Bayesian Model Averaging of TAN Models for Clustering

Guzmán Santafé, Jose A. Lozano and Pedro Larrañaga
Intelligent Systems Group, Computer Science and A. I. Dept.
University of the Basque Country, Spain

Abstract

Selecting a single model for clustering ignores the uncertainty left by finite data as to which is the correct model to describe the dataset. In fact, the fewer samples the dataset has, the higher the uncertainty is in model selection. In these cases, a Bayesian approach may be beneficial, but unfortunately this approach is usually computationally intractable and only approximations are feasible. For supervised classification problems, it has been demonstrated that model averaging calculations, under some restrictions, are feasible and efficient. In this paper, we extend the expectation model averaging (EMA) algorithm originally proposed in Santafé et al. (2006) to deal with model averaging of naive Bayes models for clustering. Thus, the extended algorithm, EMA-TAN, allows to perform an efficient approximation for a model averaging over the class of tree augmented naive Bayes (TAN) models for clustering. We also present some empirical results that show how the EMA algorithm based on TAN outperforms other clustering methods.

1 Introduction

Unsupervised classification or clustering is the process of grouping similar objects or data samples together into natural groups called clusters. This process generates a partition of the objects to be classified. Bayesian networks (Jensen, 2001) are powerful probabilistic graphical models that can be used for clustering purposes. The naive Bayes is the most simple Bayesian network model (Duda and Hart, 1973). It assumes that the predictive variables in the model are independent given the value of the cluster variable. Despite this being a very simple model, it has been successfully used in clustering problems (Cheeseman and Stutz, 1996). Other models have been proposed in the literature in order to relax the heavy independence assumptions that the naive Bayes model makes. For example, tree augmented naive Bayes (TAN) models (Friedman et al., 1997) allow the predictive variables to form a tree and the class variable remains as a parent of each predictive variable. On the other hand, more complicated methods have been proposed to allow the encoding of context-specific (in)dependencies in clustering problems by using, for instance, naive Bayes (Barash and Friedman, 2002) or recursive Bayesian multinets (Peña et al., 2002) which are trees that incorporate Bayesian multinets in the leaves.

The process of selecting a single model for clustering ignores model uncertainty and it can lead to the selection of a model that does not properly describe the data. In fact, the fewer samples the dataset has, the higher the uncertainty is in model selection. In these cases, a Bayesian approach may be beneficial. The Bayesian approach proposes an averaging over all models weighted by their posterior probability given the data (Madigan and Raftery, 1994; Hoeting et al., 1999). The Bayesian model averaging has been seen by some authors as a model combination technique (Domingos, 2000; Clarke, 2003) and it does not always perform as successfully as expected. However, other authors states that Bayesian model averaging is not exactly a model ensemble technique but a method for 'soft model selection' (Minka, 2002).

Although model averaging approach is normally preferred when there are a few data samples because it deals with uncertainty in model selection, this approach is usually intractable and only approximations are feasible. Typically, the Bayesian model averaging for clustering is approximated by averaging over some of

the models with the highest posterior probabilities (Friedman, 1998). However, efficient calculation of model averaging for supervised classification models under some constraints has been proposed in the literature (Dash and Cooper, 2004; Cerquides and López de Mántaras, 2005). Some of these proposals have also been extended to clustering problems. For example, Santafé et al. (2006) extend the calculations of naive Bayes models to approximate a Bayesian model averaging for clustering. In that paper, the authors introduce the expectation model averaging (EMA) algorithm, which is a variant of the well-known EM algorithm (Dempster et al., 1977) and allows to deal with the latent cluster variable and then approximate a Bayesian model averaging of naive Bayes models for clustering.

In this paper we use the structural features estimation proposed by Friedman and Koller (2003) and the model averaging calculations for supervised classifiers presented in Dash and Cooper (2004) in order to extend the EMA algorithm to TAN models (EMA-TAN algorithm). In other words, we propose a method to obtain a single Bayesian network model for clustering which approximates a Bayesian model averaging over all possible TAN models. This is possible by setting an ancestral order among the predictive variables. The result of the Bayesian model averaging over TAN models is a single Bayesian network model for clustering (Dash and Cooper, 2004).

The rest of the paper is organized as follows. Section 2 introduces the notation that is used throughout the paper as well as the assumptions that we make. Section 3 describes the EMA algorithm to approximate model averaging over the class of TAN models. Section 4 shows some experimental results with synthetic data that illustrate the behavior of the EMA algorithm. Finally, section 5 presents the conclusions of the paper and future work.

2 Notation and Assumptions

In an unsupervised learning problem there is a set of predictive variables, $X_1, \ldots, X_n$, and the latent cluster variable, C. The dataset $D = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\}$ contains data samples $\boldsymbol{x}^{(l)} = \{x_1^{(l)}, \ldots, x_n^{(l)}\}$, with $l = 1, \ldots, N$.

We define a Bayesian network model as $\mathcal{B} = \langle S, \boldsymbol{\theta} \rangle$, where S describes the structure of the model and $\boldsymbol{\theta}$ its parameter set. Using the classical notation in Bayesian networks, θ_{ijk} (with $k = 1, \ldots, r_i$ and r_i being the number of states for variable X_i) represents the conditional probability of variable X_i taking its k-th value given that its parents, $\boldsymbol{Pa}_i$, takes its j-th configuration. The conditional probability mass function for X_i given the j-th configuration of its parents is designated as $\boldsymbol{\theta}_{ij}$, with $j = 1, \ldots, q_i$, where q_i is the number of different states of $\boldsymbol{Pa}_i$. Finally, $\boldsymbol{\theta}_i = (\boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iq_i})$ denotes the set of parameters for variable X_i, and therefore, $\boldsymbol{\theta} = (\boldsymbol{\theta}_C, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$, where $\boldsymbol{\theta}_C = (\theta_{C-1}, \ldots, \theta_{C-r_C})$ is the set of parameters for the cluster variable, with r_C the number of clusters fixed in advance.

In order to use the decomposition proposed in Friedman and Koller (2003) for an efficient model averaging calculation, we need to consider an ancestral order $\boldsymbol{\pi}$ over the predictive variables.

Definition 1. Class of TAN models ($\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$): given an ancestral order $\boldsymbol{\pi}$, a model $\mathcal{B}$ belongs to $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$ if each predictive variable has up to two parents (another predictive variable and the cluster variable) and the arcs between variables that are defined in the structure S are directed down levels: $X_j \rightarrow X_i \in S \Rightarrow level_{\boldsymbol{\pi}}(X_i) < level_{\boldsymbol{\pi}}(X_j)$.

Note that, the classical conception of TAN model allows the predictive variables to form a tree and then, the cluster variable is set as a parent of each predictive variable. However, we do not restrict $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$ to only these tree models, but we also allow the predictive variables to form a forest and also the class variable may or may not be set as a parent of each predictive variable. Therefore $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$ also includes, among others, naive Bayes and selective naive Bayes models.

For a given ordering $\boldsymbol{\pi}$ and a particular variable X_i, we can enumerate all the possible parent sets for X_i in the class of TAN models $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$. In order to clarify calculations, we superscript with v any quantity related to a predictive variable X_i and thus, we are able to identify the parent set of variable X_i that

we are taking into consideration. For example, for a given order $\boldsymbol{\pi} = \langle X_1, X_2, X_3 \rangle$ the possible sets of parents for X_3 in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$ are: $\boldsymbol{Pa}^1_3 = \{\emptyset\}, \boldsymbol{Pa}^2_3 = \{X_1\}, \boldsymbol{Pa}^3_3 = \{X_2\}, \boldsymbol{Pa}^4_3 = \{C, X_1\}, \boldsymbol{Pa}^5_3 = \{C, X_2\}, \boldsymbol{Pa}^6_3 = \{C\}$ with, in this case, $v = 1, \ldots, 6$. In general, we consider, without loss of generality, $\boldsymbol{\pi} = \langle X_1, \ldots, X_n \rangle$ and therefore, for a variable X_i, $v = 1, \ldots, 2i$. Moreover, we use i to index any quantity related to the i-th predictive variable, with $i = 1, \ldots, n$.

Additionally, the following five assumptions are needed to perform an efficient approximation of model averaging over $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$:

Multinomial variables: Each variable X_i is discrete and can take r_i states. The cluster variable is also discrete and can take r_C possible states, r_C being the number of clusters fixed in advance.

Complete dataset: We assume that there are no missing values for the predictive variables in the dataset. However, the cluster variable is latent; therefore, its values are always missing.

Dirichlet priors: The parameters of every model are assumed to follow a Dirichlet distribution. Thus, α_{ijk} is the Dirichlet hyperparameter for parameter θ_{ijk} from the network, and α_{C-j} is the hyperparameter for θ_{C-j}. In fact, as we have to take into consideration each possible model in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$, the parameters of the models can be denoted as θ^v_{ijk}. Hence, we assume the existence of hyperparameters α^v_{ijk}.

Parameter independence: The probability of having the set of parameters $\boldsymbol{\theta}$ for a given structure S can be factorized as follows:

$$p(\boldsymbol{\theta}|S) = p(\boldsymbol{\theta}_C) \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|S) \tag{1}$$

Structure modularity: The prior probability $p(S)$ can be decomposed in terms of each variable and its parents:

$$p(S) \propto p_S(C) \prod_{i=1}^{n} p_S(X_i, \boldsymbol{Pa}_i) \tag{2}$$

where $p_S(X_i, \boldsymbol{Pa}_i)$ is the information contributed by variable X_i to $p(S)$, and $p_S(C)$ is the information contributed by the cluster variable.

Parameter independence assumes that the prior on parameters θ_{ijk} for a variable X_i depends only on local structures. This is known as parameter modularity (Heckerman et al., 1995). Therefore, we can state that for any two network structures S_1 and S_2, if X_i has the same parent set in both structures then $p(\theta_{ijk}|S_1) = p(\theta_{ijk}|S_2)$. As a consequence, parameter calculations for a variable X_i will be the same in every model whose structure defines that the variable X_i has the same parent set.

3 The EMA-TAN Algorithm

The EMA algorithm was originally introduced in Santafé et al. (2006) for dealing with model averaging calculations of naive Bayes models. In this section we present the extension of this algorithm (EMA-TAN) in order to average over TAN models.

Theorem 1 (Dash and Cooper, 2004)**.** *There exist, for supervised classification problems, a single model $\tilde{\mathcal{B}} = \langle \tilde{S}, \tilde{\boldsymbol{\theta}} \rangle$ which defines a joint probability distribution $p(c, \boldsymbol{x}|\tilde{\mathcal{B}})$ equivalent to the joint probability distribution produced by model averaging over all TAN models. This model $\tilde{\mathcal{B}}$ is a complete Bayesian network where the structure $\tilde{S}$ defines the relationship between variables in such a way that, for a variable X_i, the parent set of X_i in the model $\tilde{\mathcal{B}}$ is $\tilde{\boldsymbol{Pa}}_i = \cup_{v=1}^{2i} \boldsymbol{Pa}^v_i$.*

This result can be extended to clustering problems by means of the EMA-TAN algorithm. However, the latent cluster variable prevents the exact calculation of the model averaging and therefore the model $\tilde{\mathcal{B}}$ obtained by the EMA-TAN algorithm is not an exact model averaging over $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$ but an approximation. Therefore, the EMA-TAN algorithm provides a powerful tool that allows to learn a single unsupervised Bayesian network model which approximates Bayesian model averaging over $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$.

The unsupervised classifier is obtained by learning the predictive probability, $p(c, \boldsymbol{x}|D)$, averaged over the *maximum a posteriori* (MAP) parameter configurations for all the models in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$. Then, we can obtain the unsupervised classifier by using the conditional probability of the cluster variable given by Bayes' rule.

The EMA-TAN algorithm is an adaption of

the well-known EM algorithm. It uses the E step of the EM algorithm to deal with the missing values for the cluster variable. Then, it performs a model averaging step (MA) to obtain $p(c, \boldsymbol{x}|D)$ and thus the unsupervised Bayesian network model for clustering.

The EMA-TAN, as well as the EM algorithm, is an iterative process where the two steps of the algorithm are repeated successively until a stopping criterion is met. At the t-th iteration of the algorithm, a set of parameters, $\tilde{\boldsymbol{\theta}}^{(t)}$, for the Bayesian network model $\tilde{\mathcal{B}}^{(t)}$ is calculated. Note that, although we differentiate between the Bayesian network models among the iterations of the EMA-TAN algorithm, the structure of the model, $\tilde{\boldsymbol{S}}$, is constant and only the estimated parameter set changes. The algorithm stops when the difference between the sets of parameters learned in two consecutive iterations, $\tilde{\boldsymbol{\theta}}^{(t)}$ and $\tilde{\boldsymbol{\theta}}^{(t+1)}$, is less than threshold ϵ, which is fixed in advance.

In order to use the EMA-TAN algorithm, we need to set an initial parameter configuration, $\tilde{\boldsymbol{\theta}}^{(0)}$, and the value of ϵ. The values for $\tilde{\boldsymbol{\theta}}^{(0)}$ are usually taken at random and ϵ is set at a small value. Note that, even though the obtained model is a single unsupervised Bayesian network, its parameters are learned taking into account the MAP parameter configuration for every model in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$. Thus, the resulting unsupervised Bayesian network will incorporate into its parameters information about the (in)dependencies between variables described by the different TAN models.

3.1 E Step (Expectation)

Intuitively, we can see this step as a *completion* of the values for the cluster variable, which are missing. Actually, this step computes the expected sufficient statistics in the dataset for variable X_i in every model in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$ given the current model, $\tilde{\mathcal{B}}^{(t)}$. These expected sufficient statistics are used in the next step of the algorithm, MA, as if they were actual sufficient statistics from a complete dataset. From now on, $D^{(t)}$ denotes the dataset after the E step at the t-th iteration of the algorithm.

Note that, due to parameter modularity, we do not actually need to calculate the expected sufficient statistics for all the models in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$ because some of these models share the same value for the expected sufficient statistics. Hence, it is only necessary to calculate the expected sufficient statistics with different parent sets. They can be obtained as follows:

$$E(N^v_{ijk}|\tilde{\mathcal{B}}^{(t)}) = \sum_{l=1}^{N} p(x_i^k, \boldsymbol{Pa}_i^v = j|\boldsymbol{x}^{(l)}, \tilde{\mathcal{B}}^{(t)}) \quad (3)$$

where x_i^k represents the k-th value of the i-th variable. The expected sufficient statistic $E(N^v_{ijk}|\tilde{\mathcal{B}}^{(t)})$ denotes, at iteration t, the expected number of cases in the dataset D where variable X_i takes its k-th value, and the v-th parent set of X_i takes its j-th configuration.

Similarly, we can obtain the expected sufficient statistics for the cluster variable. This is a special case since for any model in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$ the parent set for C is the same (the cluster variable does not have any parent). Therefore, we refuse the use of superindex v in those quantities related only to C.

$$E(N_{C-j}|\tilde{\mathcal{B}}^{(t)}) = \sum_{l=1}^{N} p(C = j|\boldsymbol{x}^{(l)}, \tilde{\mathcal{B}}^{(t)}) \quad (4)$$

Note that, some of the expected sufficient statistics $E(N^v_{ijk}|\tilde{\mathcal{B}}^{(t)})$ do not depend on the value of C. Therefore, these values are constant throughout the iterations of the algorithm and it is necessary to calculate them only once.

3.2 MA Step (Model Averaging)

In this second step, the EMA algorithm performs the model averaging calculations which obtain a single Bayesian network model with parameters $\tilde{\boldsymbol{\theta}}^{(t+1)}$. These parameters are obtained by calculating $p(c, \boldsymbol{x}|D^{(t)})$ as an average over the MAP configurations for the models in $\mathcal{L}^{\boldsymbol{\pi}}_{TAN}$.

In order to make the calculations clearer, we first show how we can obtain $p(c, \boldsymbol{x}|S, D^{(t)})$ for a fixed structure S:

$$p(c, \boldsymbol{x}|S, D^{(t)}) = \int p(c, \boldsymbol{x}|S, \boldsymbol{\theta}) p(\boldsymbol{\theta}|S, D^{(t)}) d\boldsymbol{\theta} \quad (5)$$

The exact computation of the integral in Equation 5 is intractable for clustering problems, therefore, an approximation is needed

(Heckerman et al., 1995). However, assuming parameter independence and Dirichlet priors, and given that the expected sufficient statistics calculated in the previous E step can be used as an approximation to the actual sufficient statistics in the complete dataset, we can approximate $p(c, \boldsymbol{x}|S, D^{(t)})$ by the MAP parameter configuration. This is the parameter configuration that maximizes $p(\boldsymbol{\theta}|S, D^{(t)})$ and can be described in terms of the expected sufficient statistics and the Dirichlet hyperparameters (Heckerman et al., 1995; Cooper and Herskovits, 1992). Therefore, Equation 5 results:

$$p(c, \boldsymbol{x}|S, D^{(t)}) \approx \frac{\alpha_{C-j} + E(N_{C-j}|\tilde{\mathcal{B}}^{(t)})}{\alpha_C + E(N_C|\tilde{\mathcal{B}}^{(t)})} \cdot \quad (6)$$

$$\prod_{i=1}^{n} \frac{\alpha_{ijk}^{\mu_i} + E(N_{ijk}^{\mu_i}|\tilde{\mathcal{B}}^{(t)})}{\alpha_{ij}^{\mu_i} + E(N_{ij}^{\mu_i}|\tilde{\mathcal{B}}^{(t)})} = \hat{\theta}_{C-j} \prod_{i=1}^{n} \hat{\theta}_{ijk}^{\mu_i}$$

where $\hat{\theta}_{ijk}^{\mu_i}$ is the MAP parameter configuration for $\theta_{ijk}^{\mu_i}$ (μ_i denotes the parent index that corresponds to the parent set for X_i described in S), $\alpha_{ij}^{\mu_i} = \sum_{k=1}^{r_i} \alpha_{ijk}^{\mu_i}$, $E(N_{ij}|\tilde{\mathcal{B}}^{(t)}) = \sum_{k=1}^{r_i} E(N_{ijk}|\tilde{\mathcal{B}}^{(t)})$ and similarly for the values related to C.

Considering that the structure is not fixed *a priori*, we should average over all selective model structures in $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$ in the following way:

$$p(c, \boldsymbol{x}|D^{(t)}) = \quad (7)$$

$$\sum_S \int p(c, \boldsymbol{x}|S, \boldsymbol{\theta}) p(\boldsymbol{\theta}|S, D^{(t)}) d\boldsymbol{\theta} \;\; p(S|D^{(t)})$$

Therefore, the model averaging calculations require a summation over $2^n n!$ terms, which are the models in $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$.

Using the previous calculations for a fixed structure, Equation 8 can be written as:

$$p(c, \boldsymbol{x}|D^{(t)}) \;\; \approx \;\; \sum_S \hat{\theta}_{C-j} \prod_{i=1}^{n} \hat{\theta}_{ijk}^{\mu_i} \;\; p(S|D^{(t)})$$

$$\propto \sum_S \hat{\theta}_{C-j} \prod_{i=1}^{n} \hat{\theta}_{ijk}^{\mu_i} \;\; p(D^{(t)}|S) \;\; p(S) \quad (8)$$

Given the assumption of Dirichlet priors and parameter independence, we can approximate $p(D^{(t)}|S)$ efficiently. In order to do so, we adapt the formula to calculate the marginal likelihood with complete data (Cooper and Herskovits, 1992) to our problem with missing values. Thus, we have an approximation to $p(D^{(t)}|S)$:

$$p(D^{(t)}|S) \approx \frac{\Gamma(\alpha_C)}{\Gamma(\alpha_C + E(N_C|\tilde{\mathcal{B}}^{(t)}))} \prod_{j=1}^{r_C} \frac{\Gamma(\alpha_{C-j} + E(N_{C-j}|\tilde{\mathcal{B}}^{(t)}))}{\Gamma(\alpha_{C-j})} \cdot$$

$$\prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij}^{\mu_i})}{\Gamma(\alpha_{ij}^{\mu_i} + E(N_{ij}^{\mu_i}|\tilde{\mathcal{B}}^{(t)}))} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk}^{\mu_i} + E(N_{ijk}^{\mu_i}|\tilde{\mathcal{B}}^{(t)}))}{\Gamma(\alpha_{ijk}^{\mu_i})}$$

At this point, given structure modularity assumption, we are able to approximate $p(c, \boldsymbol{x}|D^{(t)})$ with the following expression:

$$p(c, \boldsymbol{x}|D^{(t)}) \approx \kappa \sum_S \rho_{C-j} \prod_{i=1}^{n} \rho_{ijk}^{\mu_i} \quad (9)$$

where κ is a constant and ρ_{C-j} and $\rho_{ijk}^{\mu_i}$ are defined in Equation 10.

$$\rho_{C-j} = \hat{\theta}_{C-j} \;\; p_S(C) \frac{\Gamma(\alpha_C)}{\Gamma(\alpha_C + E(N_C|\mathcal{B}^{(t)}))} \cdot$$

$$\prod_{j=1}^{r_C} \frac{\Gamma(\alpha_{C-j} + E(N_{C-j}|\mathcal{B}^{(t)}))}{\Gamma(\alpha_{C-j})} \quad (10)$$

$$\rho_{ijk}^{\mu_i} = \hat{\theta}_{ijk}^{\mu_i} \;\; p_S(X_i, \boldsymbol{Pa}_i^{\mu_i}) \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij}^{\mu_i})}{\Gamma(\alpha_{ij}^{\mu_i} + E(N_{ij}^{\mu_i}|\mathcal{B}^{(t)}))} \cdot$$

$$\prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk}^{\mu_i} + E(N_{ijk}^{\mu_i}|\mathcal{B}^{(t)}))}{\Gamma(\alpha_{ijk}^{\mu_i})}$$

Since we are assuming parameter independence, structure modularity and parameter modularity, we can apply the dynamic programming solution described in Friedman and Koller (2003), and Dash and Cooper (2004). Thus Equation 9 can be written as follows:

$$p(c, \boldsymbol{x}|D^{(t)}) \approx \lambda \;\; \rho_{C-j} \prod_{i=1}^{n} \sum_{v=1}^{2i} \rho_{ijk}^{v} \quad (11)$$

with λ being a constant.

Note that the time complexity needed to calculate the averaging over $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$ is the same as that which is needed to learn the MAP parameter configuration for $\tilde{\mathcal{B}}$.

We can see the similarity of the above-described Equation 11 with the factorization of a Bayesian network model. Indeed the joint probability distribution of the approximated Bayesian model averaging over $\mathcal{L}_{TAN}^{\boldsymbol{\pi}}$ for clustering is equivalent to a single Bayesian network model. Therefore, the parameters of the model for the next iteration of the algorithm can be calculated as follows:

$$\tilde{\theta}_{C-j}^{(t+1)} \propto \rho_{C-j} \quad , \quad \tilde{\theta}_{ijk}^{(t+1)} \propto \sum_{v=1}^{2i} \rho_{ijk}^{v} \qquad (12)$$

3.3 Multi-start EMA-TAN

The EMA-TAN is a greedy algorithm that is susceptible to be trapped in a local optima. The results obtained by the algorithm depend on the random initialization of the parameters. Therefore, we propose the use of a multi-start scheme where m different runs of the algorithm with different random initializations are performed. In Santaf et al. (2006) different criteria to obtain the final model from the multi-start process are proposed. In our case, we use the same criteria that the multi-start EM uses: the best model in terms of likelihood among all the m calculated models is selected as the final model. This is not a pure Bayesian approach to the model averaging process but, in practice, it works as well as other more complicated techniques.

4 Evaluation in Clustering Problems

It is not easy to validate clustering algorithms since clustering problems do not normally provide information about the true grouping of data samples. In general, it is quite common to use synthetic data because the true model that generated the dataset as well as the underlying clustering structure of the data are known. In order to illustrate the behavior of the EMA-TAN algorithm, we compare it with the classical EM algorithm and with the EMA algorithm (Santafé et al., 2006). For EMA-TAN evaluation, we obtain random TAN models where the number of predictive variables vary in $\{2, 4, 8, 10, 12, 14\}$, each predictive variable can take up to three states and the number of clusters is set to two. For each TAN configuration we generate 100 random models and each one of these models is sampled to obtain different datasets of sizes $40, 80, 160, 320$ and 640. In the experiments, we compare the multi-start EMA-TAN (called also EMA-TAN for convenience) with three different algorithms:

EM-TAN: a multi-start EM that learns a TAN model by using, at each step of the EM algorithm, the classical method proposed by Friedman et al. (1997) adapted to be used within the EM algorithm.

EM-BNET: a multi-start EM algorithm used to learn the MAP parameters of a complete Bayesian network for a given order $\boldsymbol{\pi}$.

EMA: the multi-start model averaging of naive Bayes for clustering.

Note that, for a given order $\boldsymbol{\pi}$, both EMA-TAN and EM-BNET models share the same network structure but their parameter sets are calculated in a different way. The number of multi-start iterations for both multi-start EM and multi-start EMA is $m = 100$.

Since the datasets are synthetically generated, we are able to know the real ordering among variables. Nevertheless, we prefer to use a more general approach and assume that this order is unknown. Therefore, we use a random ordering among predictive variables for each EMA-TAN model that we learn. As the EM-BNET algorithm also needs an ancestral order among predictive variables, in the experiments, we use the same random ordering for any pair of models (EMA-TAN vs. EM-BNET) that we compare.

Every model is used to cluster the dataset from which it has been learned. In the experiments, we compare EMA-TAN vs. EM-TAN, EMA-TAN vs. EM-BNET and EMA-TAN vs. EMA. For each test, the winner model is obtained by comparing the data partitions obtained by both models with the true partition of the dataset. Thus, the model with the best estimated accuracy (percentage of correctly classified instances) is the winner model. In Table 1, the results from the experiments with random TAN models are shown. For each model configuration, the table describes the number of wins/draws/losses of the EMA-TAN models with respect to EM-TAN, EM-BNET or EMA models on basis of the estimated accuracy of each model. We also provide information about

a Wilcoxon signed-rank test[1] used to evaluate whether the accuracy estimated by two different models is different at the 1% and 10% levels. We write the results shown in Table 1 in bold if the test is surpassed at the 10% level and inside a gray color box if the test is surpassed at the 1% level. It can be seen that, in general, the EMA-TAN models behave better than the others and the differences between estimated accuracy are, in most of the cases, statistically significant.

We can see that the compared models obtain very similar results when they have a few predictive variables. This is because the set of models that we are averaging over to obtain the EMA-TAN model is very small. Therefore, it is quite possible that other algorithms such as EM-TAN select the correct model. Hence, in some experiments with the simplest models (models with 2 predictive variables), EMA-TAN algorithm significantly lost with the other algorithms. However, when the number of predictive variables in the model increases and the dataset size is relatively big (the smallest datasets are not big enough for a reliable estimation of the parameters) the EMA-TAN considerably outperforms any other model in the test.

The experimental results from this section reinforce the idea that the results of the Bayesian model averaging outperform other methods when the model that generated the data is included in the set of models that we are averaging over. Since we are averaging over a restricted class of models, this situation may not be fulfilled when applying the EMA-TAN to real problems. Due to the lack of space, we do not include in the paper more experimentation in progress, but we are aware that it would be very interesting to check the performance of the EMA algorithm with other synthetic datasets generated by sampling naive Bayes and general Bayesian network models and also with dataset from real problems.

5 Conclusions

We have shown that it is possible to obtain a single unsupervised Bayesian network that approximates model averaging over the class of TAN models. Furthermore, this approximation can be performed efficiently. This is possible by using the EMA-TAN algorithm. The EMA is an algorithm originally proposed in Santafé et al. (2006) for averaging over naive Bayes models in clustering problems. In this paper we extend the algorithm to deal with more complicated models such as TAN (EMA-TAN algorithm). We also present an empirical evaluation by comparing the model averaging over TAN models with the model averaging over naive Bayes models and with the EM algorithm to learn a single TAN model and a Bayesian network model. These experiments conclude that, at least, when the model that generated the dataset is included in the set of models that we average over, the averaging over TAN models outperforms the other methods

Probably, one of the limitations of the proposed algorithm is that, because the learned model which approximates model averaging is a complete Bayesian network, it is computationally hard to learn the model for problems with many predictive variables. In order to overcome this situation, we can restrict the final model to a maximum of k possible parents for each predictive variable. Future work might include a more exhaustive empirical evaluation of the EMA-TAN algorithm and the application of the algorithm to real problems. Moreover, the algorithm can be extended to other Bayesian classifiers such as k-dependent naive Bayes (kDB), selective naive Bayes, etc. Another interesting future work may be the relaxation of complete dataset assumption.

[1]Since we have checked by means of a Kolmogorov-Smirnov test that not all the outcomes from the experiment can be considered to follow a normal distribution, we decided to use a Wilcoxon sign-rank test to compare the results

Acknowledgments

This work was supported by the SAIOTEK-Autoinmune (II) 2006 and Etortek research projects from the Basque Government, by the Spanish Ministerio de Educación y Ciencia under grant TIN 2005-03824, and by the Government of Navarre under a PhD grant.

References

Y. Barash and N. Friedman. 2002. Context-specific Bayesian clustering for gene expression data. *Journal of Computational Biology*, 9:169–191.

EMA-TAN vs. EM-TAN

#Var	40	80	160	320	640
2	11/71/18	**9/71/20**	15/68/17	18/70/12	**15/63/22**
4	37/24/39	**33/17/50**	50/8/42	48/5/47	46/3/51
8	51/6/43	**57/4/39**	**65/6/29**	**74/4/22**	**71/1/28**
10	48/5/47	**59/4/37**	**55/10/35**	**64/6/30**	**67/4/29**
12	**54/4/42**	**65/3/32**	**65/4/31**	**69/4/27**	**83/2/15**
14	**59/8/33**	**68/6/26**	**75/2/23**	**76/4/20**	**76/6/18**

EMA-TAN vs. EM-BNET

#Var	40	80	160	320	640
2	24/51/25	29/35/36	33/34/33	28/35/37	41/29/30
4	**54/11/35**	51/8/41	**59/8/33**	49/6/45	**57/1/42**
8	**59/8/33**	**64/5/31**	**81/1/18**	**83/0/17**	**91/0/9**
10	**67/7/26**	**76/0/24**	**82/2/16**	**84/0/16**	**94/0/6**
12	**66/5/29**	**86/1/13**	**80/0/20**	**91/0/9**	**89/0/11**
14	**74/2/24**	**83/1/16**	**92/1/7**	**92/0/8**	**96/0/4**

EMA-TAN vs. EMA

#Var	40	80	160	320	640
2	21/53/26	**24/41/35**	24/48/28	23/45/32	25/45/30
4	47/14/39	49/6/45	49/10/41	50/5/45	56/3/41
8	**52/10/38**	**58/4/38**	**80/4/16**	**90/0/10**	**89/0/11**
10	48/13/39	**55/8/37**	**69/6/25**	**81/1/18**	**88/1/11**
12	**55/8/37**	**78/6/16**	**79/1/20**	**88/2/10**	**88/0/12**
14	**55/11/34**	**68/7/25**	**81/3/16**	**84/3/13**	**92/0/8**

Table 1: Comparison between EMA-TAN models and EM-TAN, EM-BNET and EMA learned from datasets sampled from random TAN models

J. Cerquides and R. López de Mántaras. 2005. TAN classifiers based on decomposable distributions. *Machine Learning*, 59(3):323–354.

P. Cheeseman and J. Stutz. 1996. Bayesian classification (Autoclass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI Press.

B. Clarke. 2003. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research*, 4:683–712.

G. F. Cooper and E. Herskovits. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

D. Dash and G. F. Cooper. 2004. Model averaging for prediction with discrete Bayesian networks. *Journal of Machine Learning Research*, 5:1177–1203.

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

P. Domingos. 2000. Bayesian averaging of classifiers and the overfitting problem. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 223–230.

R. Duda and P. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley and Sons.

N. Friedman and D. Koller. 2003. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50:95–126.

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

N. Friedman. 1998. The Bayesian structural EM algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 129–138.

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

J. Hoeting, D. Madigan, A. E. Raftery, and C. Volinsky. 1999. Bayesian model averaging. *Statistical Science*, 14:382–401.

F.V. Jensen. 2001. *Bayesian Networks and Decision Graphs*. Springer Verlag.

D. Madigan and A. E. Raftery. 1994. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, 89:1535–1546.

T. Minka. 2002. Bayesian model averaging is not model combination. MIT Media Lab note.

J. M. Peña, J. A. Lozano, and P. Larrañaga. 2002. Learning recursive Bayesian multinets for clustering by means of constructive induction. *Machine Learning*, 47(1):63-90.

G. Santafé, J. A. Lozano, and P. Larrañaga. 2006. Bayesian model averaging of naive Bayes for clustering. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*. Accepted for publication.