Bayesian Model Averaging of Naive Bayes for Clustering

Guzmán Santafé, Jose A. Lozano, *Member, IEEE*, and Pedro Larrañaga

***Abstract*—This paper considers a Bayesian model-averaging (MA) approach to learn an unsupervised naive Bayes classification model. By using the expectation model-averaging (EMA) algorithm, which is proposed in this paper, a unique naive Bayes model that approximates an MA over selective naive Bayes structures is obtained. This algorithm allows to obtain the parameters for the approximate MA clustering model in the same time complexity needed to learn the maximum-likelihood model with the expectation–maximization algorithm. On the other hand, the proposed method can also be regarded as an approach to an unsupervised feature subset selection due to the fact that the model obtained by the EMA algorithm incorporates information on how dependent every predictive variable is on the cluster variable.**

***Index Terms*—Bayesian model averaging (MA), clustering, expectation–maximization (EM), naive Bayes.**

I. 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. The partition can be crisp or can be done assigning to each object or data sample a certain probability of which belongs to each different cluster (probabilistic clustering). Although crisp clustering has been widely used in the literature, clustering based on probability models has become more fashionable because it does not only offer data partition but also provides information about clustering uncertainty and some knowledge of the process that has generated the dataset [1], [2].

This paper faces the probabilistic clustering problem by using Bayesian networks [3]–[5], which are powerful tools to model probability distributions. Specifically, the paper proposes a new method to learn a naive Bayes clustering model (see Fig. 1). This is a kind of Bayesian network which, despite its simplicity, has been satisfactorily used in real complex clustering problems [6], [7].

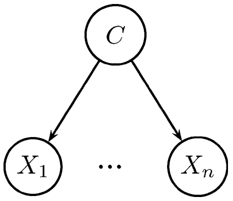


Fig. 1. Structure for a naive Bayes model for clustering. All the predictive variables $X_1, \ldots, X_n$ are independent given the cluster variable C which is latent in the dataset.

Normally, in real clustering problems, the only available information is the dataset. Therefore, even if a model is learned from the data, there is no guarantee that the obtained model should correspond to the true model, that is, the model that generated the data. Some classical methods for learning Bayesian network classification models for clustering, such as the expectation–maximization (EM) algorithm [8], [9], need to assume a model structure. Then, the algorithm approximates the maximum likelihood (ML) or the maximum *a posteriori* (MAP) parameters for the assumed structure, but actually, this structure may not be the one that best models the data. Hence, some other techniques, such as the structural EM algorithm [10], [11], have been proposed. This algorithm searches through the joint space of structures and parameters in order to find the model that best describes the data. Nevertheless, the fact that the structure and the parameters are the most likely ones given the dataset neither guarantee that this is the model that generated the data. Moreover, the fewer samples the dataset has, the higher the uncertainty is about the fact that the most likely model is the true model. In those cases where there are few data samples and the true model for the dataset is unknown *a priori*, the best choice for learning a model is the Bayesian approach. The Bayesian approach obtains the model by averaging over all model structures and all parameter configurations weighted by their posterior probability given the dataset. This is also known as Bayesian model averaging (MA) [12], [13]. However, Bayesian MA is usually intractable. Several approximations are proposed in the literature to deal with Bayesian MA with missing data, for instance select the MAP model [14] or average over some of the models with the highest posterior probabilities [15].

This paper proposes a new method to approximate the Bayesian MA of naive Bayes for clustering problems. It has already been demonstrated that Bayesian MA calculations are feasible and efficient for the learning of Bayesian networks from complete data [16] and for supervised Bayesian classification models [17]–[19]. However, in clustering problems, it is unfeasible to average over all parameter configurations because of the latent cluster variable, which makes some integrals unresolvable in closed form. Therefore, this paper proposes an approach to Bayesian MA by averaging over all

Manuscript received October 27, 2005; revised February 3, 2006. This work was supported in part by the ETORTEK and SAIOTEK programs of the Basque Government, in part by the Spanish Ministerio de Educación y Ciencia under Grant TIN 2005-03824, in part by the University of the Basque Country under Grant 9/UPV 00140.226-15334/2003, and in part by the Government of Navarre under a Ph.D. grant. This paper was recommended by Associate Editor T. Sudkamp.

The authors are with Intelligent Systems Group, Computer Science and Artificial Intelligent Department, University of the Basque Country, 20018 San Sebastián, Spain (e-mail: guzman@si.ehu.es; ja.lozano@ehu.es; pedro.larranaga@ehu.es; ccplamup@si.ehu.es).

Digital Object Identifier 10.1109/TSMCB.2006.874132

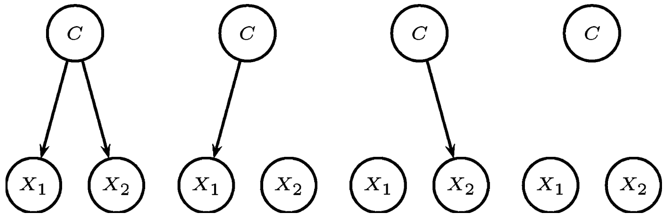


Fig. 2. All selective naive Bayes structures with two predictive variables. Each predictive variable can be dependent on or independent of C.

selective naive Bayes structures (see Fig. 2) where, for each structure, the averaging over parameters is approximated by its MAP configuration. In other words, we obtain a unique naive Bayes model for clustering, which is equivalent to the sum of the MAP configurations for every selective naive Bayes structure weighted by its posterior probability. To manage these calculations efficiently, we extend the Bayesian MA approach described by [20] to clustering problems, and we introduce the expectation MA (EMA) algorithm. This algorithm, which incorporates the MA calculations, is a variant of the well-known EM algorithm.

Since all the selective naive Bayes models are taken into account to obtain the final model by means of the EMA algorithm, the parameters of the obtained model include information about the degree of dependence of each predictive variable on the cluster variable. That is, the method proposed in this paper learns a naive Bayes model for clustering and, indirectly, includes a kind of Bayesian feature subset selection. The final model contains all the predictive variables but its parameters incorporate information about the importance of each variable for the clustering purpose. This may be very useful for common clustering problems where we have no information about the relations between variables and where all the variables may be not equally relevant.

The learning method proposed in this paper also contributes to the efficient calculation of the model. The unsupervised model learned with the EMA algorithm can be calculated in the same time complexity required to learn the ML or MAP parameters with the EM algorithm. Therefore, with similar computational cost, the EMA model accounts for the uncertainty of the model and includes information about the relevance of the predictive variables.

The rest of this paper is organized as follows. Section II introduces the notation that is used throughout the paper as well as the assumptions that we make. Section III describes the EMA algorithm and, in particular, the two steps of the algorithm: expectation and MA. Section IV presents an empirical test in order to prove that the approximation obtained by the EMA algorithm is, in fact, comparable to an MA over MAP configurations for selective naive Bayes structures. Section V provides evidence on how the EMA algorithm, used to learn a model from a dataset, is able to detect the model that has generated this dataset. Additionally, Section VI tests the behavior of the EMA algorithm in clustering problems with both synthetic and real data. Finally, Section VII presents the conclusions yielded from the paper as well as future work. In order to perform the empirical evaluation of the EMA algorithm, it has been implemented into both Elvira [21] and Weka [22] frameworks.

II. Notation and Assumptions

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

Using the classical notation in Bayesian networks, the set of parents for variable X_i, with $i = 1, \ldots, n$, is denoted as Pa_i. In our case, for all selective naive Bayes models, $\mathrm{Pa}_i \in \{\emptyset, \{C\}\}$. θ_{ijk}, with $k = 1, \ldots, r_i$ and being r_i the number of states for variable X_i, represents the conditional probability of variable X_i taking its kth value given that Pa_i takes its jth value. The conditional probability mass function for X_i given the jth configuration of its parents is designated as $\boldsymbol{\theta}_{ij}$, with $j = 1, \ldots, q_i$, where q_i is the number of different configurations of Pa_i. Finally, $\boldsymbol{\theta}_i = (\boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iq_i})$ denotes the set of parameters for variable X_i, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_C, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ represents the whole set of parameters for a selective naive Bayes model, where $\boldsymbol{\theta}_C = (\boldsymbol{\theta}_{C-1}, \ldots, \boldsymbol{\theta}_{C-r_C})$ is the set of parameters for the cluster variable, with r_C the number of clusters fixed in advance.

In order to distinguish between the parameters for different selective naive Bayes models, we introduce the notation θ_{ijk} and θ_{i-k} to denote the parameter θ_{ijk} when there is an arc between C and X_i, and when there is none, respectively. By extension to a general case, we take into consideration the same notation (Q_{ijk} and Q_{i-k}) with any quantity related to variable X_i.

Finally, we need to make the following five assumptions in order to perform the approximation to Bayesian MA.

Assumption 1—Multinomial Variables: Each variable X_i, with $i = 1, \ldots, n$, is discrete and can take r_i states. The cluster variable is also discrete and, as introduced before, it takes r_C possible states, with r_C as the number of clusters fixed in advance.

Assumption 2—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.

Assumption 3—Dirichlet Priors: The parameters of the selective naive Bayes models 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}. Moreover, we have to take into consideration each possible selective naive Bayes whose parameters can be θ_{ijk} or θ_{i-k}. Hence, we assume the existence of both sets of hyperparameters α_{ijk} and α_{i-k}.

Assumption 4—Parameter Independence: For any possible structure S, the probability distributions $\boldsymbol{\theta}_{ij}$ are random variables, which are considered independent for any i and j. Thus, the probability of having the set of parameters $\boldsymbol{\theta}$ for a given structure S can be factorized as follows:

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

Assumption 5—Structure Modularity: For any possible selective naive Bayes structure, S, we have a prior probability, $p(S)$. The structure modularity assumption states that the prior over structures, $p(S)$, can be decomposed in terms of each variable and its parents

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

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

III. EMA Algorithm

In this section, we present the EMA algorithm. This is a method to learn a unique unsupervised naive Bayes model comparable to a Bayesian MA over selective naive Bayes models. The unsupervised classifier is obtained by means of learning the predictive probability, $p(C, \boldsymbol{X}|D)$, averaged over the MAP configurations for all selective naive Bayes models. Thus, we can obtain the unsupervised classifier using the conditional probability of the cluster variable

$$p(c^i|\boldsymbol{x}, D) = \frac{p(c^i, \boldsymbol{x}|D)}{\sum_{j=1}^{r_C} p(c^j, \boldsymbol{x}|D)}. \tag{3}$$

The EMA 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 an MA step to obtain $p(C, \boldsymbol{X}|D)$ and thus the unsupervised naive Bayes model.

The EMA, 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 tth iteration of the algorithm, a set of parameters $\boldsymbol{\theta}^{(t)}$ for a naive Bayes model is calculated. The algorithm stops when the difference between the sets of parameters learned in two consecutive iterations, $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t+1)}$, is less than threshold ϵ, which is fixed in advance.

In order to use the EMA algorithm, we need to set an initial parameter configuration $\boldsymbol{\theta}^{(0)}$ and the value of ϵ. The values for $\boldsymbol{\theta}^{(0)}$ are usually taken at random and ϵ is set at a small value. Note that, even though the obtained model is a unique naive Bayes, its parameters are learned taking into account the MAP parameter configuration for every selective naive Bayes structure. Thus, the resulting unsupervised naive Bayes will incorporate into its parameters information about the independence between variables described by the different selective naive Bayes models.

A. 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 given the current parameters of the model, $\boldsymbol{\theta}^{(t)}$. 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 tth iteration of the algorithm.

The expected sufficient statistics, which are used in the MA step to estimate a new model, can be obtained as follows:

$$E\left(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right) = \sum_{l=1}^{n} p\left(c^{\,j}, x_i^k|\boldsymbol{x}^{(l)}, \boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right) \tag{4}$$

where c^j is the jth value for the class variable, x_i^k is the kth value for variable X_i, and S_{nB} is the structure of the model that approximates the Bayesian MA, that is, a naive Bayes structure. The expected sufficient statistic $E(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$ denotes, at iteration t, the expected number of cases in the dataset D where variable X_i takes the value x_i^k, and C takes c^j.

Similarly, we can obtain the expected sufficient statistics for the variable X_i in those selective naive Bayes models where X_i is independent of C and for the cluster variable

$$E\left(N_{i-k}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right) = \sum_{l=1}^{n} p\left(x_i^k|\boldsymbol{x}^{(l)}, \boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)$$

$$E\left(N_{C-j}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right) = \sum_{l=1}^{n} p\left(c^j|\boldsymbol{x}^{(l)}, \boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right). \tag{5}$$

Note that, in fact, $E(N_{i-k}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$ does not depend on the value of C. It denotes the number of cases where the variable X_i takes its kth value. Therefore, these values are constant throughout the iterations of the algorithm and it is necessary to calculate them only once.

B. MA Step

In the classical EM algorithm, the second step is called maximization (M). In this step, the algorithm reestimates the parameters of the model. Hence, the new parameters approximate the ML or MAP parameter configuration, given the expected sufficient statistics calculated in the previous E step. Instead, the EMA algorithm performs the MA step, which obtains a unique naive Bayes model with parameters $\boldsymbol{\theta}^{(t+1)}$. These parameters are obtained by calculating $p(C, \boldsymbol{X}|D^{(t)})$ as an average over the MAP configurations for the 2^n selective naive Bayes structures.

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

$$p\left(c, \boldsymbol{x}|S, D^{(t)}\right) = \int p(c, \boldsymbol{x}|S, \boldsymbol{\theta}) p\left(\boldsymbol{\theta}|S, D^{(t)}\right) d\boldsymbol{\theta}. \tag{6}$$

The exact computation of the integral in (6) is intractable, therefore, an approximation is needed [14]. 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 $E(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$, and α_{ijk} [14], [23]. The MAP configuration can be calculated by transforming the coordinate system into the canonical coordinate system $\boldsymbol{\phi}_{ij} = (\phi_{ij2}, \ldots, \phi_{ijr_i})$, where $\phi_{ijk} = \log(\theta_{ijk}/\theta_{ij1})$ for

$k = 2, \ldots, r_i$. See [24] and [25] for a derivation of the MAP parameters. Using the previous considerations, (6) results

$$\begin{aligned} p\left(c, \boldsymbol{x}|S, D^{(t)}\right) &\approx \frac{\alpha_{C-j} + E\left(N_{C-j}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)}{\alpha_C + E\left(N_C|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)} \\ &\quad \times \prod_{i=1}^{n} \frac{\alpha_{ijk} + E\left(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)}{\alpha_{ij} + E\left(N_{ij}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)} \\ &= \tilde{\theta}^S_{C-j} \prod_{i=1}^{n} \tilde{\theta}^S_{ijk} \end{aligned} \tag{7}$$

where $\tilde{\theta}^S_{ijk}$ is the MAP parameter configuration for S, $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, $E(N_{ij}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}) = \sum_{k=1}^{r_i} E(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$ and similarly for the values related to C.

Note that S is a specific selective naive Bayes structure that represents the dependence between variables. If X_i is independent of C, in (7) we should use $E(N_{i-k}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$ instead of $E(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$.

Considering that the structure is not fixed *a priori*, we should average over all selective naive Bayes models in the following way:

$$p\left(c, \boldsymbol{x}|D^{(t)}\right) = \sum_S \int p\left(c, \boldsymbol{x}|S, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}|S, D^{(t)}\right) d\boldsymbol{\theta}\, p\left(S|D^{(t)}\right). \tag{8}$$

Therefore, the MA calculations require a summation over 2^n terms, which are the 2^n selective naive Bayes structures with n predictive variables.

Using the previous calculations for a fixed structure, (8) can be written as

$$\begin{aligned} p\left(c, \boldsymbol{x}|D^{(t)}\right) &\approx \sum_S \tilde{\theta}^S_{C-j} \prod_{i=1}^{n} \tilde{\theta}^S_{ijk} p\left(S|D^{(t)}\right) \\ &\propto \sum_S \tilde{\theta}^S_{C-j} \prod_{i=1}^{n} \tilde{\theta}^S_{ijk} p\left(D^{(t)}|S\right) p(S). \end{aligned} \tag{9}$$

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 [23] to our problem with missing values. Thus, we have an approximation to $p(D^{(t)}|S)$

$$\begin{aligned} p\left(D^{(t)}|S\right) \approx &\frac{\Gamma(\alpha_C)}{\Gamma\left(\alpha_C + E\left(N_C|\boldsymbol{\theta}^{(t)}, S\right)\right)} \\ &\times \prod_{j=1}^{r_C} \frac{\Gamma\left(\alpha_{C-j} + E\left(N_{C-j}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)\right)}{\Gamma(\alpha_{C-j})} \\ &\times \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma\left(\alpha_{ij} + E\left(N_{ij}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)\right)} \\ &\times \prod_{k=1}^{r_i} \frac{\Gamma\left(\alpha_{ijk} + E\left(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)\right)}{\Gamma(\alpha_{ijk})}. \end{aligned}$$

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

$$p\left(c, \boldsymbol{x}|D^{(t)}\right) \approx \kappa \sum_S \rho^S_{C-j} \prod_{i=1}^{n} \rho^S_{ijk} \tag{10}$$

where κ is a constant and ρ^S_{C-j} and ρ^S_{ijk} are defined as

$$\begin{aligned} \rho^S_{C-j} &= \tilde{\theta}^S_{C-j} p_S(C) \frac{\Gamma(\alpha_C)}{\Gamma\left(\alpha_C + E\left(N_C|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)\right)} \\ &\quad \times \prod_{j=1}^{r_C} \frac{\Gamma\left(\alpha_{C-j} + E\left(N_{C-j}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)\right)}{\Gamma(\alpha_{C-j})} \\ \rho^S_{ijk} &= \tilde{\theta}^S_{ijk} p_S(X_i, \mathrm{Pa}_i) \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma\left(\alpha_{ij} + E\left(N_{ij}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)\right)} \\ &\quad \times \prod_{k=1}^{r_i} \frac{\Gamma\left(\alpha_{ijk} + E\left(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}}\right)\right)}{\Gamma(\alpha_{ijk})}. \end{aligned} \tag{11}$$

Since we are assuming parameter independence and structure modularity, the calculations for ρ^S_{ijk} only depend on X_i and Pa_i. Therefore, if two different structures S_1 and S_2 represent the same relationship between X_i and C, $\rho^{S_1}_{ijk}$ will be the same as $\rho^{S_2}_{ijk}$. Hence, for all the selective naive Bayes models we only need to calculate ρ_{ijk} (if X_i is dependent on C) and ρ_{i-k} (if X_i is independent of C). The value ρ_{ijk} is calculated as shown in (11), and ρ_{i-k} is calculated as ρ_{ijk} but using $E(N_{i-k}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$ and α_{i-k}. Thus, (10) can be written [20] in terms of ρ_{i-k} and ρ_{ijk} as follows:

$$p\left(c, \boldsymbol{x}|D^{(t)}\right) \approx \rho_{C-j} \prod_{i=1}^{n} (\rho_{i-k} + \rho_{ijk}). \tag{12}$$

Note that after the transformations described above and once ρ_{ijk}, ρ_{i-k}, and ρ_{C-j} terms have been calculated, the expression $p(c, \boldsymbol{x}|D^{(t)})$ that required a $O(2^n)$ time, can now be evaluated in $O(n)$ time.

In order to calculate the ρ_{ijk}, ρ_{i-k}, and ρ_{C-j} terms, we need to set Dirichlet priors α_{C-j}, α_{i-k}, α_{ijk} and priors over structure $p_S(C)$, $p_S(X_i, \mathrm{Pa}_i)$ for all S and $i = 1, \ldots, n$. The values for all these priors are assumed to be known. Furthermore, we need the expected sufficient statistics $E(N_{C-j}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$ and $E(N_{ijk}|\boldsymbol{\theta}^{(t)}, S_{\mathrm{nB}})$ which have been calculated in the previous E step in $O(r_C \cdot n \cdot N)$ time.

Now, taking into account the factorization of the joint probability for a naive Bayes model

$$p(c, \boldsymbol{x}|\boldsymbol{\theta}, S_{\mathrm{nB}}) = \theta_C \prod_{i=1}^{n} \theta_{ijk} \tag{13}$$

similarity with the above-described (12) is observable.

Indeed, we can calculate the parameters $\boldsymbol{\theta}^{(t+1)}$ for a unique naive Bayes model, which approximates a Bayesian MA of selective naive Bayes for clustering as follows:

$$\begin{aligned}\theta_{ijk}^{(t+1)} &\propto (\rho_{i-k} + \rho_{ijk}) \\ \theta_{C-j}^{(t+1)} &\propto \rho_{C-j}\end{aligned} \tag{14}$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, r_C$.

The parameters for a Bayesian MA classifier are calculated in $O(n \cdot N \cdot r_{\max} \cdot r_C^2)$ time, where $r_{\max} = \max_{1 \leq i \leq n} r_i$. Note that the increment in time complexity in relation to the time needed by the EM algorithm to calculate the ML or MAP parameters, $O(n \cdot N \cdot r_C)$, is insignificant.

Remember that the EMA algorithm calculates the naive Bayes model for clustering iteratively. Therefore, a new naive Bayes for clustering is estimated at each iteration. Hence, the real-time complexity of the algorithm is $O(n \cdot N \cdot r_{\max} \cdot r_C^2 \cdot \mathrm{It})$ where It is the number of iterations of the EMA algorithm. The EMA algorithm depends on the random initialization of the parameters for the first naive Bayes model, so the total number of iterations may change each time the EMA algorithm is run.

IV. Testing EMA Versus Brute Force

The EMA algorithm is an approximation to Bayesian MA of selective naive Bayes because the existence of a latent cluster variable prevents an exact resolution in closed form of both the averaging over parameters and the marginal likelihood. Actually, only approximations are feasible.

The aim of this section is to demonstrate that the approximation to Bayesian MA given by the EMA algorithm is comparable to other more expensive and accurate techniques. Therefore, the EMA model is compared to a model obtained by a brute force approach where the averaging over parameters is also approximated by the MAP configuration but the marginal likelihood is calculated by Gibbs sampling [26]. This brute force method learns the 2^n selective naive Bayes models from the dataset and then averages them over weighted by their posterior probabilities in order to obtain the final model. In order to compare both EMA and brute force models, we propose a comparison test based on Monte Carlo techniques [27], [28].

Both models for unsupervised classification are estimated from the same dataset. This dataset is sampled from a random selective naive Bayes model.

Since it is computationally very expensive to construct a classifier by means of a brute force approach, the comparison between the EMA and brute force methods has only been performed for unsupervised classification models with ten and 12 predictive dichotomic variables. The cluster variable is also considered to take only two possible values. On the other hand, the datasets used for the experiments contain 300 samples ($N = 300$). These are not very large datasets, but the number of samples should be high enough to learn the models.

In order to learn the unsupervised model for clustering with both the EMA algorithm and the brute force method, it is needed to set the priors over structures and the hyperparameters. Usually, there is no explicit information about them. Therefore, for the experiment, we choose noninformative values for those parameters: $\alpha_{ijk} = 1, \alpha_{i-k} = 1, \alpha_{C-j} = 1$ and $p_S(X_i, \mathrm{Pa}_i) = 1, p_S(C) = 1$ for all i, j, k.

Each one of the 2^n models for the brute force approach is learned by obtaining an approximation for its MAP parameters using the EM algorithm. Since the EM is a greedy algorithm, we use a multistart EM.[1] The more times we run the EM algorithm, the more reliable the results are, but we have to find an agreement between efficiency and reliability. In our experiments, the multistart EM runs the EM algorithm 30 times ($m = 30$) to learn each one of the 2^n selective naive Bayes models. Finally, the brute force model is calculated as an average over the 2^n selective naive Bayes models weighted by the posterior probability for the structure of that model, $p(S|D) \propto p(D|S)p(S)$.

The exact calculation for $p(D|S)$, in a problem with missing values, is also intractable [23]. Since we attempt to compute a reference model to be compared with the model obtained by the EMA algorithm, the approximation to $p(D|S)$ must be as accurate as possible. The most accurate approximations, but also the most time-consuming ones, are the ones obtained by Monte Carlo methods. In this experiment, the approximation to $p(D|S)$ is given by the Candidate method, [29] which is an approximation for the marginal likelihood based on Bayes' theorem and Gibbs sampling [26].

The EMA, like the EM, is a greedy algorithm. Therefore, we also run a multistart EMA with $m = 30$. Since the EMA algorithm, in contrast to the EM algorithm, does not maximize the log-likelihood score, we decide to maintain a Bayesian methodology and obtain the final model of the multistart EMA algorithm by averaging over the m calculated models. Hence, the contribution of each one of these models to the final one is proportional to its likelihood score.

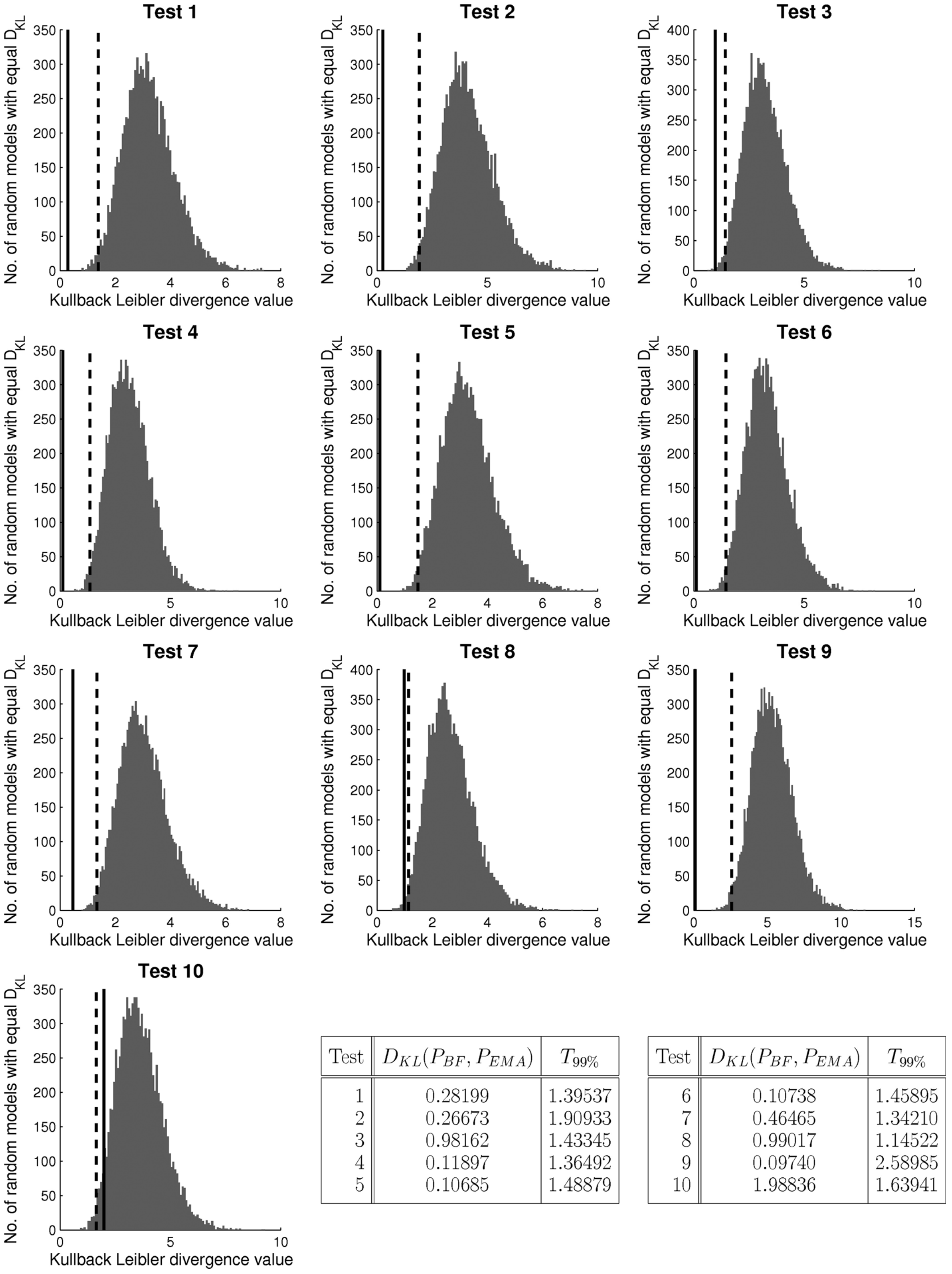
Once the brute force and the EMA models are obtained, we measure how different they are. This measure is given by the well-known Kullback–Leibler divergence, which is denoted by the following formula [30]:

$$D_{\mathrm{KL}}(P_{\mathrm{BF}}, P_{\mathrm{EMA}}) = \sum_{c,\boldsymbol{x}} p_{\mathrm{BF}}(c, \boldsymbol{x}) \log_2 \frac{p_{\mathrm{BF}}(c, \boldsymbol{x})}{p_{\mathrm{EMA}}(c, \boldsymbol{x})} \tag{15}$$

where P_{BF} is the probability mass function estimated with a brute force approach and P_{EMA} is the one estimated with the EMA algorithm. This divergence indicates how similar P_{EMA} is with respect to P_{BF}.

In order to know if the difference between both models is significant, the probability distribution of D_{KL} is needed. This is simulated by sampling a large number of random naive Bayes models and measuring their Kullback–Leibler divergence in relation to P_{BF}. In our case, we take 10 000 naive Bayes models with random parameters. We think this is a large enough number of models and it does not require excessive computational time. Thus, we can test if both probability distributions P_{EMA} and P_{BF} are close to each other.

[1] The EM algorithm is run m times, and the best model among the m runs in terms of log likelihood is selected.



Test	$D_{KL}(P_{BF}, P_{EMA})$	$T_{99\%}$
1	0.28199	1.39537
2	0.26673	1.90933
3	0.98162	1.43345
4	0.11897	1.36492
5	0.10685	1.48879

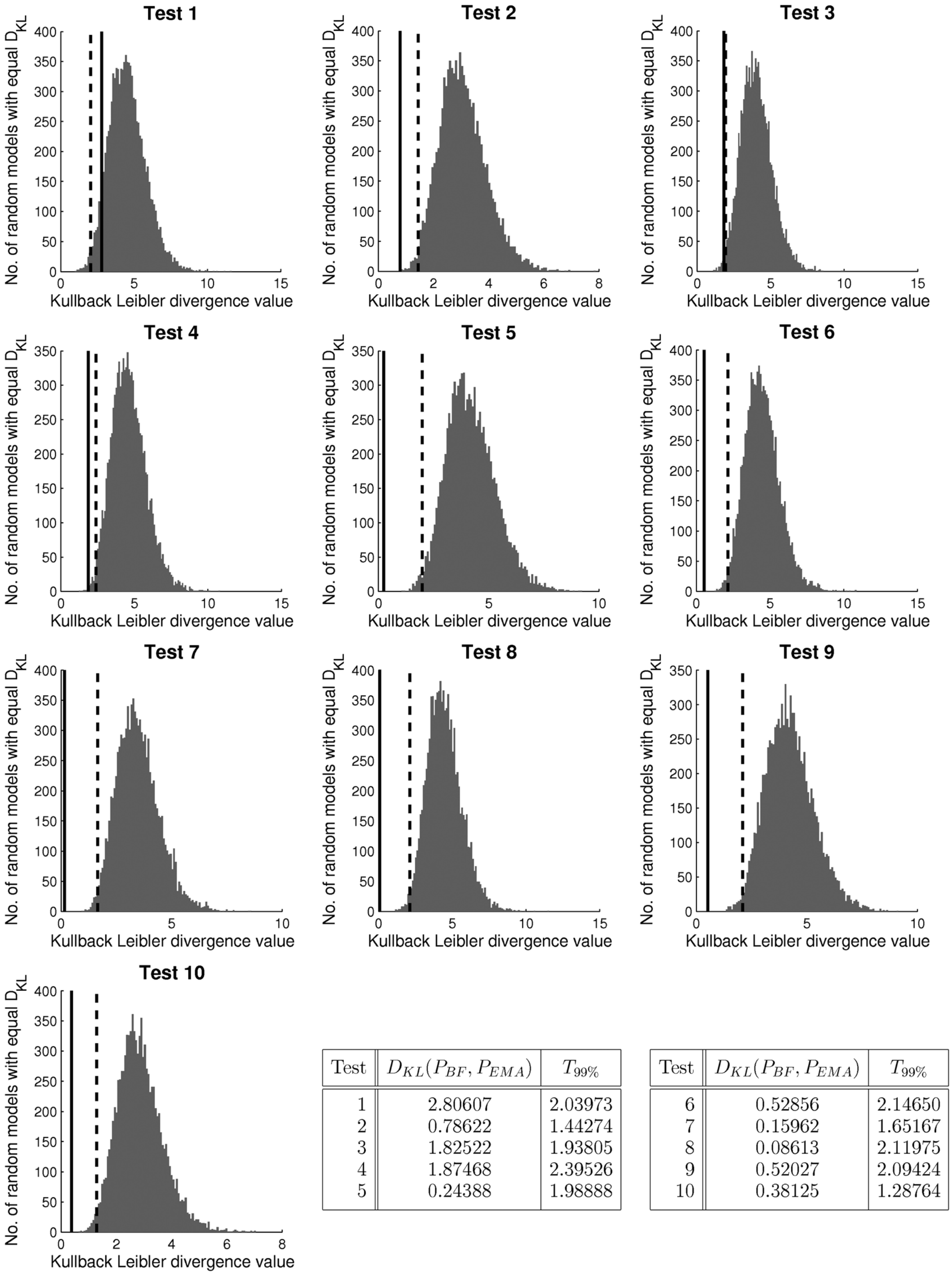
Test	$D_{KL}(P_{BF}, P_{EMA})$	$T_{99\%}$
6	0.10738	1.45895
7	0.46465	1.34210
8	0.99017	1.14522
9	0.09740	2.58985
10	1.98836	1.63941

Fig. 3. Tests for models with ten predictive variables. The dashed line represents the test value at the 99% level, $T_{99\%}$, and the solid line represents the distance between the EMA and brute force models, $D_{\mathrm{KL}}(P_{\mathrm{BF}}, P_{\mathrm{EMA}})$. These same values are given in the tables at the bottom of the figure.

The experiment described above depends on the random initializations for the EM and EMA algorithms. Therefore, ten independent tests (for models with ten and twelve predictive variables) have been performed. The results of these tests are shown in Figs. 3 and 4, respectively.

The results of the tests shown that, in all of them but two, the EMA is closer to the brute force model than 99% of the random generated models (test value at the 99% level, $T_{99\%}$). In fact, only in test 10 from Fig. 3 and test 1 from Fig. 4, $D_{\mathrm{KL}}(P_{\mathrm{BF}}, P_{\mathrm{EMA}})$ is bigger than the test value $T_{99\%}$. However, the $D_{\mathrm{KL}}(P_{\mathrm{BF}}, P_{\mathrm{EMA}})$ value is very close to $T_{99\%}$, and in both tests it is smaller than a test value at the 95% and 90% levels, respectively.

The result of the test for each model can be considered a random variable, which follows a binomial distribution $B(10, 0.01)$. In the experiments, for each model, we performed ten independent tests and in, at least, nine of them the EMA model is closer to the brute force model than the 99% of the random generated models. The probability of these results is $p(B(10, 0.01) \geq 9) = 10^{-18}$. This is such a small probability



Test	$D_{KL}(P_{BF}, P_{EMA})$	$T_{99\%}$
1	2.80607	2.03973
2	0.78622	1.44274
3	1.82522	1.93805
4	1.87468	2.39526
5	0.24388	1.98888

Test	$D_{KL}(P_{BF}, P_{EMA})$	$T_{99\%}$
6	0.52856	2.14650
7	0.15962	1.65167
8	0.08613	2.11975
9	0.52027	2.09424
10	0.38125	1.28764

Fig. 4. Tests for models with 12 predictive variables. The dashed line represents the test value at the 99% level, $T_{99\%}$, and the solid line represents the distance between the EMA and brute force models, $D_{\mathrm{KL}}(P_{\mathrm{BF}}, P_{\mathrm{EMA}})$. These same values are given in the tables at the bottom of the figure.

that it would be very unlikely to obtain the results shown before if the P_{EMA} and P_{BF} models were not close to each other.

V. Test for Model Detection

The EMA algorithm is learned by averaging over the MAP parameter configuration for all selective naive Bayes models. Thus, the higher $P(S|D)$ is, the more the model with structure S contributes to the model learned by the EMA. Therefore, if we sample a dataset D from a selective naive Bayes model with structure S, this is supposed to be the structure with the highest $P(S|D)$, that is, the MAP structure. Moreover, as the size of the dataset increases, the peak of the posterior probability density function of the structures becomes sharper at the MAP structure. Consequently, as the size of D increases, the MAP model, which is supposed to produce the dataset, makes a higher contribution to the EMA model and thus, the difference between this model and the model used to generate D may decrease.

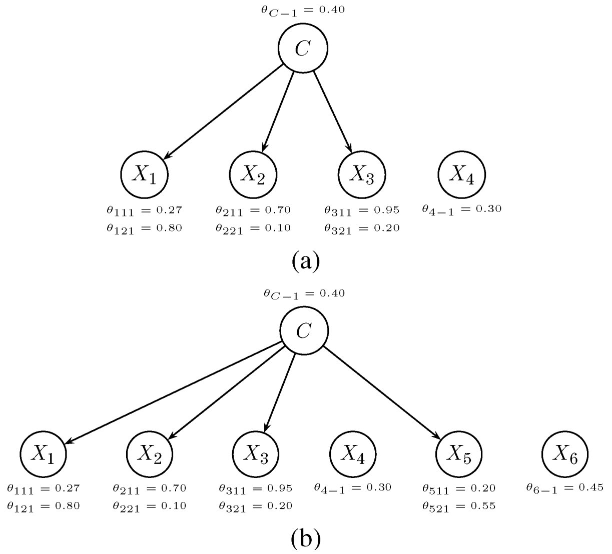


Fig. 5. Models used in the experiment. (a) Model with four predictive variables where X_4 is independent of C. (b) Model with six predictive variables where X_4 and X_6 are independent of C.

Following the idea given above, this section shows how a model learned from a dataset by using the EMA algorithm is able to detect the independencies between variables, which are described by the model used to generate the dataset. Thus, more empirical evidence about the good performance of the algorithm is provided.

In order to perform the test, a dataset is sampled from a selective naive Bayes model where some of the variables are independent of C. The model learned via the EMA algorithm should reveal these independencies between predictive variables and C. However, as the independence between variables is not explicitly given in the EMA model, a measure of independence for the variables is needed. This measure of independence is obtained using the Kullback–Leibler divergence between $p(X_i)$ and $p(X_i|c^j)$, and it is computed as follows:

$$I_P(X_i) = \frac{\sum_{j=1}^{r_C} D_{\mathrm{KL}}\left(p(X_i), p(X_i|c^j)\right)}{r_C}. \tag{16}$$

This measure of independence can be used to rank the predictive variables in terms of how independent of the cluster variable they are. Thus, it is possible to see which predictive variables are more likely to be independent of C. Moreover, comparing the measures of independence for all the predictive variables in the EMA model, the variables which are independent of C in the model used to sample the dataset should obtain smaller values of $I_{P_{\mathrm{EMA}}}(X_i)$.

In this test, two different models, which are shown in Fig. 5, are used to generate the datasets. The parameters for both models have been selected in such a way that the probability distributions of $p(c^0, X_i)$ and $p(c^1, X_i)$, for those X_i which are dependent on C, are not too closed to prevent the EMA algorithm from detecting these dependencies in the dataset. Thus, the models are sampled to generate datasets with different numbers of samples, and an EMA model is learned from each one of the datasets.

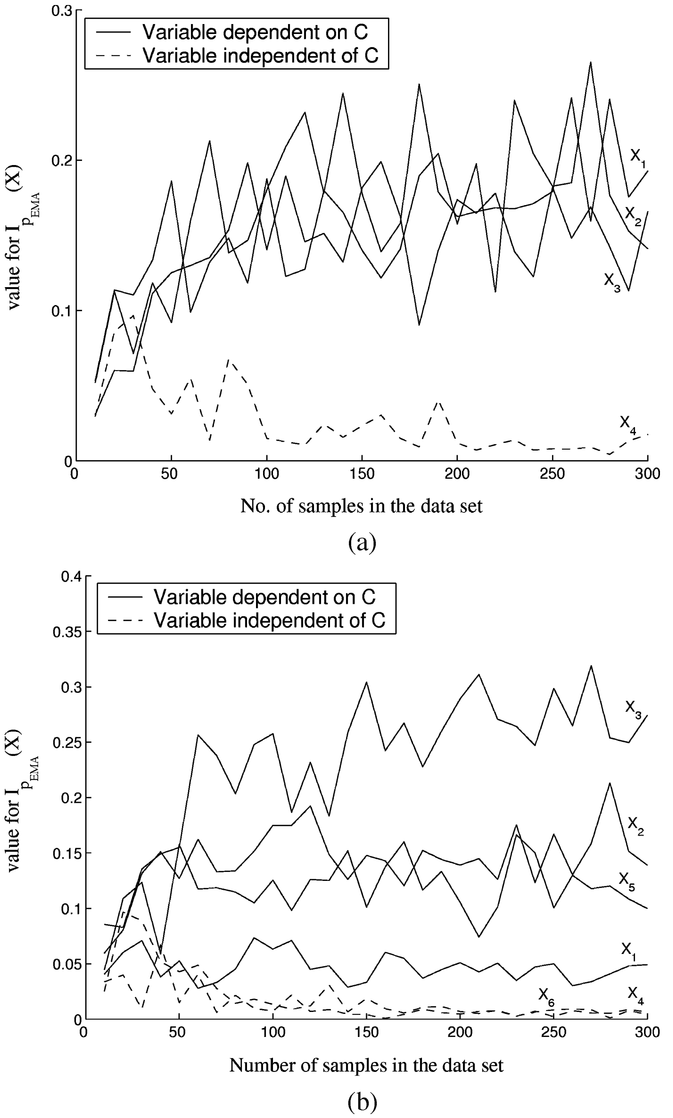


Fig. 6. Test for model detection. (a) Test with a model with four predictive variables where X_4 is independent of C. (b) Test with a model with six predictive variables where X_4 and X_6 are independent of C.

The value of $I_{P_{\mathrm{EMA}}}(X_i)$ in a model learned by the EMA is sometimes quite noisy due to the random initialization of the parameters. Therefore, the measure of independence is given by the mean of $I_{P_{\mathrm{EMA}}}(X_i)$ over a set of models learned via the EMA algorithm. Specifically, we have run the EMA algorithm 30 times in order to obtain thirty different models. Then, $I_{P_{\mathrm{EMA}}}(X_i)$ is given by the mean of the measures of independence of X_i over the 30 models.

The results of the tests are shown in Fig. 6. We can see in this figure that, as the size of the dataset increases, the difference between $I_{P_{\mathrm{EMA}}}(X_i)$ for the variables dependent on and independent of C also increases. Consequently, the EMA algorithm is able to detect the independencies between variables revealed in the dataset, and the larger the dataset, the better the EMA algorithm detects the model that generated the data.

TABLE I
COMPARISON OF EMA AND EM CLUSTERING METHODS IN DATASETS GENERATED BY SELECTIVE NAIVE BAYES MODELS. EACH POSITION OF THE TABLE INDICATES THE NUMBER OF WINS/DRAWS/LOSSES OF THE EMA MODELS WITH RESPECT TO THE EM MODELS FOR A SPECIFIC MODEL CONFIGURATION AND WITHIN A DATASET SIZE OVER 50 EXPERIMENTS. STATISTICALLY SIGNIFICANT DIFFERENCES AT 1% AND 10% LEVEL ARE DENOTED BY † AND ‡ RESPECTIVELY

#Var	#C	10	20	40	80	160	320	640
4	2	**24**/17/9‡	**24**/18/8‡	**19**/19/12†	9/25/**16**	7/22/**22**‡	4/22/**24**‡	12/25/**13**
6	2	**35**/6/9‡	**34**/5/11‡	**29**/16/5‡	**23**/19/8‡	15/17/**18**	11/13/**26**‡	15/13/**22**‡
8	2	**37**/7/6‡	**36**/6/8‡	**26**/11/13‡	**27**/15/8‡	**24**/14/12‡	**19**/14/17	13/11/**26**‡
10	2	**33**/7/10‡	**30**/9/11‡	**30**/10/10‡	**28**/9/13‡	**30**/7/13‡	**22**/9/19	**20**/11/19
20	2	**33**/7/10‡	**24**/17/9‡	**25**/20/5‡	**21**/17/12‡	**24**/23/3‡	**22**/23/5‡	**19**/23/8‡
40	2	**31**/17/2‡	**24**/22/4‡	**23**/22/5‡	**20**/27/3‡	**16**/31/3‡	**15**/34/1‡	**17**/32/1‡
4	3	**24**/10/16†	**22**/7/21	**25**/6/19†	**26**/11/14†	**23**/7/20†	**26**/9/15†	20/6/**24**
6	3	**31**/3/16‡	**34**/2/14‡	**36**/1/13‡	**29**/2/19‡	**33**/5/12‡	**31**/5/14‡	**26**/2/22
8	3	**39**/1/10‡	**38**/0/12‡	**40**/0/10‡	**42**/0/8‡	**37**/0/13‡	**33**/1/16‡	**26**/2/22
10	3	**32**/1/17‡	**29**/0/21‡	**34**/1/15‡	**35**/1/14‡	**34**/0/16‡	**34**/3/13‡	**29**/2/19†
20	3	**31**/3/16‡	**35**/1/14‡	**38**/1/11‡	**40**/4/6‡	**36**/3/11‡	**29**/3/18†	21/6/**23**
40	3	**41**/1/8‡	**44**/3/3‡	**44**/2/4‡	**36**/5/9‡	**34**/12/4‡	**27**/11/12‡	**25**/13/12†

VI. 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. On the other hand, it is possible to use other datasets, such as the ones coming from supervised learning problems, where the true cluster label is also known. This way, the cluster labeling obtained with a cluster algorithm can be compared with the real data partition. In this section, both approaches are taken into consideration in order to evaluate the EMA algorithm in clustering problems.

A. Synthetic Data

In order to illustrate the behavior of the EMA algorithm compared to the classical EM algorithm, we perform an exhaustive test of both algorithms using datasets sampled from randomly generated models, which are obtained by using a modification of the BNGenerator program [31].

For a first evaluation of the EMA algorithm in clustering problems, we obtain random selective naive Bayes models where the number of predictive variables vary in {4,6,8,10, 20,40}, the number of clusters in {2,3}, and each predictive variable can take up to five states. For each selective naive Bayes configuration, we generate 50 random models and each one of these models are sampled to obtain different datasets of sizes 10, 20, 40, 80, 160, 320, and 640. Multistart EM and multistart EMA algorithms with $m = 30$ are used to learn the EM and EMA models from each dataset. These models are used to cluster the dataset from which they have been learned. Afterwards, the winner model is determined by comparing the data partition obtained by the EM and EMA models with the true partition of the dataset. As the data partitioning done by clustering methods is sensitive to aliasing (two partitions can be the same but with different cluster labeling), we develop a comparison method insensitive to cluster labeling. This method consists in, for each data partition, obtaining its cluster matrix, that is, a $N \times N$ matrix, A, where a_{ij} with $i = 1, \ldots, N$ and $j = 1, \ldots, N$ is 1 if the ith and jth data samples are classified in the same cluster and 0 otherwise. Thus, we can obtain the cluster matrix for the true cluster labels, A_{Real}, and for the cluster labeling obtained by the EMA and EM models, A_{EMA} and A_{EM}, respectively. Hence, we can test which partition, EMA or EM, is closer to the real partition by comparing the Hamming distance between A_{Real} and $A_{\mathrm{EMA}} - D_{\mathrm{H}}(A_{\mathrm{Real}}, A_{\mathrm{EMA}})-$ and between A_{Real} and $A_{\mathrm{EM}} - D_{\mathrm{H}}(A_{\mathrm{Real}}, A_{\mathrm{EM}})-$.

In Table I, the results from the experiments with random selective naive Bayes models are shown. For each model configuration, the table describes the number of wins/draws/losses of the EMA models with respect to the EM model in relation to the Hamming distances between the models' cluster matrix and A_{Real}. We also provide information about a Wilcoxon signed-rank test used to evaluate whether the Hamming distances $D_{\mathrm{H}}(A_{\mathrm{Real}}, A_{\mathrm{EMA}})$ and $D_{\mathrm{H}}(A_{\mathrm{Real}}, A_{\mathrm{EM}})$ are different at the 1% and 10% levels (marked in the table with † and ‡, respectively). It can be seen that, in general, the EMA algorithm behaves better than the EM algorithm and the differences between the Hamming distances are, in most of the cases, statistically significant. However, in the biggest datasets, the differences between EMA and EM become smaller. In fact, in some simple models, for example the model with four variables and two clusters, and the model with six variables and two clusters, the EM algorithm beats the EMA algorithm when the datasets used to learn the models have a high enough number of samples. We hypothesize that this is because, as the sample size of the dataset increases, the posterior distributions over structures and parameters become sharper, tending to a Kronecker delta function at their MAP configuration. With enough data samples, the EM algorithm is able to approximate this MAP model. In contrast, the EMA algorithm averages over all the possible models and, even when the MAP model contributes the most to the final model, there are other less probable models with smaller contributions that, all together, may add noise to the final model.

Note that the experiments described above only use selective naive Bayes models since the EMA algorithm is based on these kinds of models. Naive Bayes and similarly selective naive Bayes are quite simple models and the restrictions that they

TABLE II
COMPARISON OF EMA AND EM CLUSTERING METHODS IN DATASETS GENERATED BY GENERAL BAYESIAN NETWORK CLASSIFIERS. EACH POSITION OF THE TABLE INDICATES THE NUMBER OF WINS/DRAWS/LOSSES OF THE EMA MODELS WITH RESPECT TO THE EM MODELS FOR A SPECIFIC MODEL CONFIGURATION AND WITHIN A DATASET SIZE OVER 50 EXPERIMENTS. STATISTICALLY SIGNIFICANT DIFFERENCES AT 1% AND 10% LEVEL ARE DENOTED BY † AND ‡ RESPECTIVELY

#Var	#C	#Pa	10	20	40	80	160	320	640
4	2	2	**35**/11/4‡	**25**/22/3‡	**13**/29/8†	**13**/27/10	12/26/12	**18**/18/14	**17**/19/14†
6	2	2	**31**/11/8‡	**23**/21/6‡	**14**/23/13	**17**/23/10†	12/21/**17**	17/15/**18**	10/21/**19**
8	2	2	**30**/11/9‡	**21**/22/7‡	12/26/12	14/21/**15**	15/16/**19**	13/16/**21**†	**19**/16/15
10	2	2	**27**/18/5‡	14/19/**17**	**20**/23/7‡	**18**/19/13†	13/18/**19**	**18**/20/12†	**20**/15/15
20	2	2	**19**/22/9‡	14/21/**15**	**19**/13/18	**27**/11/12†	**24**/14/12†	20/8/**22**	**21**/9/20
40	2	2	**16**/19/15	16/15/**19**	**20**/17/13	17/13/**20**	20/5/**25**	**22**/10/18	**26**/7/17
4	2	3	**31**/12/7‡	**20**/24/6‡	**18**/26/6‡	**15**/24/11	12/20/**18**	**16**/22/12	**13**/26/12
6	2	3	**36**/9/5‡	**22**/17/11‡	**15**/21/14	**16**/24/10	**18**/18/14†	**23**/18/9	**21**/12/17
8	2	3	**34**/12/4‡	**19**/21/10‡	11/26/**13**	13/23/**14**	**17**/23/10†	17/16/17	12/13/**25**
10	2	3	**24**/13/13‡	**18**/18/14	**21**/20/9†	16/15/**19**	16/15/**19**	**24**/6/20	**22**/11/17
20	2	3	**13**/26/11	**24**/13/13†	**20**/16/14	**20**/12/18	20/10/20	17/11/**22**	10/13/**27**†
40	2	3	**17**/20/13	**21**/9/20	18/6/**26**	**23**/9/18	20/6/**24**	**23**/7/20	18/6/**26**
4	3	2	**27**/7/16‡	**30**/4/16‡	**28**/5/17†	**35**/1/14‡	**25**/3/23	22/2/**26**	22/3/**26**
6	3	2	**36**/1/13‡	**34**/3/13‡	**31**/1/18‡	**30**/8/12‡	**29**/6/15†	**32**/5/13†	**27**/6/17†
8	3	2	**33**/4/13‡	**32**/0/18‡	**39**/1/10‡	**28**/3/19†	**24**/5/21	**32**/0/18‡	**27**/2/21
10	3	2	**30**/3/17‡	**27**/3/20†	**29**/3/18	**27**/4/19	**27**/4/19	**28**/4/18	**26**/3/21
20	3	2	**32**/0/18†	**35**/2/13‡	**36**/2/12‡	**32**/2/16†	**29**/2/19	**30**/0/20	**32**/1/17†
40	3	2	**26**/8/16†	23/2/**25**	**26**/2/22†	**26**/2/22	**26**/0/24	**32**/2/16†	**29**/1/20
4	3	3	**30**/8/12‡	**35**/3/12‡	**29**/5/16‡	**23**/8/19	**24**/3/23	**29**/3/18†	23/4/23
6	3	3	**38**/3/9‡	**35**/3/12‡	**31**/2/17†	17/9/**24**	**24**/9/17	**27**/8/15†	21/5/**24**
8	3	3	**30**/1/19‡	**28**/4/18‡	**27**/5/18†	22/4/**24**	**25**/3/22	23/3/**24**	21/1/**28**
10	3	3	**35**/2/13‡	**33**/3/14‡	**29**/3/18†	**32**/4/14†	**30**/2/18†	**30**/3/17	**28**/0/22†
20	3	3	**31**/5/14‡	**36**/4/10‡	**29**/1/20†	**31**/5/14‡	**31**/2/17	23/2/**25**	**26**/1/23
40	3	3	21/7/**22**	**22**/7/21	24/1/**25**	**28**/0/22	24/2/24	24/0/**26**	22/0/**28**

present are not usually fulfilled in real problems. However, these models have been widely and successfully used in the literature [6], [7], [32], [33] applied to different problems even if the naive Bayes conditions are not satisfied. We would like to illustrate the behavior of the EMA algorithm in more realistic problems. Therefore, we repeat the experiment using more complicated models. In this case, we do not restrict the models to the selective naive Bayes family, but we use general Bayesian network classifiers. That is, we randomly generate Bayesian networks varying the number of variables in $\{4,6,8,10,20,40\}$, and the maximum number of parents for each variable in $\{2,3\}$. Each variable can take up to five states. Then, we add the cluster variable, which can take two or three states, and decide randomly which predictive variables are dependent on it.

Table II shows the results for the experiments with Bayesian network classifiers. It can be seen that, when the Bayesian classifiers used to sample the dataset are quite complex, the behavior of both the EMA and the EM algorithms is very similar. This may be because what we learn with both EMA and EM is a naive Bayes model and the dataset contains too complex interrelations between variables to be modeled with the restrictions of a naive Bayes. Moreover, the dataset may not have enough samples to capture the complexity of the model that generated the data. On the other hand, except for the most complex models used in the experiments, the EMA algorithm performs better than the EM algorithm. Nevertheless, the differences in the Hamming distances of the cluster matrices for both models are statistically significant in only some experiments.

B. Deoxyribonucleic Acid (DNA) Microarray Data

Nowadays, it is very widespread to use DNA microarrays to monitorize the expression level of thousands of genes at the same time. Although the popularization of different microarray techniques has decreased the experimentation cost, it is still quite expensive. Therefore, microarray datasets normally contain thousands of variables (expression levels of genes) and only a few cases (experiments). Since MA techniques account for model uncertainty, they are preferable when only a few data samples are available, which is precisely the case of DNA microarray data. Recently, some MA methods have been proposed to deal with both supervised classification [34] and clustering [35], [36] in DNA microarray problems.

The famous acute myeloid leukemia (AML)/acute lymphoblastic leukemia (ALL) dataset from the Whitehead Institute [37] is one of the first problems that appear in the literature where machine learning techniques are used to classify data from DNA microarrays. The original dataset consists of 72 samples from patients diagnosed with ALL (47 samples), and AML (25 samples). Each one of these data samples contains the expression value of 7129 probes, corresponding to 6817 human genes, which were obtained by means of high-density oligonucleotide microarrays produced by Affymetrix.

The use of all the 7129 variables to learn a clustering model seems, in principle, nonsense because not all the variables in the original dataset are relevant for clustering purposes and they may blur the real data aggregation. Since the real class label of each data sample is known in this problem, Golub *et al.* [37] propose to filter out variables by selecting only the 50 most informative ones in relation to their correlation with the

TABLE III
ESTIMATED ACCURACY FOR CLUSTERING METHODS WITH LEUKEMIA DATASET

	Mean	Standard Deviation
multi-start EM	58.86	6.53
multi-start EMA (BC)	82.50	3.01
multi-start EMA (UA)	79.31	7.13
multi-start EMA (Av)	85.28	10.6
SOM	89.47	- -

class variable. This approximation is, in general, impossible for clustering problems since the true class label is unknown. By contrast, the EMA algorithm provides a powerful tool that integrates an implicit Bayesian variable selection in the learning process of the clustering model. Thus, although all the variables are included in the clustering model, only the relevant ones are taken into account when clustering the data.

For the experiment, we use all the 7129 variables and learn clustering models with both multistart EM and multistart EMA algorithms. In the case of the EMA algorithm, we develop three different policies of multistart in order to select the final model: best choice (BC), which selects the model with highest likelihood value; uniform averaging (UA), in which all the models calculated in the multistart process equally contribute to the final averaged model; and finally averaging (Av), which obtains the final model by averaging over all the models calculated in the multistart process and where the contribution of each model is proportional to its likelihood value. This last one is the policy we have so far used in the experiments.

Table III shows the estimated accuracy for the multistart EM and multistart EMA algorithms using, in both cases, $m = 100$. This estimated accuracy value correspond to the percentage of correct classified samples that can be calculated by comparing the cluster labeling obtained by the multistart EM and multistart EMA algorithms with the real cluster partition of the leukemia dataset which, in this specific problem, is known. The table also provides the accuracy value reported in Golub *et al.* [37], where self-organizing maps (SOMs) are used to cluster the leukemia dataset (the standard deviation value is not reported in the paper). Note that the results obtained by the EMA algorithm can compete with the ones obtained by Golub *et al.* even when the EMA algorithm uses all the 7129 variables, and the SOM, only the 50 most informative variables with respect to the class. That is, in this particular case, the EMA algorithm is able to detect the irrelevant variables in the problem and obtain a reasonable estimated accuracy value. In contrast, the multistart EM algorithm does not have enough data samples to estimate the MAP model and it may consider all the variables equally important for clustering purposes. Thus, the EM algorithm is influenced by irrelevant variables which leads it to obtain a low value for estimated accuracy.

VII. DISCUSSION

We have shown that it is possible to obtain a unique unsupervised naive Bayes classifier that approximates a MA over the MAP configurations for selective naive Bayes models. Furthermore, this approximation can be performed in the same time complexity needed to learn the ML or MAP parameters for a naive Bayes model with the EM algorithm. In order to do so, we introduced the EMA algorithm. This is an extension of the EM algorithm that makes feasible a Bayesian averaging approach to unsupervised classification with naive Bayes models. Moreover, we have provided empirical evidence on the fact that the approximation to MA obtained by the EMA algorithm is, actually, camparable to MA over MAP parameter configurations for selective naive Bayes. Additionally, the EMA algorithm is able to detect the structure of the model that has generated the data. Therefore, the method proposed in this paper can also be regarded as a Bayesian approach to unsupervised feature subset selection.

In the paper, we also performed an exhaustive test to illustrate the behavior of the EMA algorithm in clustering problems. We found that the EMA algorithm is a powerful learning algorithm that may be very useful for clustering problems where there are lots of variables (many of them probably irrelevant for clustering purposes) and only a few data samples.

Probably one of the limitations for the EMA algorithm is that the number of clusters is assumed to be known. This is not so usual in clustering problems. However, there are a lot of clustering algorithms (k-means, SOM, EM, etc.) with the same limitations. In the literature, we can find several proposals to overcome this problem. For instance, we can learn models with a number of clusters from $r_{C_{\min}}$ to $r_{C_{\max}}$ and select the best model according to a validity index [38]. Furthermore, there are some other techniques proposed to learn the dimensionality of a hidden variable in a Bayesian network classifier. For instance, [39] proposes a method based on the EM algorithm where the model starts with a maximal number of clusters, which are merged in a greedy fashion to obtain the final model. Additionally, [40] proposes another method also based on the EM algorithm, where it is not only possible to merge two states of the cluster variable but also to split a cluster into two new different states. It is very interesting, for future work, to use these methods in conjunction with the EMA algorithm to allow an automatic determination of the number of clusters.

The EMA algorithm attempts to iteratively approximate the Bayesian MA model. Hence, the distance between the model learned by EMA and the Bayesian MA model should decrease at each iteration of the algorithm. For future work, it would also be very interesting to demonstrate the monotonicity of the EMA in terms of how this distance decreases throughout the iterations of the algorithm. Additionally, another theoretical point of interest for future studies could be the derivation of an upper bound for the approximation error of the model provided by the EMA algorithm in terms of the dataset size and the number of parameters in the model.

This paper takes into consideration only selective naive Bayes structures, but the EMA algorithm can be extended to calculate more complex models. In fact, the calculations for Bayesian MA in supervised classification problems have already been extended to models such as the tree-augmented naive Bayes [18], [19], [41]. Thus, these calculations could also be extended to unsupervised classification problems.

More future work might include the use of the EMA algorithm with real-word clustering problems and the relaxation of the assumption of nonmissing data for predictive variables. The consideration of missing values for the predictive variables can be solved in the E step of the EMA algorithm, but it implies more complicated calculations when estimating the expected sufficient statistics.

Acknowledgment

The authors would like to thank the anonymous reviewers, to G. F. Cooper, and to J. Cerquides for their helpful comments, and to the Servicio General de Informática/Informatikaren Zerbitzu Orokorra-Servicios Generales de Investigación Universidad del País Vasco/Euskal Herriko Unibertsitatea (supported by the National Program for the Promotion of Human Resources within the National Plan of Scientific Research, Development and Innovation—Fondo Social Europeo, MCyT, and Basque Government) for their generous allocation of computational resources.

References

[1] J. D. Banfield and A. E. Raftery, "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, no. 3, pp. 803–821, Sep. 1993.

[2] H. Bensmail, G. Celeux, A. E. Raftery, and C. P. Robert, "Inference in model-based cluster analysis," *Statist. Comput.*, vol. 7, no. 1, pp. 1–10, Mar. 1997.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann, 1988.

[4] F. Jensen, *Bayesian Networks and Decision Graphs*. New York: Springer-Verlag, 2001.

[5] R. E. Neapolitan, *Learning Bayesian Networks*. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[6] P. Cheeseman and J. Stutz, "Bayesian classification (Autoclass): Theory and results," in *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press, 1996, pp. 153–180.

[7] Y. Barash and N. Friedman, "Context-specific Bayesian clustering for gene expression data," *J. Comput. Biol.*, vol. 9, no. 2, pp. 169–191, Apr. 2002.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. R. Statist. Soc., Ser. B* , vol. 39, no. 1, pp. 1–38, 1977.

[9] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.

[10] N. Friedman, "Learning belief networks in the presence of missing values and hidden variables," in *Proc. 14th Int. Conf. Mach. Learn.*, 1997, pp. 125–133.

[11] J. Peña, J. Lozano, and P. Larrañaga, "An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering," *Pattern Recognit. Lett.*, vol. 21, no. 8, pp. 779–786, Jul. 2000.

[12] D. Madigan and A. E. Raftery, "Model selection and accounting for model uncertainty in graphical models using Occam's window," *J. Amer. Statist. Assoc.*, vol. 89, no. 428, pp. 1535–1546, Dec. 1994.

[13] J. Hoeting, D. Madigan, A. E. Raftery, and C. Volinsky, "Bayesian model averaging," *Statist. Sci.*, vol. 14, no. 4, pp. 382–401, 1999.

[14] D. Heckerman, "A tutorial on learning with Bayesian networks," Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-95-06, 1995.

[15] N. Friedman, "The Bayesian structural EM algorithm," in *Proc. 14th Conf. Uncertainty Artif. Intell.*, 1998, pp. 129–138.

[16] N. Friedman and D. Koller, "Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks," *Mach. Learn.*, vol. 50, no. 1/2, pp. 95–126, Jan./Feb. 2003.

[17] J. Cerquides and R. López de Mántaras, "The indifferent naive Bayes classifier," in *Proc. 16th Int. FLAIRS Conf.*, 2003, pp. 341–345.

[18] D. Dash and G. F. Cooper, "Model averaging for prediction with discrete Bayesian networks," *J. Mach. Learn. Res.*, vol. 5, pp. 1177–1203, 2004.

[19] J. Cerquides and R. López de Mántaras, "TAN classifiers based on decomposable distributions," *Mach. Learn.*, vol. 59, no. 3, pp. 323–354, Jun. 2005.

[20] D. Dash and G. Cooper, "Exact model averaging with naive Bayesian classifiers," in *Proc. 19th Int. Conf. Mach. Learn.*, 2002, pp. 91–98.

[21] Elvira Consortium, "Elvira: An environment for probabilistic graphical models," in *Proc. 1st Eur. Workshop Probab. Graph. Models*, 2002, pp. 222–230.

[22] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA: Morgan Kaufmann, 2005.

[23] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, no. 4, pp. 309–347, Oct. 1992.

[24] B. Thiesson, "Score and information for recursive exponential models with incomplete data," in *Proc. 13th Annu. Conf. Uncertainty Artif. Intell.*, 1997, pp. 453–463.

[25] D. J. C. MacKay, "Choice of basis for Laplace approximation," *Mach. Learn.*, vol. 33, no. 1, pp. 77–86, Oct. 1998.

[26] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–742, Nov. 1984.

[27] Y. A. Shereider, *Method of Statistical Testing: Monte Carlo Method*. New York: Elsevier, 1964.

[28] I. M. Sobol, *The Monte Carlo Method*. Moscow, Russia: Mir Publishers, 1984.

[29] D. M. Chickering and D. Heckerman, "Efficient approximation for the marginal likelihood of Bayesian networks with hidden variables," *Mach. Learn.*, vol. 29, no. 2/3, pp. 181–212, Nov./Dec. 1997.

[30] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[31] J. S. Ide, F. G. Cozman, and F. Ramos, "Generating random Bayesian networks with constraints on induced width," in *Proc. 16th Eur. Conf. Artif. Intell.*, vol. 2004, pp. 323–327.

[32] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Mach. Learn.*, vol. 29, no. 2/3, pp. 103–130, 1997.

[33] D. Hand, K. You, "Idiot's Bayes—Not so stupid after all?" *Int. Statist. Rev.*, vol. 69, no. 3, pp. 385–398, Dec. 2001.

[34] K. Y. Yeung, R. E. Bumgarner, and A. E. Raftery, "Bayesian model averaging: Development of an improved multiclass, gene selection and classification tool for microarray data," *Bioinformatics*, vol. 21, no. 10, pp. 2394–2402, May 2005.

[35] M. Medvedovic and J. Guo, "Bayesian model-averaging in unsupervised learning from microarray data," in *Proc. 4th Workshop Data Mining Bioinformatics*, 2005, pp. 40–47.

[36] C. Vogl, F. Sanchez-Cabo, G. Stocker, S. Hubbard, O. Wolkenhauer, and Z. Trajanoski, "A fully Bayesian model to cluster gene-expression profiles," *Bioinformatics*, vol. 21 (Supplement 2), pp. ii130–ii135, 2005.

[37] T. R. Golub, D. K. Slonim, P. Tamayo, C. Juard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomgield, and E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, Oct. 1999.

[38] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 58, no. 2, pp. 159–179, Jun. 1985.

[39] G. Elidan and N. Friedman, "Learning the dimensionality of hidden variables," in *Proc. 17th Conf. Uncertainty Artif. Intell.*, 2001, pp. 144–151.

[40] G. Karciauskas, T. Kocka, F. V. Jensen, P. Larrañaga, and J. A. Lozano, "Learning of latent class models by splitting and merging components," in *Proc. 2nd Workshop Probab. Graph. Models*, 2004, pp. 137–144.

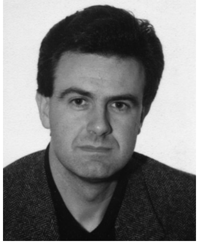
[41] D. Dash and G. Cooper, "Model averaging with discrete Bayesian network classifiers," in *Proc. 9th Int. Workshop Artif. Intell. Statist.*, 2003, pp. 38–45.

Guzmán Santafé received the M.S. degree in computer science from the University of the Basque Country, San Sebastián, Spain, in 2002. He is currently working toward the Ph.D. degree at the University of the Basque Country as a member of the Intelligent Systems Group.

His research interests include clustering, Bayesian model averaging, discriminative learning of probabilistic graphical models for classification, and machine learning techniques applied to bioinformatics.

Jose A. Lozano (M'04) received the M.S. degrees in mathematics and computer science and the Ph.D. degree from the University of the Basque Country, San Sebastián, Spain, in 1991, 1992, and 1998, respectively.

Since 1999, he has been an Associate Professor of computer science at the University of the Basque Country. He has edited three books and has published over 25 refereed journal papers. His main research interests are evolutionary computation, machine learning, probabilistic graphical models, and bioinformatics.



Pedro Larrañaga received the M.S. degree in mathematics from the University of Valladolid, Valladolid, Spain, in 1981, and the Ph.D. degree in computer science from the University of the Basque Country, San Sebastián, Spain, in 1995.

He is currently a Professor of computer science and artificial intelligence at the University of the Basque Country. He has published over 40 refereed journal papers. His main research interests are in the areas of evolutionary computation, machine learning, probabilistic graphical models, and bioinformatics.