Discriminative Learning of Bayesian Network Classifiers via the TM Algorithm

Guzmán Santafé, Jose A. Lozano, and Pedro Larrañaga

Intelligent Systems Group,
Department of Computer Science and Artificial Intelligence,
University of the Basque Country, Spain
{guzman, lozano, ccplamup}@si.ehu.es

Abstract. The learning of probabilistic classification models can be approached from either a generative or a discriminative point of view. Generative methods attempt to maximize the unconditional log-likelihood, while the aim of discriminative methods is to maximize the conditional log-likelihood. In the case of Bayesian network classifiers, the parameters of the model are usually learned by generative methods rather than discriminative ones. However, some numerical approaches to the discriminative learning of Bayesian network classifiers have recently appeared. This paper presents a new statistical approach to the discriminative learning of these classifiers by means of an adaptation of the TM algorithm [1]. In addition, we test the TM algorithm with different Bayesian classification models, providing empirical evidence of the performance of this method.

1 Introduction

Supervised classification is a part of machine learning which has a large number of applications in many tasks such as pattern recognition and medical diagnosis. In general, supervised classification assumes the existence of two different kinds of variables: the predictive variables, $\boldsymbol{X} = (X_1, \ldots, X_n)$, and the class variable or response, C. A supervised classifier attempts to learn the relationship between the predictive and the class variables. Hence, it is able to assign a class value to a new data sample $\boldsymbol{x} = (x_1, \ldots, x_n)$ whose response is unknown.

The learning of a classification model can be approached, among other paradigms, from either a generative or a discriminative point of view [2, 3, 4, 5]. Generative classifiers, also called informative classifiers, obtain the parameters of the model by maximizing the unconditional log-likelihood function. Models like discriminant analysis [6] or naïve Bayes [7] are typical examples of generative classifiers. On the other hand, discriminative classifiers obtain the parameters of the model by maximizing the conditional log-likelihood function (e.g. logistic regression [8]) or just model the class boundaries (e.g. neural networks [9]).

Bayesian networks [10, 11] are widely used for classification tasks due to their simplicity and accuracy. Usually, Bayesian network learners are generative but, recently, there has been a considerable growth of interest in the discriminative learning of Bayesian network classifiers [12, 13]. The use of a discriminative

L. Godo (Ed.): ECSQARU 2005, LNAI 3571, pp. 148–160, 2005.

learning for classification purposes seems more natural because the classification model directly maximizes the probability of the class given the predictive variables, which is what we use to classify new instances. However, generative classifiers can sometimes yield better performance than discriminative ones [4]. Normally, generative learning performs better in those cases where the classification model learned from a dataset is close to the one that has generated this dataset. On the other hand, when the learned model is different from the original one, generative classifiers normally perform worse than discriminative ones [3].

The aim of this paper is to propose a statistical approach to the discriminative learning of Bayesian network classifiers, in contrast to other more generic numerical optimization schemes [12, 13], via the adaptation of the TM algorithm. The TM algorithm [1] is a general iterative process that allows the maximization of the conditional log-likelihood in models where the unconditional log-likelihood function is easier to maximize, which is the case of Bayesian networks. We introduce the theoretical development of the algorithm in the context of Bayesian classification models. Additionally, we evaluate the performance of Bayesian network classifiers learned with the TM algorithm by comparing their estimated accuracy with the estimated accuracy of the classifiers learned by a classical generative method. This empirical evaluation is performed using simple models such as naïve Bayes [7] and tree augmented naïve Bayes (TAN) [14].

The rest of this paper is organized as follows. In Section 2, the general structure of the TM algorithm is described, and this structure is particularized to the exponential family of distributions. In Section 3, we adapt the TM algorithm to be used with Bayesian network classifiers. Section 4 provides empirical results of the performance of the TM algorithm and, finally, the conclusions yielded from the paper are exposed in Section 5.

2 The TM Algorithm by Edwards and Lauritzen

This section introduces the TM algorithm in the same way as [1] but bearing in mind the classification purpose of the model that we want to learn. Thus, we expect to give the reader a general and intuitive idea about how the TM algorithm works.

2.1 General Structure of the TM Algorithm

Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be a vector where each X_i, with $i = 1, \ldots, n$, is a predictive variable, and let C be the class variable. Since we are focusing on classification problems, we consider C a unidimensional variable, but in general both $\boldsymbol{X}$ and C could be multivariate variables.

We denote the unconditional, marginal and conditional log-likelihood functions as follows:

$$l(\boldsymbol{\theta}) = \log f(\boldsymbol{x}, c|\boldsymbol{\theta}) \quad , \quad l_{\boldsymbol{x}}(\boldsymbol{\theta}) = \log f(\boldsymbol{x}|\boldsymbol{\theta}) \quad , \quad l^{\boldsymbol{x}}(\boldsymbol{\theta}) = \log f(c|\boldsymbol{x}, \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is the parameter set of the unconditional probability distribution for the variable $(\boldsymbol{X}, C)$.

The foundations of the TM algorithm are based on the tilted unconditional log-likelihood function, $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$. This function is an approximation to $l^{\boldsymbol{x}}(\boldsymbol{\theta})$, which we want to maximize, at point $\boldsymbol{\theta}_r$. Note that $l^{\boldsymbol{x}}(\boldsymbol{\theta})$ can be expressed in terms of the unconditional and the marginal log-likelihood:

$$l^{\boldsymbol{x}}(\boldsymbol{\theta}) = l(\boldsymbol{\theta}) - l_{\boldsymbol{x}}(\boldsymbol{\theta})$$

Therefore, if we expand $l_{\boldsymbol{x}}(\boldsymbol{\theta})$ in a first order Taylor series about the point $\boldsymbol{\theta}_r$, and then omit the terms which are constant with respect to $\boldsymbol{\theta}$, we can approximate $l^{\boldsymbol{x}}(\boldsymbol{\theta})$ by $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$ as follows:

$$l^{\boldsymbol{x}}(\boldsymbol{\theta}) \approx q(\boldsymbol{\theta}|\boldsymbol{\theta}_r) = l(\boldsymbol{\theta}) - \boldsymbol{\theta}^T \dot{l}_{\boldsymbol{x}}(\boldsymbol{\theta}_r) \tag{1}$$

where $\dot{l}_{\boldsymbol{x}}(\boldsymbol{\theta}_r)$ is the derivative of $l_{\boldsymbol{x}}(\boldsymbol{\theta})$ at point $\boldsymbol{\theta}_r$.

The tilted unconditional log-likelihood function and the conditional log-likelihood have the same gradient at $\boldsymbol{\theta}_r$, thus, we can maximize $l^{\boldsymbol{x}}(\boldsymbol{\theta})$ by maximizing $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$

Since the approximation of $l^{\boldsymbol{x}}(\boldsymbol{\theta})$ is at point $\boldsymbol{\theta}_r$, we need an iterative process in order to maximize the conditional log-likelihood. This process alternates between two steps, T and M. In the T step, the tilted unconditional log-likelihood described above is obtained. The second step of the algorithm, the M step, consists in maximizing the tilted unconditional log-likelihood function:

$$\boldsymbol{\theta}_{r+1} = \arg\max_{\boldsymbol{\theta}} \quad q(\boldsymbol{\theta}|\boldsymbol{\theta}_r) \tag{2}$$

Under regularity conditions of the usual type and due to the fact that the expected score statistic for the conditional model is equal to 0, $\dot{l}_{\boldsymbol{x}}(\boldsymbol{\theta})$ can be calculated as the expectation of the score statistic for the unconditional model.

$$\dot{l}_{\boldsymbol{x}}(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}}\{\dot{l}_{\boldsymbol{x}}(\boldsymbol{\theta})|\boldsymbol{x}\} = E_{\boldsymbol{\theta}}\{\dot{l}(\boldsymbol{\theta}) - \dot{l}^{\boldsymbol{x}}(\boldsymbol{\theta})|\boldsymbol{x}\} = E_{\boldsymbol{\theta}}\{\dot{l}(\boldsymbol{\theta})|\boldsymbol{x}\}$$

Therefore, the M step involves the solution of the following equation:

$$E_{\boldsymbol{\theta}_r}\{\dot{l}(\boldsymbol{\theta}_r)|\boldsymbol{x}\} = \dot{l}(\boldsymbol{\theta}) \tag{3}$$

In summary, the relevance of the TM algorithm is that it allows us to obtain a model that maximizes the conditional log-likelihood, $l^{\boldsymbol{x}}(\boldsymbol{\theta})$, by using the unconditional log-likelihood, $l(\boldsymbol{\theta})$. This is very useful for models like Bayesian network classifiers, where the obtention of the unconditional (generative) model is much easier than the obtention of the conditional (discriminative) one.

The TM algorithm begins by making its initial parameters the ones which maximize the unconditional log-likelihood given the dataset. Then, both the T and the M steps are repeated until the value of the conditional log-likelihood converges. See [15] for details about the convergence of the TM algorithm.

2.2 The TM Algorithm for the Exponential Family

The TM algorithm can be easily particularized for probability distributions belonging to the exponential family. In this case, the unconditional log-likelihood is given by the following formula:

$$l(\boldsymbol{\theta}) = \boldsymbol{\alpha}^T u(c, \boldsymbol{x}) + \boldsymbol{\beta}^T v(\boldsymbol{x}) - \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{4}$$

where

$$\psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \log \int exp\{\boldsymbol{\alpha}^T u(c, \boldsymbol{x}) + \boldsymbol{\beta}^T v(\boldsymbol{x})\} \mu(dc|\boldsymbol{x}) \mu(d\boldsymbol{x})$$

Let us introduce a new parametrization for $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\eta})$, with:

$$\boldsymbol{\eta} = \frac{\partial}{\partial \boldsymbol{\beta}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta})$$

Moreover, if we define two new random variables, $\boldsymbol{\mathcal{U}} = u(C, \boldsymbol{X})$ and $\boldsymbol{\mathcal{V}} = v(\boldsymbol{X})$, it can be demonstrated that the maximum likelihood parameters are $\hat{\boldsymbol{\theta}} = (\boldsymbol{u}, \boldsymbol{v})$ with $\boldsymbol{u} = E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}\}$ and $\boldsymbol{v} = \boldsymbol{\eta} = E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{V}}\}$.

Following the general structure of the TM algorithm, Equation 3 has to be solved in order to maximize the approximation to the conditional log-likelihood given by $q(\boldsymbol{\theta}|\boldsymbol{\theta}_r)$. Thus, we have:

$$E_{\boldsymbol{\theta}} \left\{ \frac{\partial}{\partial \boldsymbol{\theta}} l(\boldsymbol{\theta}) \middle| \boldsymbol{x} \right\} = E_{\boldsymbol{\theta}} \left\{ \boldsymbol{\mathcal{U}} - \frac{\partial}{\partial \boldsymbol{\alpha}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}), \boldsymbol{\mathcal{V}}^T \ \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} - \frac{\partial}{\partial \boldsymbol{\beta}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} \middle| \boldsymbol{x} \right\}$$

$$= \left(E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\} - E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}\}, (E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{V}}\} - \boldsymbol{\eta})^T \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} \right) = (E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\} - E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}\}, 0) \tag{5}$$

and also:

$$\dot{l}(\boldsymbol{\theta}) = \left(\boldsymbol{\mathcal{U}} - \frac{\partial}{\partial \boldsymbol{\alpha}} \psi(\boldsymbol{\alpha}, \boldsymbol{\eta}), \boldsymbol{\mathcal{V}}^T \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} - \frac{\partial}{\partial \boldsymbol{\beta}} \psi(\boldsymbol{\alpha}, \boldsymbol{\beta}) \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\eta}} \right) = (\boldsymbol{\mathcal{U}} - E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}\}, 0) \tag{6}$$

Finally, the solution of Equation 3 gives the value of the sufficient statistics at the $r + 1$-th iteration of the TM algorithm:

$$\begin{aligned} \boldsymbol{u}_{r+1} &= \boldsymbol{u}_r + \boldsymbol{u}_0 - E_{\boldsymbol{\theta}_r}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\} \\ \boldsymbol{\theta}_{r+1} &= \hat{\boldsymbol{\theta}}(\boldsymbol{u}_{r+1}, \boldsymbol{v}) \end{aligned} \tag{7}$$

where the initial sufficient statistics, $\boldsymbol{u}_0$ and $\boldsymbol{v}$, are given by the maximum likelihood estimators obtained from the data set. Moreover, $\hat{\boldsymbol{\theta}}(\boldsymbol{u}_{r+1}, \boldsymbol{v})$ denotes the maximum likelihood estimations of $\boldsymbol{\theta}$ obtained from sufficient statistics $\boldsymbol{u}_{r+1}$ and $\boldsymbol{v}$.

Generally, it may happen that an iteration of the TM algorithm yields an illegal set of parameters $\boldsymbol{\theta}$ or that the conditional log-likelihood decreases from one iteration to another. These situations must be corrected by applying a linear search. Thus, the sufficient statistics at step $r + 1$ are calculated as:

$$\boldsymbol{u}_{r+1} = \boldsymbol{u}_r + \lambda(\boldsymbol{u}_0 - E_{\boldsymbol{\theta}_r}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\}), \quad \text{with } \lambda \in (0, 1) \tag{8}$$

being λ the one that maximizes the conditional log-likelihood.

3 The TM Algorithm for Bayesian Classifiers

In this section we show how the TM algorithm can be adapted to the Bayesian classification models considered in this paper. Even when Bayesian networks belong to the exponential family, the adaptation of the calculations shown in Section 2.2 is not trivial. As an example, Section 3.1 shows the calculations needed

to apply the TM algorithm to a naïve Bayes model with multinomial variables. Calculations for the TAN model are similar. Therefore, for these models, Section 3.2 only shows the sufficient statistics, $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$, used in the algorithm. See [16] for more details about the adaptation of the TM algorithm to Bayesian network classifiers with dichotomic and multinomial variables.

3.1 The TM Algorithm for a Naïve Bayes with Multinomial Variables

We assume that each variable can take multiple states, therefore $C \in \{0, \ldots, v_0\}$ and $X_i \in \{0, \ldots, v_i\}$ with $v_0 + 1$ and $v_i + 1$ as the number of possible states for variables C and X_i, respectively.

The general algorithm for probability distributions of the exponential family requires the expression of the unconditional log-likelihood via Equation 4. This can be achieved by writing the naïve Bayes unconditional model as follows:

$$p(c, \boldsymbol{x}) = \frac{1}{(p(c))^{n-1}} \prod_{i=1}^{n} p(x_i, c) \tag{9}$$

In order to identify the sufficient statistics for the TM algorithm, we can rewrite the unconditional model as follows:

$$p(c, \boldsymbol{x}) = \left[\prod_{j=0}^{v_0} (p(C=j))^{w_j \prod_{l=0}^{j-1}(c-l) \prod_{l=j+1}^{v_0}(l-c)} \right]^{-(n-1)} \cdot$$

$$\prod_{i=1}^{n} \prod_{j=0}^{v_0} \prod_{k=0}^{v_i} (p(C=j, X_i=k))^{w_{jk}^i \prod_{l=0}^{j-1}(c-l) \prod_{l=j+1}^{v_0}(l-c) \prod_{l=0}^{k-1}(x_i-l) \prod_{l=k+1}^{v_i}(l-x_i)}$$

where w_j and w_{jk}^i are the following constants:

$$w_j = \frac{1}{\prod_{l=0}^{j-1}(j-l) \prod_{l=j+1}^{v_0}(l-j)} \quad , \quad w_{jk}^i = \frac{1}{\prod_{l=0}^{j-1}(j-l) \prod_{l=j+1}^{v_0}(l-j) \prod_{l=0}^{k-1}(k-l) \prod_{l=k+1}^{v_i}(l-k)}$$

Note that the values of w_j and w_{jk}^i have no influence on the selection of the sufficient statistics for the TM algorithm.

If we have a dataset with N samples, the unconditional log-likelihood can be written using the previous equation as follows:

$$l(\boldsymbol{\theta}) = \sum_{d=1}^{N} \Big\{ -(n-1) \sum_{j=0}^{v_0} \Big[w_j \prod_{l=0}^{j-1} (c^{(d)} - l) \prod_{l=j+1}^{v_0} (l - c^{(d)}) \log(p(C=j)) \Big] +$$

$$\sum_{i=1}^{n} \sum_{j=0}^{v_0} \sum_{k=0}^{v_i} \Big[w_{jk}^i \prod_{l=0}^{j-1} (c^{(d)} - l) \prod_{l=j+1}^{v_0} (l - c^{(d)}) \prod_{l=0}^{k-1} (c_i^{(d)} - l) \prod_{l=k+1}^{v_i} (l - c_i^{(d)}) \log(p(C=j, X_i=k)) \Big] \Big\}$$

where $c^{(d)}$ and $x_i^{(d)}$ are the values of variables C and X_i in the d-th sample of the dataset, respectively.

A few transformations in Equation 10 can match its terms with the ones from Equation 4. We thus obtain the sufficient statistics $\boldsymbol{\mathcal{U}} = (\boldsymbol{\mathcal{U}}_1, \boldsymbol{\mathcal{U}}_2)$ and $\boldsymbol{\mathcal{V}}$:

$$\begin{aligned} \boldsymbol{\mathcal{U}}_1 &= (M_0^s | s = 1, \ldots, v_0) \\ \boldsymbol{\mathcal{U}}_2 &= (M_{0x_i}^{st} | s = 1, \ldots, v_0 \ , \quad t = 1, \ldots, v_i \ , \quad i = 1, \ldots, n) \\ \boldsymbol{\mathcal{V}} &= (M_{x_i}^t | t = 1, \ldots, v_i \ , \quad i = 1, \ldots, n) \end{aligned}$$

where M_0^s, $M_{0x_i}^{st}$ and $M_{x_i}^t$ terms from the former equation are defined as:

$$M_0^s = \sum_{d=1}^{N} (C^{(d)})^s \ , \quad M_{0x_i}^{st} = \sum_{d=1}^{N} (C^{(d)})^s (X_i^{(d)})^t \ , \quad M_{x_i}^d = \sum_{r=1}^{N} (X_i^{(d)})^t \tag{10}$$

It was shown in Section 2.2 that, at each iteration, the calculation of $E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\}$ is needed to update the sufficient statistics $\boldsymbol{\mathcal{U}}$. This requires the following calculations:

$$E_{\boldsymbol{\theta}_r}[M_0^s|\boldsymbol{x}] = \sum_{d=1}^{N} \sum_{c=0}^{v_0} p_{\boldsymbol{\theta}_r}(C = c|\boldsymbol{X} = \boldsymbol{x}^{(d)}) c^s \tag{11}$$

$$E_{\boldsymbol{\theta}_r}[M_{0x_i}^{st}|\boldsymbol{x}] = \sum_{d=1}^{N} \sum_{c=0}^{v_0} p_{\boldsymbol{\theta}_r}(C = c|\boldsymbol{X} = \boldsymbol{x}^{(d)}) c^s (x_i^{(d)})^t$$

where $s = 1, \ldots, v_0$; $t = 1, \ldots, v_i$ and $i = 1, \ldots, n$.

Since we assume that the structure of the model is a naïve Bayes, we need to obtain $p(C = c)$ and $p(X_i = l|C = c)$ to calculate $p(C = c|\boldsymbol{X} = \boldsymbol{x}^{(k)})$, where $c = 1, \ldots, v_0$; $i = 1, \ldots, n$ and $l = 1, \ldots, v_i$. In order to obtain these probabilities, let us define a new set of sufficient statistics $\boldsymbol{\mathcal{N}} = (N_0^c, N_i^l, N_{0i}^{cl} | c = 1, \ldots, v_0; \ \ i = 1, \ldots, n; \ \ l = 1, \ldots, v_i)$. On the one hand, N_0^c counts the number of cases in which $C = c$, and N_i^l the number of cases in which $X_i = l$. On the other hand, N_{0i}^{cl} denotes the number of times that both $C = c$ and $X_i = l$ happen.

The sufficient statistics $\boldsymbol{\mathcal{N}}$ are related to the sufficient statistics set, $(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{V}})$, from the TM algorithm. In the special case where all the variables are dichotomic, both sets of sufficient statistics are the same. However when the variables are multinomial, this relationship is given by linear systems of equations which can be obtained by means of Equation 10. Therefore, using these systems of equations we are able to obtain the values of $\boldsymbol{\mathcal{N}}$ from $\boldsymbol{\mathcal{U}}$ and vice versa.

As an example, we show how one of the linear systems of equations can be obtained from Equation 10. M_0^s, with $s = 1, \ldots, v_0$, are sufficient statistics from the set $\boldsymbol{\mathcal{U}}$ and, as mentioned in Equation 10, $M_0^s = \sum_{d=1}^{N} (C^{(d)})^s$. Since $\sum_{d=1}^{N} (C^{(d)}) = 0 \cdot N_0^0 + \ldots + v_0 \cdot N_0^{v_0}$, the system of equation that relates both $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{N}}$ for the variable C can be written in the matrix form as follows:

$$\underbrace{\begin{pmatrix} 1 & 2 & \ldots & v_0 \\ 1 & 2^2 & \ldots & v_0^2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 2^{v_0} & \ldots & v_0^{v_0} \end{pmatrix}}_{\boldsymbol{COEFFS^*}} \underbrace{\begin{pmatrix} N_0^1 \\ N_0^2 \\ \vdots \\ N_0^{v_0} \end{pmatrix}}_{\boldsymbol{\mathcal{N}^*}} = \underbrace{\begin{pmatrix} M_0^1 \\ M_0^2 \\ \vdots \\ M_0^{v_0} \end{pmatrix}}_{\boldsymbol{\mathcal{U}^*}} \tag{12}$$

Once we have obtained the values of the statistics in $\boldsymbol{\mathcal{N}}$, we are able to calculate $p(C = c)$ and $p(X_i = l|C = c)$ by:

$$p(C = c) = \frac{N_0^c}{N}, \qquad p(X_i = l|C = c) = \frac{N_{0i}^{cl}}{N_0^c}$$

and therefore calculate the value of $E_{\boldsymbol{\theta}}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\}$. Finally, we are able to iterate the algorithm and thus obtain the new value for the statistic $\boldsymbol{\mathcal{U}}$ (see Equation 7). These $p(C = c)$ and $p(X_i = l|C = c)$ are also the parameters $\boldsymbol{\theta}$ of the naïve

Obtain $\boldsymbol{n}_0$ from the dataset
Calculate $\boldsymbol{u}_0$ from $\boldsymbol{n}_0$
while stopping criterion is not met
  Calculate $E_{\boldsymbol{\theta}_r}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\}$
  Update $\boldsymbol{u}$:
    $\boldsymbol{u}_{r+1} = \boldsymbol{u}_r + \boldsymbol{u}_0 - E_{\boldsymbol{\theta}_r}\{\boldsymbol{\mathcal{U}}|\boldsymbol{x}\}$
  Calculate $\boldsymbol{n}_{r+1}$ from $\boldsymbol{u}_{r+1}$
  Calculate $\boldsymbol{\theta}_{r+1}$ from $\boldsymbol{n}_{r+1}$
  if illegal $\boldsymbol{\theta}_{r+1}$ or conditional log-likelihood decreases
    Find the best legal $\boldsymbol{\theta}_{r+1}$ via linear search
  end if
end while

Fig. 1. General pseudocode for the discriminative learning of Bayesian classifiers. Note that $\boldsymbol{n}_r$ and $\boldsymbol{u}_r$ are the values of the statistics in $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\mathcal{U}}$ at iteration r, respectively

Bayes classifier that we are learning. Hence, we have to calculate $\boldsymbol{\mathcal{N}}$ in order to obtain $\boldsymbol{\theta}$. A general pseudo-algorithm for the discriminative learning of Bayesian classifiers is given in Figure 1.

The process of maximizing the conditional log-likelihood with the TM algorithm looks computationally hard because we have to solve several linear systems of equations at each iteration. However, from one iteration of the algorithm to another one, in the systems of equations, only the values in $\boldsymbol{\mathcal{U}}^*$ change (see Equation 12). Therefore, we can obtain the LU transformation of $\boldsymbol{COEFFS}^*$, which is constant throughout the algorithm. Thus, the solution of the systems of equations at each iteration is quite simple. Moreover, the LU transformation is also the same for every problem with the same number of variables and the same number of states per variable. Hence, it may be feasible to calculate these transformations and store the solutions for future use.

3.2 The TM Algorithm for TAN

In this section we introduce the adaptation of the TM algorithm in order to maximize the conditional log-likelihood with TAN models where the variables are assumed to be multinomial. The development of the TM algorithm for TAN models assumes that the structure of the model is already known. Therefore, before performing the discriminative learning of a TAN model, we need to set its structure.

The adaptation of the TM algorithm for TAN is similar to the adaptation for a naïve Bayes model shown above. Hence, we only provide the sufficient statistics that the TM algorithm uses.

In the case of TAN models, we need to differentiate between two kinds of predictive variables: the one which has only one parent, that is, the root of the tree formed by the predictive variables, and the rest of predictive variables, which have two parents: the class and another predictive variable. We assume, without loss of generality, that the root variable is the first one, X_1. If we develop the $l(\boldsymbol{\theta})$ function for a TAN model with multinomial variables in a similar way to Equation 4, we can identify the following set of sufficient statistics $\boldsymbol{\mathcal{U}} = (\boldsymbol{\mathcal{U}}_1, \boldsymbol{\mathcal{U}}_2, \boldsymbol{\mathcal{U}}_3)$ and $\boldsymbol{\mathcal{V}} = (\boldsymbol{\mathcal{V}}_1, \boldsymbol{\mathcal{V}}_2)$, where:

$$\begin{aligned}
\boldsymbol{\mathcal{U}_1} &= (M_0^w | w = 1, 2, \ldots, v_0) \\
\boldsymbol{\mathcal{U}_2} &= (M_{0x_i}^{wt} | w = 1, 2, \ldots, v_0;\ t = 1, 2, \ldots, v_i;\ i = 1, \ldots, n) \\
\boldsymbol{\mathcal{U}_3} &= (M_{0x_i x_{j(i)}}^{wtz} | w = 1, \ldots, v_0;\ t = 1, 2, \ldots, v_i;\ z = 1, 2, \ldots, v_{j(i)};\ i = s+1, \ldots, n) \\
\boldsymbol{\mathcal{V}_1} &= (M_{x_i}^{t} | t = 1, \ldots, v_i) \\
\boldsymbol{\mathcal{V}_2} &= (M_{x_i x_{j(i)}}^{tz} | t = 0, 1, \ldots, v_i;\ z = 0, 1, \ldots, v_{j(i)};\ i = s+1\ \ldots, n)
\end{aligned}$$

with M_c^w, $M_{cx_i}^{wt}$, $M_{cx_i x_{j(i)}}^{wtz}$, $M_{x_i}^t$ and $M_{x_i x_{j(i)}}^{tz}$ defined as follows:

$$M_c^w = \sum_{k=1}^{N} (C^{(k)})^w \quad , \quad M_{cx_i}^{wt} = \sum_{k=1}^{N} (C^{(k)})^w (X_i^{(k)})^t \quad , \quad M_{cx_i x_{j(i)}}^{wtz} = \sum_{k=1}^{N} (C^{(k)})^w (X_i^{(k)})^t (X_{j(i)}^{(k)})^z$$

$$M_{x_i}^t = \sum_{k=1}^{N} (X_i^{(k)})^t \quad , \quad M_{x_i x_{j(i)}}^{tz} = \sum_{k=1}^{N} (X_i^{(k)})^t (X_{j(i)}^{(k)})^z$$

The adaptation of the TM algorithm for TAN models is equal to the one shown in Figure 1 but using the set of sufficient statistics described above.

4 Experimental Results

In this section we present an empirical test which attempts to illustrate the performance of the TM algorithm applied to Bayesian classification models such as naïve Bayes and TAN.

In the case of naïve Bayes models, the structure does not depend on the data, that is, a naïve Bayes structure may only differ from another one in the number of predictive variables. However, the structure of TAN models is learned from the data using the algorithm proposed by [14], which takes the conditional mutual information of two variables given the class into account.

We have evaluated the TM algorithm for the discriminative learning of Bayesian classifiers using sixteen datasets obtained from the UCI repository [17]. Moreover, we use the `Corral` and `Mofn-3-7-10` datasets, which were developed by [18] to evaluate methods for subset selection, and the `Tips` dataset [19]. `Tips` is a medical dataset to identify the subgroup of patients surviving within the first six months after the *transjugular intrahepatic portosystemic shunt* (TIPS) placement, a non-surgical method to avoid portal hypertension.

Table 1. Estimated accuracy obtained in the experiments with naïve Bayes and TAN models

	NB	*NB–TM*	*NB vs. NB–TM*	*TAN*	*TAN–TM*	*TAN vs. TAN–TM*
Australian	85.65± 2.61	**88.41±2.67**	0.112	86.08±2.88	**88.98±3.53**	○ 0.094
Breast	97.37± 1.64	**98.98±0.74**	• 0.036	**97.37±1.64**	95.46±1.41	• 0.016
Chess	87.77± 0.91	**95.15±0.41**	• 0.009	92.40±1.73	**96.81±0.49**	• 0.009
Cleve	83.14± 4.89	**87.53±4.72**	○ 0.072	82.77±1.61	**87.85±3.24**	• 0.043
Corral	86.77± 9.27	**90.61±6.27**	0.197	**100.00±0.00**	99.20±1.60	0.317
Crx	86.68± 4.70	**88.52±1.59**	0.600	86.06±1.33	**89.59±1.56**	○ 0.075
Flare	92.12± 2.16	**95.12±1.21**	• 0.015	95.78±2.79	**96.72±1.23**	0.527
German	75.40± 3.50	**78.90±4.00**	○ 0.059	72.80±2.22	**84.00±0.89**	• 0.009
Glass	74.31± 7.32	**76.18±6.92**	0.344	72.90±2.74	**81.75±3.88**	• 0.045
Heart	83.33± 6.73	**86.67±4.44**	0.390	72.90±2.74	**81.75±3.87**	• 0.036
Hepatitis	85.00±10.15	**93.75±5.56**	0.316	87.50±6.84	**100.00±0.00**	• 0.004
Iris	94.67± 3.40	**95.33±3.40**	0.746	93.33±2.11	**96.00±2.49**	0.142
Lymphography	83.77± 4.97	**91.22±3.49**	0.141	79.08±2.28	**98.98±1.65**	• 0.008
Mofn-3-7-10	86.63± 2.53	**100.00±0.00**	• 0.005	90.86±1.79	**100.00±0.00**	• 0.005
Pima	77.96± 1.31	**79.95±1.47**	0.136	79.17±3.72	**79.82±3.72**	0.136
Soybean-large	96.26± 1.64	**97.51±1.42**	• 0.014	98.58±0.71	**99.29±0.66**	• 0.014
Tips	88.78± 4.65	**100.00±0.00**	• 0.019	89.87±6.20	**100.00±0.00**	• 0.005
Vehicle	61.94± 1.58	**78.61±1.51**	• 0.009	71.63±4.19	**83.46±3.72**	• 0.009
Vote	89.88± 2.45	**98.39±1.17**	• 0.008	93.56±1.55	**99.08±0.86**	• 0.008

The discriminative learning of Bayesian network classifiers described in the paper does not deal with missing data or continuous variables. Therefore, a preprocessing step was needed before using the datasets. On the one hand, every data sample which contained missing data was removed. On the other hand, variables with continuous values were discretized using the method described by [20], which is a variant of the Fayyad and Irani's [21] discretization method. The accuracy of the classifiers is measured by five-fold cross validation and it is based on the percentage of successful predictions. The same pre-processing and validation methodology has been used before in the literature for the generative [14] or discriminative [12] learning of Bayesian network classifiers using all the datasets that have been used in this paper, except for Tips.

The TM algorithm iteratively maximizes the conditional log-likelihood and it stops when a certain criterion is met. In the experiments, the algorithm stops when the difference between the conditional log-likelihood value in two consecutive steps is less than 0.001. On the other hand, as pointed out in Section 2.2, the TM calculations may lead the parameters of the model to illegal values. These situations are solved by applying a linear search where we look for λ in interval $(0, 1)$ with a 0.01 increment (see Equation 8).

Table 1 shows the estimated accuracy for the naïve Bayes (NB) and TAN classifiers learned using both generative and discriminative approaches. The generative approach that we use is the classical learning of Bayesian classifiers using the maximum likelihood parameters. In contrast, the discriminative learning is carried out using the TM algorithm proposed in this paper. In order to com-

Table 2. Conditional log-likelihood values for the experiments with naïve Bayes and TAN models

	NB	*NB–TM*	*TAN*	*TAN–TM*
Australian	−291.71	−190.38	−195.97	−195.97
Breast	−136.74	−22.71	−22.13	−10.41
Chess	−917.76	−478.22	−591.46	−307.04
Cleve	−124.65	−93.24	−96.36	−82.27
Corral	−37.58	−25.17	−10.19	−3.28
Crx	−251.71	−177.90	−173.71	−173.71
Flare	−287.91	−216.18	−152.76	−137.43
German	−488.91	−456.81	−409.05	−378.12
Glass	−141.86	−141.86	−117.60	−116.96
Heart	−112.91	−86.11	−88.21	−73.66
Hepatitis	−23.29	−9.27	−9.12	−2.61
Iris	−21.52	−13.85	−17.90	−16.57
Lymphography	−46.94	−28.06	−25.90	−13.17
Mofn-3-7-10	−269.32	−3.75	−232.03	−44.74
Pima	−361.36	−340.55	−33.24	331.22
Soybean-large	−108.15	−43.27	−22.14	−22.14
Tips	−45.66	−0.12	−2.77	−0.03
Vehicle	−1487.18	−355.30	−360.27	−297.16
Vote	−257.63	−13.66	−49.55	−13.88

pare the estimated accuracy for both discriminative and generative models we perform a Mann-Whitney test [22], whose results are also shown in Table 1. In addition to the Mann-Whitney p-value, we mark with • those experiments where the difference between generative and discriminative models is significant at the 95% level, and with ∘ if it is significant only at the 90% level. The TM algorithm improves the estimated accuracy for naïve Bayes in all the datasets and, in the case of TAN models, only in Breast and Corral does the generative model obtain a higher estimated accuracy. This may be due to a worse performance of discriminative learning when the structure of the classifier is correct [3], that is the structure that perfectly models the relationship between the variables, and TAN is a structure a bit more complex that can model the dataset better. However, even if the estimated accuracy is usually higher in discriminative models, the difference with respect to generative models is not always significant. In most of the cases if the improvement obtained by the discriminative method is not significant at the 95% level, it is because of a high standard deviation. A cause of this high standard deviation may be the small number of folds used in the cross-validation process. For instance, leaving-one-out cross-validation, used to measure the estimated accuracy of a naïve Bayes and a TAN learned from a dataset such as Corral, leads to a decrease in the standard deviation while the estimated accuracy does not change very much. Nevertheless, we have decided to maintain the cross-validation schema in order to agree with the one used by [12]. Thus we have a point of reference for the result obtained in our experiment. Although it is difficult to compare the results of both papers because we do not have all the data need to perform a statistical test, TM-learning, whose results

are reported in this paper, seems to obtain slightly better results than the [12] method in most of the datasets.

On the other hand, the results of Table 1 only measure the goodness of the TM algorithm indirectly. Actually, the aim of the algorithm is to maximize the conditional log-likelihood and not to maximize directly the estimated accuracy of the classifier. In Table 2, the improvement of the conditional log-likelihood score for the discriminative model with respect to the generative one is shown. As described in Sections 2 and 3, the TM algorithm begins with the same parameters obtained by the generative model (that is the maximun likelihood parameters) and, following an iterative process, it modifies these parameters to maximize the conditional log-likelihood. Note that the TM algorithm is able to obtain a model with higher value for the conditional log-likelihood score in all datasets except for the TAN model learned from `Australian`, `Crx` and `Soybean-large`. This is because, in these three cases, the parameters that maximize the unconditional log-likelihood also represent a maximum for the conditional log-likelihood score. This maximum is not necessarily a global maximum but may be a local one because of the possible non-concavity of the conditional log-likelihood score [13]. However, even when generative and discriminative TAN are the same models for `Australian`, `Crx` and `Soybean-large`, the difference between the estimated accuracies is significant at the 90% level. This is because the conditional log-likelihood value reported in Table 2 is obtained from a classifier which has been learned using the whole dataset. On the other hand, for the cross-validation process, whose results are shown in Table 1, we learn the classifiers using only part of the dataset. Therefore, for each fold, the generative and discriminative classifiers are not necessarily the same.

5 Conclusions

Bayesian classifiers are usually generative classifiers, that is, their parameter configuration attempts to maximize the unconditional log-likelihood function. As far as we know, all the techniques for the discriminative learning of Bayesian classification models are generic numerical optimization methods [12, 13]. This paper presents a new statistical approach to the discriminative learning of Bayesian network classifiers by adapting the TM algorithm proposed by [1]. We present a theoretical development of the TM algorithm to be used with naïve Bayes and TAN, therefore providing an efficient discriminative learning of these models. However, the fact that the discriminative learning maximizes the conditional log-likelihood does not necessarily lead to a better performance of these kind of classifiers. It depends on the dataset and the classifier selected to model this dataset. This idea has been also shown, for example, by [4] and it is reasserted by the results from the experiments that we include in Section 4.

Discriminative learning with the TM algorithm, as it has been presented in this paper, can only be used in supervised classification problems with no missing values, but it can be extended to deal with missing values an with other problems such as unsupervised classification by using a hybrid of the TM and EM algorithms. On the other hand, the same idea can be extended to structural

learning, that is, searching in the space of structures and parameters in order to find the model which maximizes the conditional log-likelihood function.

Acknowledgments

This work was supported in part by the Spanish Ministerio de Ciencia y Tecnología under TIC2001-2973-C05-03 grant, by the ETORTEK-BIOLAN SAIOTEK S-PE04UN25 projects of the Basque Government, by the Navarra Goverment under PhD grant, and by the University of the Basque Country under grant 9/UPV 00140.226-15334/2003.

The authors thank the Clínica Universitaria de Navarra, Spain, for providing the Tips dataset.

References

1. Edwards, D., Lauritzen, S.L.: The TM algorithm for maximising a conditional likelihood function. Biometrika **88** (2001) 961–972
2. Dawid, A.P.: Properties of diagnostic data distributions. Biometrics **32** (1976) 647–658
3. Rubinstein, Y.D., Hastie, T.: Discriminative vs. informative learning. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. (1997) 49–53
4. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes. In: Proceedings of the Sixteenth Advances in Neural Information Processing Systems 14. (2002)
5. Jebara, T.: Machine Learning: Discriminative and Generative. Kluwer Academic Publishers (2003)
6. Fisher, R.A.: The use of multiple measurement. Annals of Eugenics **7** (1936) 179–188
7. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley and Sons (1973)
8. Hosmer, D., Lemeshow, S.: Applied Logistic Regression. John Wiley and Sons (1989)
9. Bishop, C.: Neural Networks for Pattern Recognition. Oxford Press (1996)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann (1988)
11. Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall (2003)
12. Greiner, R., Zhou, W., Su, X., Shen, B.: Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. Machine Learning (2004) Accepted for publication.
13. Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., Tirri, H.: On discriminative Bayesian network classifiers and logistic regression. Machine Learning (2004) Accepted for publication.
14. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29** (1997) 131–164
15. Sundberg, R.: The convergence rate of the TM algorithm of Edwards and Lauritzen. Biometrika **89** (2002) 478–483

16. Santafé, G., Lozano, J.A., Larrañaga, P.: El algoritmo TM para clasificadores Bayesianos (in Spanish). Technical Report EHU-KZAA-IK-2/04, University of the Basque Country (2004)
17. Blake, C., Merz, C.: UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn (1998)
18. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273–324
19. Inza, I., Merino, M., Larrañaga, P., Quiroga, J., Sierra, B., Girala, M.: Feature subset selection by genetic algorithms and estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with TIPS. Artificial Intelligence in Medicine **23** (2001) 187–205
20. Dougherty, J.R., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Proceedings of the Twelfth International Conference on Machine Learning. (1995) 194–202
21. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence. (1993) 1022–1027
22. Mann, H., Whitney, D.: On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistics **18** (1947) 50–60