha_{k'}} \prod_{i,l} \frac{\exp \Phi_{kil}}{\sum_{k''} \exp \Phi_{k''il}} \right). \tag{16}$$

7 EMPIRICAL EVALUATION

We now want to illustrate the usefulness of our method by reporting the results of our test runs. The globally optimal supervised parameters were obtained by maximizing (15), using gradient ascent with standard line search. As the test bed, we took 32 real-world data sets from the UCI repository. Continuous data was discretized (discretizations 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 [10].

8 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 of the Naive

Table 1: Leave-one-out cross-validation results. Name and size of the data set, prediction loss of unsupervised vs. supervised Naive Bayes (log-score/percentage of correct predictions). Best scores boldfaced.

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

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. In [13] we extend our theoretical results to more general classes of Bayesian network models including TAN (tree-augmented NB) models. In the future we intend to perform experiments that also involve such more complicated models. We further plan to investigate how to prevent over-fitting of small data samples by using theoretically more elaborate parameter priors than the simple technical prior used here.

Acknowledgements. This research has been supported by the National Technology Agency, and the Academy of Finland.

REFERENCES

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

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

[3] A.P. Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77:605–610, 1982.

[4] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[5] R. Greiner, A. Grove, and D. Schuurmans. Learning Bayesian nets that perform well. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Providence, August 1997.

[6] R. Greiner and W. Zhou. Discriminant parameter learning of belief net classifiers, 2001. from http://www.cs.ualberta.ca/∼greiner/.

[7] D. Heckerman and C. Meek. Models and selection criteria for regression and classification. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI-97)*, Providence, August 1997.

[8] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. On predictive distributions and Bayesian networks. *Statistics and Computing*, 10:39–54, 2000.

[9] P. Kontkanen, P. Myllymäki, and H. Tirri. Classifier learning with supervised marginal likelihood. In J. Breese and D. Koller, editors, *Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence (UAI'01)*. Morgan Kaufmann Publishers, 2001.

[10] A.Y. Ng and M.I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. *Advances in Neural Information Processing Systems*, 14:605–610, 2001.

[11] T. Santner and D. Duffy. *The Statistical Analysis of Discrete Data*. Springer-Verlag, New York, 1989.

[12] N. Turdaliev. Calibration and Bayesian learning, 1999. http://minneapolisfed.org/research/wp/wp596.ps.

[13] H. Wettig, P. Grünwald, T. Roos, P. Myllymäki, and H. Tirri. On supervised learning of Bayesian network parameters. Technical Report 2002–1, Helsinki Institute for Information Technology (HIIT), 2002. http://cosco.hiit.fi/Articles/hiit2002-1.ps.