Learning the Tree Augmented Naive Bayes Classifier from incomplete datasets

Olivier C.H. François and Philippe Leray
LITIS Lab., INSA de Rouen, BP 08, av. de l'Université
76801 Saint-Etienne-Du-Rouvray, France.

Abstract

The Bayesian network formalism is becoming increasingly popular in many areas such as decision aid or diagnosis, in particular thanks to its inference capabilities, even when data are incomplete. For classification tasks, Naive Bayes and Augmented Naive Bayes classifiers have shown excellent performances. Learning a Naive Bayes classifier from incomplete datasets is not difficult as only parameter learning has to be performed. But there are not many methods to efficiently learn Tree Augmented Naive Bayes classifiers from incomplete datasets. In this paper, we take up the structural EM algorithm principle introduced by (Friedman, 1997) to propose an algorithm to answer this question.

1 Introduction

Bayesian networks are a formalism for probabilistic reasoning increasingly used in decision aid, diagnosis and complex systems control. Let $\mathbb{X} = \{X_1, \ldots, X_n\}$ be a set of discrete random variables. A Bayesian network $\mathcal{B} =< \mathcal{G}, \Theta >$ is defined by a directed acyclic graph $\mathcal{G} =< \mathbb{N}, \mathbb{U} >$ where $\mathbb{N}$ represents the set of nodes (one node for each variable) and $\mathbb{U}$ the set of edges, and parameters $\Theta = \{\theta_{ijk}\}_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant q_i, 1 \leqslant k \leqslant r_i}$the set of conditional probability tables of each node X_i knowing its parents' state Pi (with r_i and q_i as respective cardinalities of X_i and P_i).

If $\mathcal{G}$ and Θ are known, many inference algorithms can be used to compute the probability of any variable that has not been measured conditionally to the values of measured variables. Bayesian networks are therefore a tool of choice for reasoning in uncertainty, based on incomplete data, which is often the case in real applications.

It is possible to use this formalism for classification tasks. For instance, the Naive Bayes classifier has shown excellent performance. This model is simple and only need parameter learning that can be performed with incomplete datasets. Augmented Naive Bayes classifiers (with trees, forests or Bayesian networks), which often give better performances than the Naive Bayes classifier, require structure learning. Only a few methods of structural learning deal with incomplete data.

We introduce in this paper a method to learn Tree Augmented Naive Bayes (TAN) classifiers based on the expectation-maximization (EM) principle. Some previous work by (Cohen et al., 2004) also deals with TAN classifiers and EM principle for partially unlabeled data. In there work, only the variable corresponding to the class can be partially missing whereas any variable can be partially missing in the approach we propose here.

We will therefore first recall the issues relating to structural learning, and review the various ways of dealing with incomplete data, primarily for parameter estimation, and also for structure determination. We will then examine the structural EM algorithm principle, before proposing and testing a few ideas for improvement based on the extension of the Maximum Weight Spanning Tree algorithm to deal with incomplete data. Then, we will show how to use the introduced method to learn the well-known Tree Augmented Naive Bayes classifier from incomplete datasets and we will give some experiments on real data.

2 Preliminary remarks

2.1 Structural learning

Because of the super-exponential size of the search space, exhaustive search for the best structure is impossible. Many heuristic methods have been proposed to determine the structure of a Bayesian network. Some of them rely on human expert knowledge, others use real data which need to be, most of the time, completely observed.

Here, we are more specifically interested in score-based methods. Primarily, greedy search algorithm adapted by (Chickering et al., 1995) and maximum weight spanning tree (MWST) proposed by (Chow and Liu, 1968) and applied to Bayesian networks in (Heckerman et al., 1995). The greedy search carried out in directed acyclic graph (DAG) space where the interest of each structure located near the current structure is assessed by means of a BIC/MDL type measurement (Eqn.1)[1] or a Bayesian score like BDe (Heckerman et al., 1995).

$$BIC(\mathcal{G},\Theta) = \log P(\mathcal{D}|\mathcal{G},\Theta) - \frac{\log N}{2}\mathrm{Dim}(\mathcal{G}) \quad (1)$$

where $\mathrm{Dim}(\mathcal{G})$ is the number of parameters used for the Bayesian network representation and N is the size of the dataset $\mathcal{D}$.

The BIC score is decomposable. It can be written as the sum of the local score computed for each node as $BIC(\mathcal{G},\Theta) = \sum_i bic(X_i, P_i, \Theta_{X_i|P_i})$ where $bic(X_i, P_i, \Theta_{X_i|P_i}) =$

$$\sum_{X_i=x_k} \sum_{P_i=pa_j} N_{ijk} \log \theta_{ijk} - \frac{\log N}{2}\mathrm{Dim}(\Theta_{X_i|P_i}) \quad (2)$$

with N_{ijk} the occurrence number of $\{X_i = x_k$ and $P_i = pa_j\}$ in $\mathcal{D}$.

The principle of the MWST algorithm is rather different. This algorithm determines the best tree that links all the variables, using a mutual information measurement like in (Chow and Liu, 1968) or the BIC score variation when two variables become linked as proposed by (Heckerman et al., 1995). The aim is to find an optimal solution, but in a space limited to trees.

[1] As (Friedman, 1997), we consider that the BIC/MDL score is a function of the graph $\mathcal{G}$ and the parameters Θ, generalizing the classical definition of the BIC score which is defined with our notation by $BIC(\mathcal{G},\Theta^*)$ where Θ^* is obtained by maximizing the likelihood or $BIC(\mathcal{G},\Theta)$ score for a given $\mathcal{G}$.

2.2 Bayesian classifiers

Bayesian classifiers as Naive Bayes have shown excellent performances on many datasets. Even if the Naive Bayes classifier has underlying heavy independence assumptions, (Domingos and Pazzani, 1997) have shown that it is optimal for conjunctive and disjunctive concepts. They have also shown that the Naive Bayes classifier does not require attribute independence to be optimal under Zero-One loss.

Augmented Naive Bayes classifier appear as a natural extension to the Naive Bayes classifier. It allows to relax the assumption of independence of attributes given the class variable. Many ways to find the best tree to augment the Naive Bayes classifier have been studied. These Tree Augmented Naive Bayes classifiers (Geiger, 1992; Friedman et al., 1997) are a restricted family of Bayesian Networks in which the class variable has no parent and each other attribute has as parents the class variable and at most one other attribute. The BIC score of such a Bayesian network is given by Eqn.3.

$$BIC(\mathcal{T}_{AN},\Theta) = bic(C,\emptyset,\Theta_{C|\emptyset}) + \sum_i bic(X_i,\{C,P_i\},\Theta_{X_i|\{C,P_i\}}) \quad (3)$$

where C stands for the class node and P_i could only be the emptyset $\emptyset$ or a singleton $\{X_j\}$, $X_j \notin \{C, X_i\}$.

Forest Augmented Naive Bayes classifier (FAN) is very close to the TAN one. In this model, the augmented structure is not a tree, but a set of disconnected trees in the attribute space (Sacha, 1999).

2.3 Dealing with incomplete data

2.3.1 Practical issue

Nowadays, more and more datasets are available, and most of them are incomplete. When we want to build a model from an incomplete dataset, it is often possible to consider only the complete samples in the dataset. But, in this case, we do not have a lot of data to learn the model. For instance, if we have a dataset with 2000 samples on 20 attributes with a probability of 20% that a data is missing, then, only

23 samples (in average) are complete. Generalizing from the example, we see that we cannot ignore the problem of incomplete datasets.

2.3.2 Nature of missing data

Let $\mathcal{D} = \{X_i^l\}_{1 \leqslant i \leqslant n, 1 \leqslant l \leqslant N}$ our dataset, with $\mathcal{D}_o$ the observed part of $\mathcal{D}$, $\mathcal{D}_m$ the missing part and $\mathcal{D}_{co}$ the set of completely observed cases in $\mathcal{D}_o$. Let also $\mathcal{M} = \{M_{il}\}$ with $M_{il} = 1$ if X_i^l is missing, 0 if not. We then have the following relations:

$$\begin{aligned}
\mathcal{D}_m &= \{X_i^l \ / \ M_{il} = 1\}_{1 \leqslant i \leqslant n, 1 \leqslant l \leqslant N} \\
\mathcal{D}_o &= \{X_i^l \ / \ M_{il} = 0\}_{1 \leqslant i \leqslant n, 1 \leqslant l \leqslant N} \\
\mathcal{D}_{co} &= \{[X_1^l \dots X_n^l] \ / \ [M_{1l} \dots M_{nl}] = [0 \dots 0]\}_{1 \leqslant l \leqslant N}
\end{aligned}$$

Dealing with missing data depends on their nature. (Rubin, 1976) identified several types of missing data:

- MCAR (Missing Completly At Random): $P(\mathcal{M}|\mathcal{D}) = P(\mathcal{M})$, the probability for data to be missing does not depend on $\mathcal{D}$,
- MAR (Missing At Random): $P(\mathcal{M}|\mathcal{D}) = P(\mathcal{M}|\mathcal{D}_o)$, the probability for data to be missing depends on observed data,
- NMAR (Not Missing At Random): the probability for data to be missing depends on both observed and missing data.

MCAR and MAR situations are the easiest to solve as observed data include all necessary information to estimate missing data distribution. The case of NMAR is trickier as outside information has to be used to model the missing data distribution.

2.3.3 Learning Θ with incomplete data

With MCAR data, the first and simplest possible approach is the *complete case analysis*. This is a parameter estimation based on $\mathcal{D}_{co}$, the set of completely observed cases in $\mathcal{D}_o$. When $\mathcal{D}$ is MCAR, the estimator based on $\mathcal{D}_{co}$ is unbiased. However, with a high number of variables the probability for a case $[X_1^l \dots X_n^l]$ to be completely measured is low and $\mathcal{D}_{co}$ may be empty.

One advantage of Bayesian networks is that, if only X_i and $P_i = Pa(X_i)$ are measured, then the corresponding conditional probability table can be estimated. Another possible method with MCAR cases is the *available case analysis*, *i.e.* using for the estimation of each conditional probability $P(X_i|Pa(X_i))$ the cases in $\mathcal{D}_o$ where X_i and $Pa(X_i)$ are measured, not only in $\mathcal{D}_{co}$ (where all X_i's are measured) as in the previous approach.

Many methods try to rely more on all the observed data. Among them are *sequential updating* (Spiegelhalter and Lauritzen, 1990), *Gibbs sampling* (Geman and Geman, 1984), and *expectation maximisation* (EM) in (Dempster et al., 1977). Those algorithms use the missing data MAR properties. More recently, *bound and collapse* algorithm (Ramoni and Sebastiani, 1998) and *robust Bayesian estimator* (Ramoni and Sebastiani, 2000) try to resolve this task whatever the nature of missing data.

EM has been adapted by (Lauritzen, 1995) to Bayesian network parameter learning when the structure is known. Let $\log P(\mathcal{D}|\Theta) = \log P(\mathcal{D}_o, \mathcal{D}_m|\Theta)$ be the data log-likelihood. $\mathcal{D}_m$ being an unmeasured random variable, this log-likelihood is also a random variable function of $\mathcal{D}_m$. By establishing a reference model Θ^*, it is possible to estimate the probability density of the missing data $P(\mathcal{D}_m|\Theta^*)$ and therefore to calculate $Q(\Theta : \Theta^*)$, the expectation of the previous log-likelihood:

$$Q(\Theta : \Theta^*) = E_{\Theta^*}\left[\log P(\mathcal{D}_o, \mathcal{D}_m|\Theta)\right] \tag{4}$$

So $Q(\Theta : \Theta^*)$ is the expectation of the likelihood of any set of parameters Θ calculated using a distribution of the missing data $P(\mathcal{D}_m|\Theta^*)$. This equation can be re-written as follows.

$$Q(\Theta : \Theta^*) = \sum_{i=1}^{n} \sum_{X_i = x_k} \sum_{P_i = pa_j} N_{ijk}^* \log \theta_{ijk} \tag{5}$$

where $N_{ijk}^* = E_{\Theta^*}[N_{ijk}] = N \times P(X_i = x_k, P_i = pa_j|\Theta^*)$is obtained by inference in the network $< \mathcal{G}, \Theta^* >$ if the $\{X_i, P_i\}$ are not completely measured, or else only by mere counting.

(Dempster et al., 1977) proved convergence of the EM algorithm, as the fact that it was not necessary to find the global optimum Θ^{i+1} of function $Q(\Theta : \Theta^i)$ but simply a value which would increase function Q (*Generalized* EM).

2.3.4 Learning $\mathcal{G}$ with incomplete dataset

The main methods for structural learning with incomplete data use the EM principle: *Alternative Model Selection* EM (AMS-EM) pro-

posed by (Friedman, 1997) or *Bayesian Structural* EM (BS-EM) (Friedman, 1998). We can also cite the *Hybrid Independence Test* proposed in (Dash and Druzdzel, 2003) that can use EM to estimate the essential sufficient statistics that are then used for an independence test in a constraint-based method. (Myers et al., 1999) also proposes a structural learning method based on genetic algorithm and MCMC. We will now explain the structural EM algorithm principle in details and see how we could adapt it to learn a TAN model.

3 Structural em algorithm

3.1 General principle

The EM principle, which we have described above for parameter learning, applies more generally to structural learning (Algorithm 1 as proposed by (Friedman, 1997; Friedman, 1998)).

Algorithm 1 : Generic EM for structural learning

1: **Init:** $i = 0$
Random or heuristic choice of the initial Bayesian network $(\mathcal{G}^0, \Theta^0)$
2: **repeat**
3: $i = i + 1$
4: $(\mathcal{G}^i, \Theta^i) = \underset{\mathcal{G},\Theta}{\text{argmax}} \quad Q(\mathcal{G}, \Theta : \mathcal{G}^{i-1}, \Theta^{i-1})$
5: **until** $|Q(\mathcal{G}^i, \Theta^i : \mathcal{G}^{i-1}, \Theta^{i-1}) - Q(\mathcal{G}^{i-1}, \Theta^{i-1} : \mathcal{G}^{i-1}, \Theta^{i-1})| \leqslant \epsilon$

The maximization step in this algorithm (step 4) has to be performed in the joint space $\{\mathcal{G}, \Theta\}$ which amounts to searching the best structure and the best parameters corresponding to this structure. In practice, these two steps are clearly distinct[2]:

$$\mathcal{G}^i = \underset{\mathcal{G}}{\text{argmax}} \quad Q(\mathcal{G}, \bullet : \mathcal{G}^{i-1}, \Theta^{i-1}) \tag{6}$$

$$\Theta^i = \underset{\Theta}{\text{argmax}} \quad Q(\mathcal{G}^i, \Theta : \mathcal{G}^{i-1}, \Theta^{i-1}) \tag{7}$$

where $Q(\mathcal{G}, \Theta : \mathcal{G}^*, \Theta^*)$ is the expectation of the likelihood of any Bayesian network $< \mathcal{G}, \Theta >$ computed using a distribution of the missing data $P(\mathcal{D}_m | \mathcal{G}^*, \Theta^*)$.

Note that the first search (Eqn.6) in the space of possible graphs takes us back to the initial problem, i.e. the search for the best structure in a super-exponential space. However, with *Generalised* EM it is sufficient to look for a better solution rather than the best possible one, without affecting the algorithm convergence properties. This search for a better solution can then be done in a limited space, like for example $\mathcal{V}_{\mathcal{G}}$, the set of the neigbours of graph $\mathcal{G}$ that have been generated by removal, addition or inversion of an arc.

Algorithm 2 : Detailed EM for structural learning

1: **Init:** $finished = false$, $i = 0$
Random or heuristic choice of the initial Bayesian network $(\mathcal{G}^0, \Theta^{0,0})$
2: **repeat**
3: $j = 0$
4: **repeat**
5: $\Theta^{i,j+1} = \underset{\Theta}{\text{argmax}} \quad Q(\mathcal{G}^i, \Theta : \mathcal{G}^i, \Theta^{i,j})$
6: $j = j + 1$
7: **until** convergence $(\Theta^{i,j} \rightarrow \Theta^{i,j^o})$
8: **if** $i = 0$ or $|Q(\mathcal{G}^i, \Theta^{i,j^o} : \mathcal{G}^{i-1}, \Theta^{i-1,j^o}) - Q(\mathcal{G}^{i-1}, \Theta^{i-1,j^o} : \mathcal{G}^{i-1}, \Theta^{i-1,j^o})| > \epsilon$ **then**
9: $\mathcal{G}^{i+1} = \arg\max_{\mathcal{G} \in \mathcal{V}_{\mathcal{G}^i}} \quad Q(\mathcal{G}, \bullet : \mathcal{G}^i, \Theta^{i,j^o})$
10: $\Theta^{i+1,0} = \underset{\Theta}{\text{argmax}} \quad Q(\mathcal{G}^{i+1}, \Theta : \mathcal{G}^i, \Theta^{i,j^o})$
11: $i = i + 1$
12: **else**
13: $finished = true$
14: **end if**
15: **until** $finished$

Concerning the search in the space of the parameters (Eqn.7), (Friedman, 1997) proposes repeating the operation several times, using a clever initialisation. This step then amounts to running the parametric EM algorithm for each structure $\mathcal{G}^i$, starting with structure $\mathcal{G}^0$ (steps 4 to 7 of Algorithm 2). The two structural EM algorithms proposed by Friedman can therefore be considered as greedy search algorithms, with EM parameter learning at each iteration.

3.2 Choice of function Q

We now have to choose the function Q that will be used for structural learning. The likelihood used for parameter learning is not a good indicator to determine the best graph since it gives more importance to strongly connected structures. Moreover, it is impossible to compute marginal likelihood when data are incomplete, so that it is necessary to rely on an efficient approximation like those reviewed by (Chicker-

[2]The notation $Q(\mathcal{G}, \bullet : \dots)$ used in Eqn.6 stands for $E_\Theta[Q(\mathcal{G}, \Theta : \dots)]$ for Bayesian scores or $Q(\mathcal{G}, \Theta^o : \dots)$ where Θ^o is obtained by likelihood maximisation.

ing and Heckerman, 1996). In complete data cases, the most frequently used measurements are the BIC/MDL score and the Bayesian BDe score (see paragraph 2.1). When proposing the MS-EM and MWST-EM algorithms, (Friedman, 1997) shows how to use the BIC/MDL score with incomplete data, by applying the principle of Eqn.4 to the BIC score (Eqn.1) instead of likelihood. Function Q^{BIC} is defined as the BIC score expectation by using a certain probability density on the missing data $P(\mathcal{D}_m|\mathcal{G}^*,\Theta^*)$:

$$Q^{BIC}(\mathcal{G},\Theta:\mathcal{G}^*,\Theta^*) = \quad (8)$$
$$E_{\mathcal{G}^*,\Theta^*}\left[\log P(\mathcal{D}_o,\mathcal{D}_m|\mathcal{G},\Theta)\right] - \frac{1}{2}\mathrm{Dim}(\mathcal{G})\log N$$

As the BIC score is decomposable, so is Q^{BIC}.

$$Q^{BIC}(\mathcal{G},\Theta:\mathcal{G}^*,\Theta^*) = \sum_i Q^{bic}(X_i,P_i,\Theta_{X_i|P_i}:\mathcal{G}^*,\Theta^*)$$

where $\quad Q^{bic}(X_i,P_i,\Theta_{X_i|P_i}:\mathcal{G}^*,\Theta^*) = \quad (9)$

$$\sum_{X_i=x_k}\sum_{P_i=pa_j} N^*_{ijk}\log\theta_{ijk} - \frac{\log N}{2}\mathrm{Dim}(\Theta_{X_i|P_i}) \quad (10)$$

with $N^*_{ijk} = E_{\mathcal{G}^*,\Theta^*}[N_{ijk}] = N * P(X_i = x_k, P_i = pa_j|\mathcal{G}^*,\Theta^*)$ obtained by inference in the network $\{\mathcal{G}^*,\Theta^*\}$ if $\{X_i,P_i\}$ are not completely measured, or else only by mere counting. With the same reasoning, (Friedman, 1998) proposes the adaptation of the BDe score to incomplete data.

4 TAN-EM, a structural EM for classification

(Leray and François, 2005) have introduced MWST-EM an adaptation of MWST dealing with incomplete datasets. The approach we propose here is using the same principles in order to efficiently learn TAN classifiers from incomplete datasets.

4.1 MWST-EM, a structural EM in the space of trees

Step 1 of Algorithm 2, like all the previous algorithms, deals with the choice of the initial structure. The choice of an oriented chain graph linking all the variables proposed by (Friedman, 1997) seems even more judicious here, since this chain graph also belongs to the tree space. Steps 4 to 7 do not change. They deal with the running of the parametric EM algorithm for each structure $\mathcal{B}^i$, starting with structure $\mathcal{B}^0$.

There is a change from the regular structural EM algorithm in step 9, i.e. the search for a better structure for the next iteration. With the previous structural EM algorithms, we were looking for the best DAG among the neighbours of the current graph. With MWST-EM, we can directly get the best tree that maximises function Q.

In paragraph 2.1, we briefly recalled that the MWST algorithm used a similarity function between two nodes which was based on the BIC score variation whether X_j is linked to X_i or not. This function can be summed up in the following (symmetrical) matrix:

$$\left[M_{ij}\right]_{1\leqslant i,j\leqslant n} = \left[bic(X_i,X_j,\Theta_{X_i|X_j}) - bic(X_i,\emptyset,\Theta_{X_i})\right] \quad (11)$$

where the local *bic* score is defined in Eqn.2.

Running maximum (weight) spanning algorithms like Kruskal's on matrix M enables us to obtain the best tree $\mathcal{T}$ that maximises the sum of the local scores on all the nodes, i.e. function BIC of Eqn.2.

By applying the principle we described in section 3.2, we can then adapt MWST to incomplete data by replacing the local *bic* score of Equn.11 with its expectation; to do so, we use a certain probability density of the missing data $P(\mathcal{D}_m|\mathcal{T}^*,\Theta^*)$:

$$\left[M^Q_{ij}\right]_{i,j} = \Big[Q^{bic}(X_i,P_i=\{X_j\},\Theta_{X_i|X_j}:\mathcal{T}^*,\Theta^*) - Q^{bic}(X_i,P_i=\emptyset,\Theta_{X_i}:\mathcal{T}^*,\Theta^*)\Big] \quad (12)$$

With the same reasoning, running a maximum (weight) spanning tree algorithm on matrix M^Q enables us to get the best tree $\mathcal{T}$ that maximises the sum of the local scores on all the nodes, i.e. function Q^{BIC} of Eqn.9.

4.2 TAN-EM, a structural EM for classification

The score used to find the best TAN structure is very similar to the one used in MWST, so we can adapt it to incomplete datasets by defining the following score matrix:

$$\left[M^Q_{ij}\right]_{i,j} = \Big[Q^{bic}(X_i,P_i=\{C,X_j\},\Theta_{X_i|X_jC}:\mathcal{T}^*,\Theta^*) - Q^{bic}(X_i,P_i=\{C\},\Theta_{X_i|C}:\mathcal{T}^*,\Theta^*)\Big] \quad (13)$$

Using this new score matrix, we can use the approach previously proposed for MWST-EM to

get the best augmented tree, and connect the class node to all the other nodes to obtain the TAN structure. We are currently using the same reasoning to find the best forest "extension".

4.3 Related works

(Meila-Predoviciu, 1999) applies MWST algorithm and EM principle, but in another framework, learning mixtures of trees. In this work, the data is complete, but a new variable is introduced in order to take into account the weight of each tree in the mixture. This variable isn't measured so EM is used to determine the corresponding parameters.

(Peña et al., 2002) propose a change inside the framework of the SEM algorithm resulting in an alternative approach for learning Bayes Nets for clustering more efficiently.

(Greiner and Zhou, 2002) propose maximizing conditional likelihood for BN parameter learning. They apply their method to MCAR incomplete data by using *available case analysis* in order to find the best TAN classifier.

(Cohen et al., 2004) deal with TAN classifiers and EM principle for partially unlabeled data. In there work, only the variable corresponding to the class can be partially missing whereas any variable can be partially missing in our TAN-EM extension.

5 Experiments

5.1 Protocol

The experiment stage aims at evaluating the Tree Augmented Naive Bayes classifier on incomplete datasets from UCI repository[3]: `Hepatitis`, `Horse`, `House`, `Mushrooms` and `Thyroid`.

The TAN-EM method we proposed here is compared to the Naive Bayes classifier with EM parameters learning. We also indicate the classification rate obtained by three methods: MWST-EM, SEM initialised with a random chain and SEM initialised with the tree given by MWST-EM (SEM+T). The first two methods are dedicated to classification tasks while the others do not consider the class node as a specific variable.

We also give an α confidence interval for each classification rate, based on Eqn.14 proposed by (Bennani and Bossaert, 1996):

$$I(\alpha, N) = \frac{T + \frac{Z_\alpha^2}{2N} \pm Z_\alpha \sqrt{\frac{T(1-T)}{N} + \frac{Z_\alpha^2}{4N^2}}}{1 + \frac{Z_\alpha^2}{N}} \qquad (14)$$

where N is the number of samples in the dataset, T is the classification rate and $Z_\alpha = 1,96$ for $\alpha = 95\%$.

5.2 Results

The results are summed up in Table 1. First, we could see that even if the Naive Bayes classifier often gives good results, the other tested methods allow to obtain better classification rates. But, where all runnings of NB-EM give the same results, as EM parameter learning only needs an initialisiation, the other methods do not always give the same results, and then, the same classification rates. We have also noticed (not reported here) that, excepting NB-EM, TAN-EM seems the most stable method concerning the evaluated classification rate while MWST-EM seems to be the less stable.

The method MWST-EM can obtain very good structures with a good initialisation. Then, initialising it with the results of MWST-EM gives us stabler results (see (Leray and François, 2005) for a more specific study of this point).

In our tests, except for this `house` dataset, TAN-EM always obtains a structure that lead to better classification rates in comparison with the other structure learning methods.

Surprisingly, we also remark that MWST-EM can give good classification rates even if the class node is connected to a maximum of two other attributes.

Regarding the log-likelihood reported in Table 1, we see that the TAN-EM algorithm finds structures that can also lead to a good approximation of the underlying probability distribution of the data, even with a strong constraint on the graph structure.

Finally, the Table 1 illustrates that TAN-EM and MWST-EM have about the same complexity (regarding the computational time) and

[3] `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Datasets	N	learn	test	#C	%I	NB-EM	MWST-EM	TAN-EM	SEM	SEM+T
Hepatitis	20	90	65	2	8.4	70.8 [58.8;80.5]	73.8 [62.0;83.0]	**75.4** [63.6;84.2]	66.1 [54.0;76.5]	66.1 [54.0;76.5]
						-1224.2 ; 29.5	**-1147.6** ; 90.4	**-1148.7** ; 88.5	-1211.5 ; 1213.1	-1207.9 ; 1478.5
Horse	28	300	300	2	88.0	75 [63.5;83.8]	77.9 [66.7;86.2]	**80.9** [69.9;88.5]	66.2 [54.3;76.3]	66.2 [54.3;76.3]
						-5589.1 ; 227.7	**-5199.6** ; 656.1	-5354.4 ; 582.2	-5348.3 ; 31807	-5318.2 ; 10054
House	17	290	145	2	46.7	89.7 [83.6;93.7]	**93.8** [88.6;96.7]	92.4 [86.9;95.8]	92.4 [86.9;95.8]	**93.8** [88.6;96.7]
						-2203.4 ; 110.3	-2518.0 ; 157.0	**-2022.2** ; 180.7	-2524.4 ; 1732.4	-2195.8 ; 3327.2
Mushrooms	23	5416	2708	2	30.5	**92.8** [91.7;93.8]	74.7 [73.0;73.4]	91.3 [90.2;92.4]	74.9 [73.2;76.5]	74.9 [73.2;76.5]
						-97854 ; 2028.9	-108011 ; 6228.2	**-87556** ; 5987.4	-111484 ; 70494	-110828 ; 59795
Thyroid	22	2800	972	2	29.9	95.3 [93.7;96.5]	93.8 [92.1;95.2]	**96.2** [94.7;97.3]	93.8 [92.1;95.2]	93.8 [92.1;95.2]
						-39348 ; 1305.6	-38881 ; 3173.0	**-38350** ; 3471.4	**-38303** ; 17197	-39749 ; 14482

Table 1: First line: best classification rate (on 10 runs, except Mushrooms on 5, in %) on test dataset and its confidence interval, for the following learning algorithms: NB-EM, MWST-EM, SEM, TAN-EM and SEM+T. Second line: log-likelihood estimated with test data and calculation time (sec) for the network with the best classification rate. The first six columns give us the name of the dataset and some of its properties : number of attributes, learning sample size, test sample size, number of classes and percentage of incomplete samples.

are a good compromise between NB-EM (classical Naive Bayes with EM parameter learning) and MWST-EM (greedy search with incomplete data).

6 Conclusions and prospects

Bayesian networks are a tool of choice for reasoning in uncertainty, with incomplete data. However, most of the time, Bayesian network structural learning only deal with complete data. We have proposed here an adaptation of the learning process of Tree Augmented Naive Bayes classifier from incomplete datasets (and not only partially labelled data). This method has been successfuly tested on some datasets.

We have seen that TAN-EM was a good classification tool compared to other Bayesian networks we could obtained with structural EM like learning methods.

Our method can easily be extended to unsupervised classification tasks by adding a new step in order to determine the best cardinality for the class variable.

Related future works are the adaptation of some other Augmented Naive Bayes classifiers for incomplete datasets (FAN for instance), but also the study of these methods with MAR datasets.

MWST-EM, TAN-EM and SEM methods are respective adaptations of MWST, TAN and greedy search to incomplete data. These algorithms are applying in (subspace of) DAG space. (Chickering and Meek, 2002) proposed an optimal search algorithm (GES) which deals with Markov equivalent space. Logically enough, the next step in our research is to adapt GES to incomplete datasets. Then we could test results of this method on classification tasks.

7 Acknowledgement

This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

Y. Bennani and F. Bossaert. 1996. Predictive neural networks for traffic disturbance detection in the telephone network. In *Proceedings of IMACS-CESA'96*, page xx, Lille, France.

D. Chickering and D. Heckerman. 1996. Efficient Approximation for the Marginal Likelihood of Incomplete Data given a Bayesian Network. In *UAI'96*, pages 158–168. Morgan Kaufmann.

D. Chickering and C. Meek. 2002. Finding optimal bayesian networks. In Adnan Darwiche and Nir Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 94–102, S.F., Cal. Morgan Kaufmann Publishers.

D. Chickering, D. Geiger, and D. Heckerman. 1995. Learning bayesian networks: Search methods and experimental results. In *Proceedings of Fifth Conference on Artificial Intelligence and Statistics*, pages 112–128.

C.K. Chow and C.N. Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467.

I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang. 2004. Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1553–1568.

D. Dash and M.J. Druzdzel. 2003. Robust independence testing for constraint-based learning of causal structure. In *Proceedings of The Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*, pages 167–174.

A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39:1–38.

P. Domingos and M. Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130.

N. Friedman, D. Geiger, and M. Goldszmidt. 1997. bayesian network classifiers. *Machine Learning*, 29(2-3):131–163.

N. Friedman. 1997. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the 14th International Conference on Machine Learning*, pages 125–133. Morgan Kaufmann.

N. Friedman. 1998. The bayesian structural EM algorithm. In Gregory F. Cooper and Serafín Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 129–138, San Francisco, July. Morgan Kaufmann.

D. Geiger. 1992. An entropy-based learning algorithm of bayesian conditional trees. In *Uncertainty in Artificial Intelligence: Proceedings of the Eighth Conference (UAI-1992)*, pages 92–97, San Mateo, CA. Morgan Kaufmann Publishers.

S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November.

R. Greiner and W. Zhou. 2002. Structural extension to logistic regression. In *Proceedings of the Eighteenth Annual National Conference on Artificial Intelligence (AAI02)*, pages 167–173, Edmonton, Canada.

D. Heckerman, D. Geiger, and M. Chickering. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243.

S. Lauritzen. 1995. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201.

P. Leray and O. François. 2005. bayesian network structural learning and incomplete data. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005), Espoo, Finland*, pages 33–40.

M. Meila-Predoviciu. 1999. *Learning with Mixtures of Trees*. Ph.D. thesis, MIT.

J.W. Myers, K.B. Laskey, and T.S. Lewitt. 1999. Learning bayesian network from incomplete data with stochatic search algorithms. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI99)*.

J.M. Peña, J. Lozano, and P. Larrañaga. 2002. Learning recursive bayesian multinets for data clustering by means of constructive induction. *Machine Learning*, 47:1:63–90.

M. Ramoni and P. Sebastiani. 1998. Parameter estimation in Bayesian networks from incomplete databases. *Intelligent Data Analysis*, 2:139–160.

M. Ramoni and P. Sebastiani. 2000. Robust learning with missing data. *Machine Learning*, 45:147–170.

D.B. Rubin. 1976. Inference and missing data. *Biometrika*, 63:581–592.

J.P. Sacha. 1999. *New Synthesis of bayesian Network Classifiers and Cardiac SPECT Image Interpretation*. Ph.D. thesis, The University of Toledo.

D. J. Spiegelhalter and S. L. Lauritzen. 1990. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605.