
ELSEVIER International Journal of Approximate Reasoning 28 (2001) 23–50

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Performance evaluation of compromise conditional Gaussian networks for data clustering

J.M. Peña, J.A. Lozano, P. Larrañaga

Department of Computer Science and Artificial Intelligence, Intelligent Systems Group, University of Basque Country, P.O. Box 649, E-20080 Donostia – San Sebastián, Spain

Received 1 May 2000; accepted 1 April 2001

Abstract

This paper is devoted to the proposal of two classes of compromise conditional Gaussian networks for data clustering as well as to their experimental evaluation and comparison on synthetic and real-world databases. According to the reported results, the models show an ideal trade-off between efficiency and effectiveness, i.e., a balance between the cost of the unsupervised model learning process and the quality of the learnt models. Moreover, the proposed models are very appealing due to their closeness to human intuition and computational advantages for the unsupervised model induction process, while preserving a rich enough modeling power.

Keywords: Data clustering; Conditional Gaussian networks; Naive Bayes models; Tree augmented naive Bayes models; Extended naive Bayes models

1. Introduction

A basic problem that arises in a variety of fields, such as pattern recognition, machine learning, and statistics, is the so-called *data clustering problem* [1,2,7,8,16,18]. From the point of view adopted in this paper, the data clustering problem may be defined as the inference of a generalized

E-mail address: ccbpepaj@si.ehu.es (J.M. Peña).

0888-613X/01/$ - see front matter
PII: S0888-613X(01)00048-2

joint probability distribution from a database. We assume that, in addition to the observed random variables or predictive attributes, there is a hidden random variable. This last unobserved random variable would reflect the cluster membership for every case in the database. Thus, the clustering problem is also referred to as an example of learning from unlabeled data due to the existence of such a hidden random variable. In this paper, we focus on learning conditional Gaussian networks for data clustering.

Due to the difficulty involved in learning densely connected conditional Gaussian networks (even more when working with large databases in terms of both number of cases and number of random variables), and the painfully slow probabilistic inference when working with them, it is necessary to develop methods for learning simple conditional Gaussian networks while preserving the quality of the learnt models. A symmetrical problem is found when working in discrete domains, and some of the proposed solutions are based on achieving such a balance between efficiency and effectiveness [7,10–12,19,24,27–34].

Keeping this idea in mind, we propose two classes of compromise conditional Gaussian networks for data clustering. The models of the first class belong to what in [10–12,19] are called tree augmented naive Bayes models (more recently referred to as mixtures of trees with shared structure [24]). These models are used in the referred works as Bayesian classifiers, while in [31,33] they are evaluated in the framework of data clustering for discrete domains. In this work, we evaluate them with respect to the data clustering problem for continuous domains. Tree augmented naive Bayes models for data clustering are defined by the following condition: Predictive attributes may have at most one other predictive attribute as a parent. The second class of models that we present and evaluate with respect to the data clustering problem for continuous domains is proposed by Pazzani [27,28] as Bayesian classifiers. In [30,32] this class of models are applied to data clustering in discrete domains. These models are similar to naive Bayes models [7], which is why we name them extended naive Bayes models, with the exception that the number of nodes in the structures can be smaller than the original number of random variables in the database as some of them can be grouped together under the same node as fully correlated.

When applied to data clustering, the compromise conditional Gaussian networks proposed are a weaker representation of some domains than general conditional Gaussian networks for data clustering. However, our experimental results on both synthetic and real-world databases show that the performance and modeling power of these models is still significant, while they offer obvious advantages from the point of view of the cost of the unsupervised model learning process.

The remainder of this paper is organized as follows. In Section 2, we introduce general conditional Gaussian networks for data clustering. Section 3 is

dedicated to present in detail our proposal of compromise conditional Gaussian networks for data clustering. Some experimental results on several synthetic and real-world databases are compiled in Section 4. Finally, in Section 5 we draw conclusions.

2. Unsupervised learning of conditional Gaussian networks

This section starts introducing the notation used throughout this paper. Then, we give a formal definition of conditional Gaussian networks for data clustering. We also present the Bayesian structural EM algorithm [9], which is used for explanatory purposes as well as in our experiments due to its good performance in unsupervised learning of conditional Gaussian networks.

2.1. Notation

We follow the usual convention of denoting unidimensional random variables by uppercase letters and their states by the same letters in lowercase. We use a letter or letters in boldface uppercase to designate a multidimensional random variable and the same boldface lowercase letter or letters to denote an assignment of state to the multidimensional random variable. The generalized joint probability distribution for $\boldsymbol{X}$ is represented as $\rho(\boldsymbol{x})$. Additionally, $\rho(\boldsymbol{x}\,|\,\boldsymbol{y})$ denotes the generalized conditional probability distribution for $\boldsymbol{X}$ given $\boldsymbol{Y}=\boldsymbol{y}$. If $\boldsymbol{X}$ is a multidimensional discrete random variable, then $\rho(\boldsymbol{x})=p(\boldsymbol{x})$ is the joint probability mass function for $\boldsymbol{X}$. Thus, $p(\boldsymbol{x}\,|\,\boldsymbol{y})$ denotes the conditional probability mass function for $\boldsymbol{X}$ given $\boldsymbol{Y}=\boldsymbol{y}$. On the other hand, if $\boldsymbol{X}$ is a multidimensional continuous random variable, then $\rho(\boldsymbol{x})=f(\boldsymbol{x})$ is the joint probability density function for $\boldsymbol{X}$. Thus, $f(\boldsymbol{x}\,|\,\boldsymbol{y})$ denotes the conditional probability density function for $\boldsymbol{X}$ given $\boldsymbol{Y}=\boldsymbol{y}$.

2.2. Conditional Gaussian networks for data clustering

As already mentioned, when facing a data clustering problem we assume the existence of a $(n+1)$-dimensional mixed random variable $\boldsymbol{X}$ partitioned as $\boldsymbol{X}=(\boldsymbol{Y},C)$ into a n-dimensional continuous random variable $\boldsymbol{Y}$ and a unidimensional discrete *hidden* random variable C. It is usual to refer to the unidimensional random variables in $\boldsymbol{Y}$ as *observed random variables* or *predictive attributes*. When dealing with a data clustering problem, $\boldsymbol{X}$ is said to have a *conditional Gaussian distribution* [5,20–22] if the distribution for $\boldsymbol{Y}$, conditioned on each state of C, is a multivariate normal distribution. That is,

$$f(\boldsymbol{y}\,|\,C=c)=f_c(\boldsymbol{y})\equiv\mathcal{N}(\boldsymbol{y};\boldsymbol{\mu}(c),\Sigma(c)), \tag{1}$$

whenever $p(c) = p(C = c) > 0$. Given $C = c$, $\boldsymbol{\mu}(c)$ is the n-dimensional mean vector, and $\Sigma(c)$, the $n \times n$ variance matrix, is positive definite.

We define a conditional Gaussian network (CGN) for $\boldsymbol{X}$ for data clustering as a graphical model that encodes a conditional Gaussian distribution for $\boldsymbol{X}$. Essentially, CGNs for data clustering belong to a class of mixed probabilistic graphical models introduced for the first time by Lauritzen and Wermuth [22], and further developed in [20,21]. This class groups models in which both discrete and continuous random variables can be present and for which the conditional distribution for the continuous random variables given the discrete random variables is restricted to be multivariate Gaussian. More recently, CGNs have been successfully applied to data clustering [29,33].

Concretely, a CGN for $\boldsymbol{X}$ for data clustering is defined by a directed acyclic graph $\boldsymbol{s}$ (model structure) determining the conditional (in)dependencies among the predictive attributes in $\boldsymbol{Y}$, a set of local probability density functions for these predictive attributes, and a multinomial distribution for the cluster random variable C. The model structure yields to a graphical factorization of the generalized joint probability distribution for $\boldsymbol{X}$ as follows:

$$\rho(\boldsymbol{x}) = \rho(\boldsymbol{y}, c) = p(c) f(\boldsymbol{y} \mid c) = p(c) f_c(\boldsymbol{y}) = p(c) \prod_{i=1}^{n} f_c(y_i \mid \mathbf{pa}(\boldsymbol{s})_i), \tag{2}$$

where $\mathbf{pa}(\boldsymbol{s})_i$ denotes the configuration of the parents of Y_i, $\mathbf{Pa}(\boldsymbol{s})_i$, for all i. The local probability density functions for the predictive attributes and the multinomial distribution for the cluster random variable are those in the previous equation, and we assume that they depend on a finite set of parameters $\boldsymbol{\theta}_s \in \boldsymbol{\Theta}_s$. Therefore, Eq. (2) can be rewritten as follows:

$$\begin{aligned} \rho(\boldsymbol{x} \mid \boldsymbol{\theta}_s) &= \rho(\boldsymbol{y}, c \mid \boldsymbol{\theta}_s) \\ &= p(c \mid \boldsymbol{\theta}_s) f(\boldsymbol{y} \mid c, \boldsymbol{\theta}_s) \\ &= p(c \mid \boldsymbol{\theta}_s) f_c(\boldsymbol{y} \mid \boldsymbol{\theta}_s^c) \\ &= p(c \mid \boldsymbol{\theta}_s) \prod_{i=1}^{n} f_c(y_i \mid \mathbf{pa}(\boldsymbol{s})_i, \boldsymbol{\theta}_i^c), \end{aligned} \tag{3}$$

where $\boldsymbol{\theta}_s^c = (\boldsymbol{\theta}_1^c, \ldots, \boldsymbol{\theta}_n^c)$ denotes the parameters for the local probability density functions for the predictive attributes when $C = c$.

If $\boldsymbol{s}^{\mathrm{h}}$ denotes the hypothesis that the true generalized joint probability distribution for $\boldsymbol{X}$ can be factorized according to the conditional independence assertions reflected in $\boldsymbol{s}$, then we obtain from Eq. (3) that,

$$\begin{aligned}\rho(\boldsymbol{x}\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}}) &= \rho(\boldsymbol{y},c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}}) = p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})f(\boldsymbol{y}\,|\,c,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})f_c(\boldsymbol{y}\,|\,\boldsymbol{\theta}_s^c,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\prod_{i=1}^{n} f_c(y_i\,|\,\mathbf{pa}(\boldsymbol{s})_i,\boldsymbol{\theta}_i^c,\boldsymbol{s}^{\mathrm{h}}).\end{aligned} \tag{4}$$

In order to encode a conditional Gaussian distribution for $\boldsymbol{X}$, the local probability density function for every predictive attribute Y_i of a CGN for $\boldsymbol{X}$ for data clustering must be the *linear-regression model*. Thus, when $C = c$,

$$f_c(y_i\,|\,\mathbf{pa}(\boldsymbol{s})_i,\boldsymbol{\theta}_i^c,\boldsymbol{s}^{\mathrm{h}}) \equiv \mathcal{N}\left(y_i; m_i^c + \sum_{Y_j\in\mathbf{Pa}(\boldsymbol{s})_i} b_{ji}^c(y_j - m_j^c), v_i^c\right), \tag{5}$$

where $\mathcal{N}(y;\mu,\sigma^2)$ is an univariate normal distribution with mean μ and standard deviation σ ($\sigma > 0$). When $C = c$, the parameters of the local probability density function for every Y_i are given by $\boldsymbol{\theta}_i^c = (m_i^c, \boldsymbol{b}_i^c, v_i^c)$. The interpretation of the components of the local parameters $\boldsymbol{\theta}_i^c$ is as follows for all i and all c state of C: m_i^c is the unconditional mean of Y_i when $C = c$, $\boldsymbol{b}_i^c = (b_{1i}^c,\ldots,b_{i-1i}^c)^t$ is a column vector where every b_{ji}^c is a linear coefficient reflecting the strength of the relationship between Y_j and Y_i when $C = c$ if $Y_j \in \mathbf{Pa}(\boldsymbol{s})_i$, otherwise $b_{ji}^c = 0$, and v_i^c is the conditional variance of Y_i given $\mathbf{Pa}(\boldsymbol{s})_i$ when $C = c$. See Fig. 1 for an example of a CGN for data clustering with three continuous predictive attributes and one binary cluster random variable.

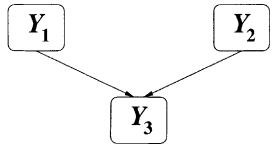
Note that the model structure is independent of the value of the cluster random variable C, thus, the model structure is the same for all the values of C. However, the parameters of the local probability density functions for the predictive attributes in $\boldsymbol{Y}$ do depend on the value of C and they may be different for the different values of the random variable C.

2.3. Bayesian structural EM algorithm for unsupervised learning of conditional Gaussian networks

One of the methods for learning CGNs from unlabeled data is the well-known Bayesian structural EM (BS-EM) algorithm developed by Friedman in [9]. Due to its good performance, this algorithm has received special attention in the literature and has motivated several variants of itself [25,30–32,37]. We use the BS-EM algorithm for explanatory purposes as well as in our experiments presented in Section 4.

When applying the BS-EM algorithm in a data clustering problem, we assume that we have a database of N cases, $\boldsymbol{d} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$, where every case is represented by an assignment to n of the $n+1$ unidimensional random variables involved in the problem domain. In this case, it is convenient to see $\boldsymbol{d}$ as a partial assignment to a set of $(n+1)N$ unidimensional random variables that

• **Model structure**



• **Local probability density functions**

$$
\begin{aligned}
&\boldsymbol{\theta}_1^{c_1} = (m_1^{c_1}, 0, v_1^{c_1}) && f_{c_1}(y_1 \mid \boldsymbol{\theta}_1^{c_1}, \boldsymbol{s}^h) \equiv \mathcal{N}(y_1; m_1^{c_1}, v_1^{c_1})\\
&\boldsymbol{\theta}_2^{c_1} = (m_2^{c_1}, 0, v_2^{c_1}) && f_{c_1}(y_2 \mid \boldsymbol{\theta}_2^{c_1}, \boldsymbol{s}^h) \equiv \mathcal{N}(y_2; m_2^{c_1}, v_2^{c_1})\\
&\boldsymbol{\theta}_3^{c_1} = (m_3^{c_1}, \boldsymbol{b}_3^{c_1}, v_3^{c_1}) && f_{c_1}(y_3 \mid y_1, y_2, \boldsymbol{\theta}_3^{c_1}, \boldsymbol{s}^h) \equiv \mathcal{N}(y_3; m_3^{c_1} + b_{13}^{c_1}(y_1 - m_1^{c_1}) + b_{23}^{c_1}(y_2 - m_2^{c_1}), v_3^{c_1})\\
&\boldsymbol{b}_3^{c_1} = (b_{13}^{c_1}, b_{23}^{c_1})^t &&\\
&\boldsymbol{\theta}_1^{c_2} = (m_1^{c_2}, 0, v_1^{c_2}) && f_{c_2}(y_1 \mid \boldsymbol{\theta}_1^{c_2}, \boldsymbol{s}^h) \equiv \mathcal{N}(y_1; m_1^{c_2}, v_1^{c_2})\\
&\boldsymbol{\theta}_2^{c_2} = (m_2^{c_2}, 0, v_2^{c_2}) && f_{c_2}(y_2 \mid \boldsymbol{\theta}_2^{c_2}, \boldsymbol{s}^h) \equiv \mathcal{N}(y_2; m_2^{c_2}, v_2^{c_2})\\
&\boldsymbol{\theta}_3^{c_2} = (m_3^{c_2}, \boldsymbol{b}_3^{c_2}, v_3^{c_2}) && f_{c_2}(y_3 \mid y_1, y_2, \boldsymbol{\theta}_3^{c_2}, \boldsymbol{s}^h) \equiv \mathcal{N}(y_3; m_3^{c_2} + b_{13}^{c_2}(y_1 - m_1^{c_2}) + b_{23}^{c_2}(y_2 - m_2^{c_2}), v_3^{c_2})\\
&\boldsymbol{b}_3^{c_2} = (b_{13}^{c_2}, b_{23}^{c_2})^t &&
\end{aligned}
$$

• **Multinomial distribution**

$$
\begin{aligned}
&p(c_1 \mid \boldsymbol{\theta}_s, \boldsymbol{s}^h)\\
&p(c_2 \mid \boldsymbol{\theta}_s, \boldsymbol{s}^h) = 1 - p(c_1 \mid \boldsymbol{\theta}_s, \boldsymbol{s}^h)
\end{aligned}
$$

Fig. 1. Model structure, multinomial distribution, and local probability density functions for a CGN for data clustering with three continuous predictive attributes and one binary cluster random variable.

we represent as $\boldsymbol{D} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$. Let $\boldsymbol{D}^{\boldsymbol{Y}} = \{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_N\}$ denote the set of observed random variables or predictive attributes in $\boldsymbol{D}$, that is, the nN unidimensional random variables that have assigned values in $\boldsymbol{d}$. Similarly, let $\boldsymbol{D}^C = \{C_1, \ldots, C_N\}$ denote the set of unobserved or cluster random variables in $\boldsymbol{D}$, that is, the N unidimensional random variables that reflect the unknown cluster membership of each case of $\boldsymbol{d}$. In the forthcoming, $\boldsymbol{d}^{\boldsymbol{Y}}$ and $\boldsymbol{d}^C$ represent an assignment to $\boldsymbol{D}^{\boldsymbol{Y}}$ and $\boldsymbol{D}^C$, respectively.

For learning CGNs from unlabeled data, the BS-EM algorithm performs a search over the space of CGNs for data clustering based on the well-known EM algorithm [6,23] and the direct optimization of a Bayesian score. As shown in Fig. 2, the BS-EM algorithm follows the basic intuition of the EM algorithm. To take advantage of the best estimate of the generalized joint probability distribution for the data found so far (current CGN for data clustering) in order to estimate missing values and then, to perform a structural search by using a procedure for complete data. Specifically, the BS-EM algorithm alternates between two steps. A step that finds the maximum a posteriori (MAP)

loop $l = 0, 1, \ldots$

1. Run the EM algorithm to compute the MAP parameters $\tilde{\boldsymbol{\theta}}_{\boldsymbol{s}_l}$ for $\boldsymbol{s}_l$ given $\boldsymbol{d}^{\boldsymbol{Y}}$
2. Perform a search over model structures, evaluating every one by
   $Score(\boldsymbol{s} : \boldsymbol{s}_l) = E[\log L(\boldsymbol{d} \mid \boldsymbol{s}^h) \mid \boldsymbol{d}^{\boldsymbol{Y}}, \tilde{\boldsymbol{\theta}}_{\boldsymbol{s}_l}, \boldsymbol{s}_l^h] = \sum_{\boldsymbol{d}^C} \log L(\boldsymbol{d}^{\boldsymbol{Y}}, \boldsymbol{d}^C \mid \boldsymbol{s}^h) L(\boldsymbol{d}^C \mid \boldsymbol{d}^{\boldsymbol{Y}}, \tilde{\boldsymbol{\theta}}_{\boldsymbol{s}_l}, \boldsymbol{s}_l^h)$
3. Let $\boldsymbol{s}_{l+1}$ be the model with the highest score among these encountered during the search
4. **if** $Score(\boldsymbol{s}_l : \boldsymbol{s}_l) = Score(\boldsymbol{s}_{l+1} : \boldsymbol{s}_l)$
   then return $\boldsymbol{s}_l$

Fig. 2. Schematic of the BS-EM algorithm.

parameters for the current CGN structure for data clustering, usually by means of the EM algorithm, and a step that searches over CGN structures for data clustering guided by a Bayesian score. At each iteration, the BS-EM algorithm attempts to maximize the expected Bayesian score instead of the true Bayesian score.

To completely specify the BS-EM algorithm, we have to decide on the structural search procedure (step 2 in Fig. 2). The usual approach is to perform a greedy hill-climbing search over CGN structures for data clustering considering all the possible additions, removals, and reversals of a single arc at each point in the search. This structural search procedure is desirable as it exploits the decomposition properties of CGNs for data clustering and the factorization properties of the Bayesian score. However, any structural search procedure that exploits these properties can be used.

However, the direct application of the BS-EM algorithm, as it appears depicted in Fig. 2, may result in an inefficient and unrealistic solution in order to perform unsupervised CGN selection due to the fact that the computation of Score($\boldsymbol{s} : \boldsymbol{s}_l$) implies a huge computational expense as it takes account of every possible completion of the original unlabeled database. This drawback or bottleneck may be very harmful when large databases in number of cases, number of predictive attributes, or both are considered. This is often our case as we are interested in solving data clustering problems of considerable size (medium and large size databases). To overcome this problem, we usually work with a relaxed version of the BS-EM algorithm that only considers what we call the MAP *completion* of the original unlabeled database $\boldsymbol{d}$ to compute Score($\boldsymbol{s} : \boldsymbol{s}_l$), instead of considering every possible completion. The MAP completion of $\boldsymbol{d}$ in the lth iteration of the BS-EM algorithm, here denoted by $\boldsymbol{d}'_l$, is obtained by completing every case of $\boldsymbol{d}$ with the label of the cluster where the maximum of the posterior probability distribution for the cluster random variable C given the case is achieved according to the best model found so far. Thus, this relaxed version of the BS-EM algorithm is comprised of the iteration between a parametric optimization for the current CGN for data clustering, and a structural search after (hidden) cluster random variable completion by means of probabilistic inference with the best estimate of the generalized joint

probability distribution so far (current CGN for data clustering). In this case, the score that guides the selection of the CGN structure for data clustering at each iteration of the BS-EM algorithm reduces to

$$\text{Score}(\boldsymbol{s} : \boldsymbol{s}_l) = \log L(\boldsymbol{d}'_l \,|\, \boldsymbol{s}^{\text{h}}), \tag{6}$$

where $\log L(\boldsymbol{d}'_l \,|\, \boldsymbol{s}^{\text{h}})$ denotes the the log *marginal likelihood* of the MAP completion of the original unlabeled data $\boldsymbol{d}$ given the structure of the current CGN for data clustering. This quantity can be calculated as follows.

According to [13], under the assumptions that (i) the database restricted to the values of the cluster random variable C, $\boldsymbol{d}^C$, is a multinomial sample, (ii) the database $\boldsymbol{d}$ is complete, and (iii) the parameters of the univariate multinomial distribution for C are independent and follow a Dirichlet distribution, we have that,

$$\begin{aligned} L(\boldsymbol{d} \,|\, \boldsymbol{s}^{\text{h}}) &= \prod_{l=1}^{N} \rho(\boldsymbol{x}_l \,|\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}, \boldsymbol{s}^{\text{h}}) \\ &= \prod_{l=1}^{N} p(c_l \,|\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}, \boldsymbol{s}^{\text{h}}) f(\boldsymbol{y}_l \,|\, c_l, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{l-1}, \boldsymbol{s}^{\text{h}}) \\ &= L(\boldsymbol{d}^C \,|\, \boldsymbol{s}^{\text{h}}) \prod_{l=1}^{N} f(\boldsymbol{y}_l \,|\, c_l, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{l-1}, \boldsymbol{s}^{\text{h}}) \\ &= L(\boldsymbol{d}^C \,|\, \boldsymbol{s}^{\text{h}}) \prod_{c \in \text{Val}(C)} L(\boldsymbol{d}^{\boldsymbol{Y}, c} \,|\, c, \boldsymbol{s}^{\text{h}}), \end{aligned} \tag{7}$$

where $\boldsymbol{d}^{\boldsymbol{Y}, c}$ is the database $\boldsymbol{d}$ restricted to the values for the continuous random variable $\boldsymbol{Y}$ and to cases where $C = c$, and $\text{Val}(C)$ is the set of values that the cluster random variable C can have. The term $L(\boldsymbol{d}^C \,|\, \boldsymbol{s}^{\text{h}})$ corresponds to the marginal likelihood of a trivial Bayesian network having only a single node, C. It can be calculated in closed form under reasonable assumptions according to [4]. Moreover, each term of the form $L(\boldsymbol{d}^{\boldsymbol{Y}, c} \,|\, c, \boldsymbol{s}^{\text{h}})$ represents the marginal likelihood of a domain containing only continuous random variables under the assumption that the data is sampled from a multivariate normal distribution. Then, these terms can be evaluated in factorable closed form under some reasonable assumptions according to [13,14,17].

3. Compromise conditional Gaussian networks for data clustering

After introducing unrestricted CGNs for data clustering in the previous section, we propose two classes of compromise CGNs for data clustering. These compromise models represent an appealing balance between the cost of the unsupervised model learning process (efficiency) and the quality of the learnt models (effectiveness). In addition to this trade-off between efficiency and effectiveness, some other motives to focus on compromise CGNs for data

clustering include (i) their closeness to human intuition, which results in a better readability and understandability than more general and complex CGNs for data clustering, (ii) the well-known difficulty involved in learning densely connected CGNs for data clustering, and (iii) the painfully slow probabilistic inference and simulation when working with densely connected CGNs for data clustering. Moreover, these drawbacks of densely connected CGNs for data clustering are aggravated when large databases in terms of number of cases, number of predictive attributes, or both are considered for unsupervised model learning or simulation. Thus, the development of methods for learning simple CGNs for data clustering while preserving the expressive power is completely justified.

3.1. Tree augmented naive Bayes models for data clustering

The first class of compromise CGNs for data clustering that we consider groups the so-called tree augmented naive Bayes (TANB) models for data clustering. These models are introduced for the first time in [11] as Bayesian classifiers, and are further developed in [10,12,19]. Later, TANB models are used in discrete data clustering problems [31,33]. TANB models for data clustering are defined by the following condition: Predictive attributes may have at most one other predictive attribute as a parent. See Fig. 3 (top) for an explanatory example of the structure of a TANB model for data clustering.

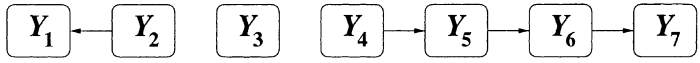
TANB models for data clustering typically imply computational advantages for the unsupervised model learning process when compared with general CGNs for data clustering as $\mathbf{Pa}(\boldsymbol{s})_i$ is either empty or a set containing only one node for all i. That is, the use of TANB models for data clustering results in a reduction of the model search space which usually decreases the learning cost, whereas the learnt models are still expressive enough. Another advantage of TANB models for data clustering is based on the fact that there is at most one path connecting every pair of predictive attributes. This is very appealing to human intuition due to the obvious conditional (in)dependence statements between subsets of predictive attributes which may be not easy to be read in densely connected CGNs for data clustering.

Notice should be taken that TANB models for data clustering are also known as *mixtures of trees with shared structure* in [24]. The interested reader should turn to this work to find out more about some advantages of these models that are out of the scope of the present work (probabilistic inference, simulation, marginalization, etc.).

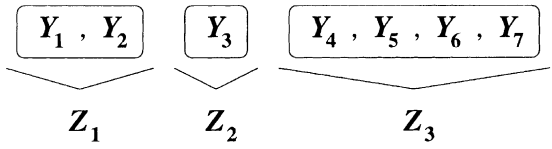
3.2. Extended naive Bayes models for data clustering

The well-known class of naive Bayes (NB) models [7,34] conforms one of the simplest examples of compromise CGNs that is usually considered in both

• **TANB model structure for data clustering**



• **ENB model structure for data clustering**



• **Equivalent CGN structure for data clustering in standard representation**

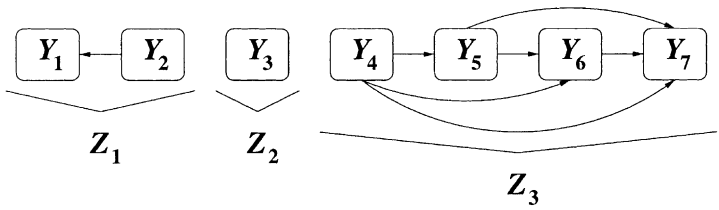


Fig. 3. Structures of a TANB model (top), and an ENB model (middle) and one of its equivalent CGNs in standard representation (bottom), when applied to data clustering.

data classification and data clustering problems. Its name comes from the naive assumption that all the predictive attributes are conditionally independent given the class (data classification) or cluster (data clustering) random variable C. Consequently, the generalized joint probability distribution for $\boldsymbol{X}$ encoded by a NB model for data clustering graphically factorizes as follows:

$$\begin{aligned}\rho(\boldsymbol{x}\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}}) &= \rho(\boldsymbol{y},c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})f(\boldsymbol{y}\,|\,c,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})f_c(\boldsymbol{y}\,|\,\boldsymbol{\theta}_s^c,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\prod_{i=1}^{n} f_c(y_i\,|\,\boldsymbol{\theta}_i^c,\boldsymbol{s}^{\mathrm{h}}).\end{aligned} \tag{8}$$

Although NB models for data clustering present a behavior rather dependent on the domain to which they are applied, because the assumption that all the predictive attributes are conditionally independent given C is often violated in practice, they perform surprisingly well in many domains. Hence, NB

models for data clustering represent an example of models that provide us with the ideal balance between efficiency and effectiveness that we pursue.

Keeping this idea in mind, extended naive Bayes (ENB) models are introduced by Pazzani [27,28] as Bayesian classifiers and, later, used by Peña et al. [30,32] for data clustering in discrete domains. ENB models for data clustering are very similar to NB models for data clustering as all the random variables represented by the nodes of the structure of an ENB model for data clustering are conditionally independent given the random variable C. The only difference with NB models for data clustering is that the number of nodes in the structure of an ENB model for data clustering can be shorter than the original number of predictive attributes in the database. The reason is that some of these predictive attributes can be grouped together under the same node as fully correlated predictive attributes given C (we call such nodes *supernodes*). In the forthcoming, we refer to the set of nodes and supernodes of an ENB model for data clustering simply as nodes. See Fig. 3 (middle) for an example of an ENB model structure for data clustering.

The structure of an ENB model for data clustering lends itself to a graphical factorization of the generalized joint probability distribution for $\boldsymbol{X}$ as follows:

$$\begin{aligned}\rho(\boldsymbol{x}\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}}) &= \rho(\boldsymbol{y},c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})f(\boldsymbol{y}\,|\,c,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})f_c(\boldsymbol{y}\,|\,\boldsymbol{\theta}_s^c,\boldsymbol{s}^{\mathrm{h}})\\ &= p(c\,|\,\boldsymbol{\theta}_s,\boldsymbol{s}^{\mathrm{h}})\prod_{i=1}^{r} f_c(\boldsymbol{z}_i\,|\,\boldsymbol{\theta}_i^c,\boldsymbol{s}^{\mathrm{h}}),\end{aligned} \tag{9}$$

where $\{\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_r\}$ is a partition of $\boldsymbol{Y}$, and r is the number of nodes in $\boldsymbol{s}$ (including the special nodes referred to as supernodes). Every $\boldsymbol{z}_i$ is the set of values in $\boldsymbol{y}$ for the original predictive attributes that are grouped together under the node $\boldsymbol{Z}_i$. In this case, $\boldsymbol{\theta}_i^c$ represents the set of parameters for the local probability density functions for the original predictive attributes grouped together under the node $\boldsymbol{Z}_i$ when $C=c$ for all i. Moreover, $\boldsymbol{s}$ is a directed acyclic graph that encodes the conditional (in)dependencies among the random variables represented by $\{\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_r\}$, and $\boldsymbol{s}^{\mathrm{h}}$ is defined as usual. Note the similarity between Eqs. (8) and (9). It should be also noticed that in our case we have that for each $\boldsymbol{Z}_i$,

$$f_c(\boldsymbol{z}_i\,|\,\boldsymbol{\theta}_i^c,\boldsymbol{s}^{\mathrm{h}}) \equiv \mathcal{N}(\boldsymbol{z}_i;\boldsymbol{\mu}(c),\Sigma(c)), \tag{10}$$

whenever $p(c)=p(C=c)>0$. Given $C=c$, $\boldsymbol{\mu}(c)$ is the multidimensional mean vector, and $\Sigma(c)$, the variance matrix, is positive definite.

ENB models for data clustering can be considered as having an intermediate place between NB models for data clustering and models with all the predictive

attributes fully correlated for data clustering. Thus, they keep the main features of both extremes. Simplicity from the former models and a better performance from the latter models. Note that both extremes are ENB models for data clustering as well.

To appreciate that ENB models for data clustering constitute a subset of CGNs for data clustering, the reader should note that every ENB model for data clustering can be translated into at least one equivalent CGN for data clustering in standard representation (see Section 2). This can be done by simply taking into account the notion of supernode. Every predictive attribute Y_i is fully correlated with the rest of the predictive attributes grouped together under the same supernode as Y_i given the cluster random variable C, but fully uncorrelated with the predictive attributes grouped together under the remaining supernodes given C. Thus, every supernode represents a local structure for the predictive attributes grouped together under the supernode that encodes no conditional independence assertions among them given C. Fig. 3 (bottom) shows one of the possible translations of the ENB model structure for data clustering depicted in the same figure (middle). Due to the fact that every supernode of a ENB model for data clustering represents a local complete structure for the predictive attributes grouped together under the supernode, the parameters of the ENB model for data clustering can be easily translated into the parameters of its equivalent CGN for data clustering in standard representation according to the work of Shachter and Kenley [35]. See also [13,14] for more details about this translation.

According to our particular objectives, the most interesting advantages derived from the use of ENB models for data clustering are: the saving of computational effort when performing unsupervised model learning due to the obvious reduction of the search space of CGNs for data clustering, and their readability as they keep the simplicity of NB models for data clustering. It is out of the scope of our present work the study of other possible advantages that would appear in the probabilistic inference, simulation, marginalization, etc., when working with ENB models for data clustering.

It should be noticed that not all the sets of conditional (in)dependence assertions encoded by TANB models for data clustering can be encoded by ENB models for data clustering and vice versa. For example, only when every supernode of a given ENB model for data clustering groups at most two original predictive attributes, all the conditional (in)dependencies encoded by the ENB model for data clustering can be encoded by an equivalent TANB model for data clustering. On the other hand, only when the length of every path between predictive attributes in a given TANB model for data clustering is at most one, all the conditional (in)dependence assertions encoded by this TANB model for data clustering can be also represented by an equivalent ENB model for data clustering. Thus, the different expressivity of TANB models and ENB models

for data clustering might imply that the former models appear to be more suitable for some domains while the latter models show a better behavior for some other domains.

4. Experimental evaluation

This section is devoted to the experimental evaluation and comparison of the two proposed classes of compromise CGNs for data clustering using both synthetic and real-world databases. In our experiments, we use the relaxed version of the BS-EM algorithm that just considers the MAP completion of the original unlabeled database in order to to compute Score($\boldsymbol{s} : \boldsymbol{s}_l$) at each iteration.

We also provide the reader with insight into the performance of the BS-EM algorithm for learning the classes of compromise CGNs for data clustering that we study. This is done through the performance of several non-parametric statistical tests in order to show statistically significant (in)dependence between the performance criteria considered and several factors.

To completely specify the BS-EM algorithm we have to decide on the structural search procedure (step 2 in Fig. 2). In our experimental evaluation, the BS-EM algorithm performs a greedy hill-climbing search over the proposed compromise model structures. When learning TANB models for data clustering, the structural search considers all the possible additions and removals of one arc at each point in the search. On the other hand, when learning ENB models for data clustering, the structural search considers all possible joints of two nodes and splits of one supernode at each point in the search. The initial model for the BS-EM algorithm is a NB model for data clustering with randomly chosen parameters.

4.1. Performance criteria and general considerations

Table 1 summarizes the criteria that we use to evaluate and compare the TANB models and ENB models for data clustering learnt by the BS-EM algorithm. The log marginal likelihood of the MAP completion of the database given the structure of the initial and the learnt TANB model or ENB model for data clustering is used in our comparison. For every synthetic database, where the original compromise CGN is available, we give special importance to the number of times that the structure of the original model is completely recovered. We also pay attention to the predictive ability of the learnt models measured as the logarithmic scoring rule of good [15]:

$$L_{\text{test}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \log f(\boldsymbol{y}_i \,|\, \boldsymbol{\theta}_{\boldsymbol{s}}, \boldsymbol{s}^{\text{h}}), \tag{11}$$

Table 1
Performance criteria

Expression	Comment
$L_{\text{initial}} \pm$ S.D.	Mean $\pm$ S.D. of the initial score (log marginal likelihood of the MAP completion of the database given the structure of the initial TANB model or ENB model for data clustering)
$L_{\text{final}} \pm$ S.D.	Mean $\pm$ S.D. of the final score (log marginal likelihood of the MAP completion of the database given the structure of the learnt TANB model or ENB model for data clustering)
Recovered	Number of times over all the independent runs for a given synthetic database that the structure of the original TANB model or ENB model is completely recovered
$L_{\text{test}} \pm$ S.D.	Mean $\pm$ S.D. of the multiple predictive accuracy of the learnt TANB model or ENB model for data clustering (logarithmic scoring rule of good)
Time $\pm$ S.D.	Mean $\pm$ S.D. of the runtime (in seconds)

where N_{test} is the number of cases in the testing database $\boldsymbol{d}_{\text{test}}$, and $\boldsymbol{y}_i$ denotes the ith case of $\boldsymbol{d}_{\text{test}}$ restricted to the values for the predictive attributes. Moreover, $f(\boldsymbol{y}_i \mid \boldsymbol{\theta}_s, \boldsymbol{s}^{\text{h}})$ can be calculated as $\sum_{c \in \text{Val}(c)} p(c \mid \boldsymbol{\theta}_s, \boldsymbol{s}^{\text{h}}) f_c(\boldsymbol{y}_i \mid \boldsymbol{\theta}_i^c, \boldsymbol{s}^{\text{h}})$. The higher the value for L_{test}, the higher the multiple predictive accuracy of the elicited compromise CGN for data clustering. In addition to this, we consider the runtime of the unsupervised model learning process as valuable information.

All the average and standard deviation values reported for the performance criteria are over ten independent runs. The values for the performance criterion recovered are also over ten independent runs. All the experiments are run on a Pentium 233 MHz computer. In all the experiments, we assume that the real number of clusters is known beforehand, thus, we do not perform a search to identify the true number of clusters in the databases considered.

The convergence criterion for the EM algorithm that the BS-EM algorithm runs at each iteration is satisfied when either the relative difference between successive values for the log marginal likelihood of the MAP completion of the database given a TANB model or ENB model structure for data clustering is less than 10^{-6} or 150 iterations are reached.

4.2. Results on synthetic databases

In this section, we describe our experimental results on synthetic data. The purpose of evaluating TANB models and ENB models for data clustering on synthetic databases is to show their effectiveness (measured in terms of difference between L_{final} and L_{initial}, ability to completely recover the original structure, recovered, and predictive ability of the learnt models, L_{test}), as well as their efficiency (measured by time).

We considered three factors or dimensions in the construction of each of the synthetic databases:

- Complexity of the structure of the model to be simulated, measured by the number of directed edges in the case of a TANB model and by the number of supernodes in the case of an ENB model.
- Values of the linear coefficients reflecting the strength of the relationships between parents and children in the model to be simulated (see Eq. (5)). In the case of an ENB model, where there is not such relationships, we simulated an equivalent CGN in standard representation (remember Section 3.2).
- Degree of overlapping between the joint probability density functions of the clusters encoded by the model to be simulated. This is a function of the complexity of the model to be simulated and the values of the unconditional means (fixed in our case), linear coefficients, and conditional variances of the predictive attributes involved (see Eq. (5). Then, we focus on the values for the conditional variances because the complexity of the model and the linear coefficients of the predictive attributes correspond to the other two factors that we study.

There were involved 20 continuous predictive attributes and one three-valued cluster random variable in the original compromise CGNs. We randomly chose four TANB model structures of different complexity. The number of directed edges of each of these four TANB model structures was 0, 5, 10, and 15. Also, we randomly chose four ENB model structures of different complexity. The number of nodes of each of these four ENB model structures was 20, 15, 10, and 5. Therefore, in the last three models some of the 20 original predictive attributes had to be randomly grouped together under supernodes. In fact, these randomly chosen ENB models implied 0, 9, 14, and 31 arcs, respectively, when translated into equivalent CGNs in standard representation as indicated in Section 3.2. We considered three possible values for the linear coefficients reflecting the strength of the relationships between parents and children, these were 1, 0.5, and 0.25. Moreover, the unconditional mean of every predictive attribute was fixed to zero for the first value of the cluster random variable, to four for the second, and to eight for the third. The values considered for the conditional variances of every predictive attribute were 1, 2, and 3. Thus, when the linear coefficients of every predictive attribute were set to one and the conditional variances to three, the degree of overlapping between the joint probability density functions of the clusters achieved the maximum for each of the model complexities considered.

Taking account of all the possible combinations of the values of the three referred factors (four degrees of complexity, three values for the linear coefficients, and three values for the conditional variances), we constructed 36 TANB models and 36 ENB models to be simulated. These 72 models are referred to as the original or underlying models in the forthcoming discussion. The local probability distribution for the cluster random variable was uniform for these 72 compromise CGNs. We sampled 8000 cases from each of these models for the synthetic learning databases and 2000 cases for the testing

databases in order to compute L_{test}. Obviously, we discarded all the entries corresponding to the cluster random variable in the 72 learning databases and 72 testing databases.

We find it convenient to use a regular grammar to clearly identify each of the 72 learning databases. Then, TANBβ_{γ}^{δ} denotes the database created from the original TANB model with β arcs, $\beta = 0, 5, 10, 15$, when all the linear coefficients take value γ, $\gamma = 1, 0.5, 0.25$, and all the conditional variances are δ, $\delta = 1, 2, 3$. Similarly, ENBβ_{γ}^{δ} denotes the database generated from the original ENB model with β nodes, $\beta = 20, 15, 10, 5$, being γ and δ as already explained.

For the non-parametric statistical tests performed in order to elicit statistically significant (in)dependence between the performance criteria involved in the evaluation and the three factors considered in the construction of the synthetic databases, we assume $\alpha = 0.05$ as the significance level. However, we report on the *P*-values obtained in order to allow the reader to judge whether or not the differences are statistically significant. Concretely, χ^2 and Kruskal–Wallis statistical tests are considered.

As Table 2 shows, when learning TANB models for data clustering from the unlabeled synthetic TANB databases, the original model structures are completely recovered most of the times. Moreover, when the structure of the underlying TANB model for a given synthetic database is discovered, the parameters of the original model are also recovered with high accuracy. Thus, the BS-EM algorithm exhibits an effective behavior for learning this class of compromise CGNs for data clustering. According to the χ^2 statistical tests performed between recovered and each of the three factors considered in the construction of the synthetic databases (denoted by complexity, coefficients, and variances in Table 3), the obtained *P*-values do not support the rejection of the null hypothesis in any case. That is, there is not statistically significant difference between the distributions of the experimental results obtained for recovered with respect to any of the three factors. Thus, not statistically significant dependence between this measure of the effectiveness of TANB models for data clustering and any of the three factors is found.

It clearly appears from Table 2 the great improvement of the log marginal likelihood achieved by the elicited TANB models for data clustering, as it is shown by the difference between L_{final} and L_{initial}. This fact reinforces our assertions about the modeling expressivity of TANB models for data clustering. According to the *P*-values reported in Table 3, the Kruskal–Wallis statistical tests carried out show that there are statistically significant differences between the distributions of the values of the difference $L_{\text{final}} - L_{\text{initial}}$ with respect to complexity and coefficients. Thus, there exists statistically significant dependence between this measure of the effectiveness and the complexity and the linear coefficients of the underlying model. The improvement of the log marginal likelihood increases when the complexity or the linear coefficients increase(s). On the other hand, the performed Kruskal–Wallis statistical tests do

Table 2
Performance achieved when learning TANB models for data clustering from the 36 synthetic TANB databases

Database	$L_{\text{initial}} \pm$ S.D.	$L_{\text{final}} \pm$ S.D.	Recovered	$L_{\text{test}} \pm$ S.D.	Time $\pm$ S.D.
TANB0_1^1	-145407 ± 17723	-129407 ± 5551	8	-13.13 ± 0.67	372 ± 583
$\text{TANB0}_{0.5}^1$	-137937 ± 17253	-128021 ± 4168	9	-12.96 ± 0.51	346 ± 810
$\text{TANB0}_{0.25}^1$	-143052 ± 15123	-126632 ± 0	10	-12.79 ± 0.00	84 ± 35
TANB0_1^2	-174686 ± 10899	-156398 ± 4503	8	-16.08 ± 0.56	378 ± 606
$\text{TANB0}_{0.5}^2$	-173937 ± 12302	-155296 ± 3446	9	-15.94 ± 0.42	286 ± 571
$\text{TANB0}_{0.25}^2$	-173544 ± 10578	-156468 ± 4642	8	-16.08 ± 0.56	449 ± 767
TANB0_1^3	-181671 ± 8306	-172230 ± 3971	8	-17.80 ± 0.48	307 ± 457
$\text{TANB0}_{0.5}^3$	-181296 ± 9251	-171213 ± 2906	9	-17.68 ± 0.36	230 ± 448
$\text{TANB0}_{0.25}^3$	-189797 ± 13783	-171226 ± 2944	9	-17.68 ± 0.36	188 ± 280
TANB5_1^1	-146001 ± 13975	-126671 ± 0	10	-12.79 ± 0.00	123 ± 15
$\text{TANB5}_{0.5}^1$	-147909 ± 15852	-127808 ± 3420	9	-12.93 ± 0.42	201 ± 252
$\text{TANB5}_{0.25}^1$	-168741 ± 25245	-126667 ± 0	10	-12.79 ± 0.00	127 ± 12
TANB5_1^2	-176656 ± 10439	-155022 ± 2498	9	-15.90 ± 0.31	273 ± 429
$\text{TANB5}_{0.5}^2$	-172351 ± 13249	-155131 ± 2832	9	-15.92 ± 0.34	212 ± 268
$\text{TANB5}_{0.25}^2$	-171420 ± 10298	-156265 ± 4157	8	-16.05 ± 0.50	394 ± 546
TANB5_1^3	-191028 ± 8035	-172423 ± 3260	7	-17.82 ± 0.39	306 ± 265
$\text{TANB5}_{0.5}^3$	-183830 ± 9192	-171084 ± 2390	9	-17.66 ± 0.28	255 ± 372
$\text{TANB5}_{0.25}^3$	-186011 ± 10597	-172946 ± 4062	7	-17.88 ± 0.48	424 ± 454
TANB10_1^1	-173421 ± 16617	-126719 ± 0	10	-12.79 ± 0.00	176 ± 21
$\text{TANB10}_{0.5}^1$	-162582 ± 11058	-129465 ± 4206	7	-13.12 ± 0.50	438 ± 446
$\text{TANB10}_{0.25}^1$	-140031 ± 17182	-128996 ± 4572	8	-13.06 ± 0.54	377 ± 451
TANB10_1^2	-187000 ± 6669	-155813 ± 2404	7	-15.99 ± 0.29	505 ± 456

(*continued on next page*)

Table 2 (*Continued*)

Database	$L_{\text{initial}} \pm$ S.D.	$L_{\text{final}} \pm$ S.D.	Recovered	$L_{\text{test}} \pm$ S.D.	Time $\pm$ S.D.
$\text{TANB10}^2_{0.5}$	-175807 ± 16556	-155695 ± 2918	8	-15.97 ± 0.35	387 ± 452
$\text{TANB10}^2_{0.25}$	-166660 ± 12249	-156109 ± 3748	8	-16.02 ± 0.44	330 ± 365
TANB10^3_{1}	-195496 ± 3861	-170338 ± 0	10	-17.56 ± 0.00	282 ± 36
$\text{TANB10}^3_{0.5}$	-184310 ± 11060	-170937 ± 1797	9	-17.63 ± 0.22	307 ± 328
$\text{TANB10}^3_{0.25}$	-183411 ± 7326	-171128 ± 2371	9	-17.65 ± 0.28	302 ± 449
TANB15^1_{1}	-165994 ± 11718	-127382 ± 1250	8	-12.87 ± 0.15	327 ± 119
$\text{TANB15}^1_{0.5}$	-142337 ± 13188	-127419 ± 2015	9	-12.87 ± 0.24	260 ± 236
$\text{TANB15}^1_{0.25}$	-143321 ± 15604	-126745 ± 0	10	-12.79 ± 0.00	179 ± 17
TANB15^2_{1}	-194388 ± 9055	-154266 ± 0	10	-15.80 ± 0.00	346 ± 49
$\text{TANB15}^2_{0.5}$	-168293 ± 8935	-154275 ± 0	10	-15.80 ± 0.00	242 ± 113
$\text{TANB15}^2_{0.25}$	-166754 ± 10621	-154274 ± 0	10	-15.80 ± 0.00	182 ± 9
TANB15^3_{1}	-206382 ± 6555	-170341 ± 0	10	-17.58 ± 0.00	950 ± 146
$\text{TANB15}^3_{0.5}$	-192429 ± 5549	-171188 ± 1623	8	-17.66 ± 0.20	490 ± 289
$\text{TANB15}^3_{0.25}$	-188984 ± 10148	-172390 ± 3073	7	-17.80 ± 0.37	473 ± 368

All the average and standard deviation values are over ten independent runs. The values for recovered are also over ten independent runs.

Table 3
P-values for the non-parametric statistical tests performed when evaluating the differences in the performance criteria values obtained from the 36 synthetic TANB databases

	Recovered	Complexity	Coefficients	Variances
Recovered	–	0.595 χ^2	0.975 χ^2	0.498 χ^2
$L_{\text{final}} - L_{\text{initial}}$	0.318 K–W	0.000 K–W	0.000 K–W	0.052 K–W
L_{test}	0.000 K–W	0.001 K–W	0.920 K–W	0.000 K–W
Time	0.000 K–W	0.000 K–W	0.031 K–W	0.016 K–W

χ^2 indicates the performance of a χ^2 statistical test and K–W indicates the performance of a Kruskal–Wallis statistical test.

not show statistically significant dependence between the improvement of the log marginal likelihood and recovered.

Table 2 also depicts the results obtained for the predictive ability of the learnt TANB models for data clustering on the testing databases, L_{test}. Statistically significant dependence exists between L_{test} and the ability to recover the underlying model, the complexity of the original TANB model, and the conditional variances of the predictive attributes, according to the Kruskal–Wallis statistical tests performed (see Table 3). The increase of the ability to discover the underlying model for a given database implies the increase of the predictive ability of the learnt TANB model for data clustering. On the other hand, the predictive ability of the learnt model decreases with the increase of the conditional variances of the predictive attributes of the underlying TANB model. As a result, we could conclude that, from the point of view of the effectiveness, TANB models for data clustering offer a good performance level as well as high reliability to recover the underlying models from the synthetic databases considered.

From the point of view of the efficiency (measured as the runtime of the unsupervised model learning process, time), our experimental results show empirical evidence of the trade-off between effectiveness and efficiency that TANB models for data clustering provide with: the runtime of the BS-EM algorithm is reasonable, taking into account the considerable dimensionality and number of cases of the synthetic databases in addition to the, sometimes, complex underlying models. According to the *P*-values reported in Table 3 for the Kruskal–Wallis statistical tests performed, statistically significant dependence between the cost of the unsupervised model learning process, i.e., runtime of the BS-EM algorithm, and each of the three factors exists. Also, the results of the tests reveal statistically significant dependence between the efficiency and the ability to recover the underlying model of a given database. As one might expect, the cost of the unsupervised model learning process increases with the increase of the complexity, the linear coefficients, or the conditional variances of the predictive attributes of the original TANB model, while decreasing as recovered increases.

Table 4 shows the performance achieved when learning ENB models for data clustering from the unlabeled synthetic ENB databases by means of the BS-EM algorithm. Roughly speaking, the results achieved on the synthetic ENB databases are similar to those presented in Table 2 for the models elicited from the synthetic TANB databases. Thus, TANB models for data clustering as well as ENB models for data clustering provide us with an appealing balance between effectiveness and efficiency. However, in the case of learning ENB models for data clustering the results appear to be more noticeable as the complexity of the original ENB models was greater than the complexity of the original TANB models. Recall that the original ENB models had equivalent CGNs in standard representation with 0, 9, 14, and 31 arcs, while the original TANB models had only 0, 5, 10, and 15 arcs.

The only ENB databases where the results are not satisfying at all are $ENB5_1^2$ and $ENB5_1^3$. The results reflect the inability of ENB models for data clustering to capture the underlying models as the overlapping between the joint probability density functions of the three existing clusters was considerable in these databases. This can be appreciated from the complexity of the models that were used to generate the databases $ENB5_1^2$ and $ENB5_1^3$ (only five nodes and all of them supernodes), the linear coefficients (equal to one), and the conditional variances of the predictive attributes (equal to two for the former database and to three for the latter). Remember Eq. (5) to appreciate the difficulty involved in recovering the original models from the databases $ENB5_1^2$ and $ENB5_1^3$.

With regard to the non-parametric statistical tests performed for the results obtained when learning ENB models for data clustering from the ENB databases via the BS-EM algorithm, we can say that they are very similar to those that we report in Table 3. However, some differences appear from the comparison between this table and Table 5. It is noticeable that the statistically significant dependence that exists between recovered and each of the three factors considered in the generation of the synthetic databases, when learning ENB models for data clustering. Remember that such statistically significant dependence is not observed when TANB models for data clustering are induced from the TANB databases. Concretely, the rate of recovered decreases as the conditional variances of the predictive attributes of the underlying model increase. Another difference between Tables 3 and 5 is the statistically significant dependence between the improvement of the log marginal likelihood achieved and the ability to recover the underlying model of a given database. To be more exact, as the ability to recover the original ENB model increases, the improvement of the log marginal likelihood reduces. The reason is that an initial model that has a poor log marginal likelihood may make the unsupervised model learning algorithm unable to discover the underlying model. However, the elicited model, despite being different from the original model, can improve substantially the initial model in terms of log marginal likelihood.

Table 4
Performance achieved when learning ENB models for data clustering from the 36 synthetic ENB databases

Database	$L_{\text{initial}} \pm$ S.D.	$L_{\text{final}} \pm$ S.D.	Recovered	$L_{\text{test}} \pm$ S.D.	Time $\pm$ S.D.
$ENB20_1^1$	-139968 ± 16709	-126631 ± 0	10	-12.79 ± 0.00	170 ± 25
$ENB20_{0.5}^1$	-132789 ± 12195	-126632 ± 0	10	-12.79 ± 0.00	155 ± 20
$ENB20_{0.25}^1$	-135483 ± 18267	-126631 ± 0	10	-12.79 ± 0.00	169 ± 15
$ENB20_1^2$	-168420 ± 14391	-154603 ± 1367	9	-15.84 ± 0.10	222 ± 142
$ENB20_{0.5}^2$	-167999 ± 15185	-154147 ± 0	10	-15.80 ± 0.00	187 ± 27
$ENB20_{0.25}^2$	-165552 ± 13372	-154604 ± 1371	9	-15.84 ± 0.10	197 ± 71
$ENB20_1^3$	-184454 ± 9316	-171068 ± 1647	8	-17.62 ± 0.12	392 ± 514
$ENB20_{0.5}^3$	-181481 ± 9833	-171067 ± 1643	8	-17.62 ± 0.11	253 ± 135
$ENB20_{0.25}^3$	-185655 ± 8843	-170650 ± 1217	9	-17.59 ± 0.09	265 ± 125
$ENB15_1^1$	-153876 ± 20567	-127074 ± 1430	9	-12.86 ± 0.12	306 ± 199
$ENB15_{0.5}^1$	-140303 ± 14803	-127078 ± 1450	9	-12.86 ± 0.13	307 ± 237
$ENB15_{0.25}^1$	-151038 ± 20620	-127593 ± 1997	8	-12.90 ± 0.18	533 ± 610
$ENB15_1^2$	-184477 ± 15610	-154525 ± 1227	9	-15.85 ± 0.09	307 ± 106
$ENB15_{0.5}^2$	-175527 ± 12872	-154530 ± 1245	9	-15.86 ± 0.10	391 ± 408
$ENB15_{0.25}^2$	-171009 ± 13400	-154883 ± 1552	8	-15.89 ± 0.13	314 ± 175
$ENB15_1^3$	-191186 ± 8337	-170949 ± 1467	8	-17.64 ± 0.10	657 ± 872
$ENB15_{0.5}^3$	-184510 ± 11663	-171349 ± 1733	7	-17.67 ± 0.12	656 ± 664
$ENB15_{0.25}^3$	-182646 ± 8536	-170993 ± 1557	8	-17.64 ± 0.11	573 ± 884
$ENB10_1^1$	-165082 ± 12643	-127231 ± 1447	9	-12.84 ± 0.13	425 ± 514
$ENB10_{0.5}^1$	-147749 ± 13386	-126741 ± 0	10	-12.79 ± 0.00	243 ± 18
$ENB10_{0.25}^1$	-151382 ± 15171	-127209 ± 1410	9	-12.84 ± 0.12	371 ± 383
$ENB10_1^2$	-192655 ± 7961	-155099 ± 1652	8	-15.87 ± 0.13	347 ± 128

(*continued on next page*)

Table 4 (*Continued*)

Database	$L_{\text{initial}} \pm$ S.D.	$L_{\text{final}} \pm$ S.D.	Recovered	$L_{\text{test}} \pm$ S.D.	Time $\pm$ S.D.
$\text{ENB10}^2_{0.5}$	-166961 ± 10491	-154269 ± 0	10	-15.80 ± 0.00	243 ± 34
$\text{ENB10}^2_{0.25}$	-171971 ± 13195	-155068 ± 1600	8	-15.87 ± 0.12	389 ± 261
ENB10^3_{1}	-204529 ± 5266	-171498 ± 1715	7	-17.64 ± 0.12	506 ± 261
$\text{ENB10}^3_{0.5}$	-187853 ± 7283	-170373 ± 0	10	-17.57 ± 0.00	328 ± 46
$\text{ENB10}^3_{0.25}$	-179607 ± 8690	-170731 ± 1077	9	-17.59 ± 0.07	306 ± 171
ENB5^1_{1}	-200981 ± 7884	-127384 ± 1695	9	-12.87 ± 0.16	521 ± 408
$\text{ENB5}^1_{0.5}$	-165111 ± 11959	-126861 ± 0	10	-12.82 ± 0.00	274 ± 19
$\text{ENB5}^1_{0.25}$	-153217 ± 18286	-127589 ± 1585	8	-12.89 ± 0.14	876 ± 1450
ENB5^2_{1}	-222545 ± 5611	-154675 ± 0	0	-15.83 ± 0.00	536 ± 106
$\text{ENB5}^2_{0.5}$	-189427 ± 6562	-154656 ± 941	9	-15.85 ± 0.07	394 ± 84
$\text{ENB5}^2_{0.25}$	-172131 ± 12697	-154664 ± 973	9	-15.85 ± 0.07	340 ± 207
ENB5^3_{1}	-236717 ± 4453	-171492 ± 0	0	-17.60 ± 0.00	707 ± 81
$\text{ENB5}^3_{0.5}$	-200792 ± 4858	-170455 ± 0	10	-17.59 ± 0.00	543 ± 118
$\text{ENB5}^3_{0.25}$	-191012 ± 14548	-170738 ± 855	9	-17.61 ± 0.06	484 ± 251

All the average and standard deviation values are over ten independent runs. The values for recovered are also over ten independent runs.

Table 5
P-values for the non-parametric statistical tests performed when evaluating the differences in the performance criteria values obtained from the 36 synthetic ENB databases

	Recovered	Complexity	Coefficients	Variances
Recovered	–	0.001 χ^2	0.000 χ^2	0.005 χ^2
$L_{\text{final}} - L_{\text{initial}}$	0.000 K–W	0.000 K–W	0.000 K–W	0.388 K–W
L_{test}	0.000 K–W	0.000 K–W	0.778 K–W	0.000 K–W
Time	0.000 K–W	0.000 K–W	0.023 K–W	0.000 K–W

χ^2 indicates the performance of a χ^2 statistical test and K–W indicates the performance of a Kruskal–Wallis statistical test.

A closer look at the learnt compromise CGNs for data clustering reveals the decisive influence of the quality of the initial models on the quality of the learnt models (see the sometimes large standard deviation values for the performance criteria that we report in Table 2 and 4). Hence, it appears to be specially relevant the study of the initial conditions in order to improve the results in terms of all the performance measures considered in this work.

4.3. Results on real-world databases

Another source of data for our evaluation consists of three well-known real-world databases from the UCI repository of machine learning databases [26]. These databases provide us with a more realistic framework for the evaluation. Moreover, they allow us to compare the performance achieved by TANB models and ENB models for data clustering in the same domains. The databases considered are the following ones:

- Waveform [3] which is an artificial database consisting of 21 predictive attributes. There are three classes. Instances of each class represent a random convex combination of two of three base waves with noise added. We used the data set generator from the UCI repository to create a learning database with 4000 cases and a testing database with 1000 cases.
- Waveform21 + 19 [3] which is an artificial database consisting of 40 predictive attributes. The first 21 predictive attributes of the instances of each class represent a random convex combination of two of three base waves with noise added. The last 19 predictive attributes are noise attributes which turn out to be irrelevant for describing the underlying three classes. We used the data set generator from the UCI repository to create a learning database with 4000 cases and a testing database with 1000 cases.
- Pima [36] which is a real database containing 768 cases and eight predictive attributes. There are two classes. Every instance represents the record of a female patient tested for diabetes. We used the first 600 cases for learning and the last 168 cases for testing.

The first two databases were chosen due to our interest in working with databases of considerable size (thousands of cases and tens of predictive

attributes). The third database, considerably shorter in both number of cases and number of predictive attributes, was chosen to get feedback on the scalability of the results. Obviously, we deleted all the class entries for the three learning and three testing databases.

As in the evaluation on synthetic data, we assume $\alpha = 0.05$ as the significance level for the non-parametric statistical tests performed in order to elicit statistically significant differences in the results for the performance criteria involved in the evaluation with respect to the domains considered. However, we report on the P-values obtained in order to allow the reader to judge whether or not the differences are statistically significant. Concretely, Mann–Whitney statistical tests are considered.

Table 6 summarizes the results obtained by the BS-EM algorithm when learning TANB models and ENB models for data clustering from the three real-world databases. This table reveals the good performance of both classes of compromise CGNs for data clustering when applied to real-world domains, as well as the suitability of the BS-EM algorithm for unsupervised model induction in these domains. By comparing the performance reached by the elicited TANB models and ENB models for data clustering, we can conclude that in the two largest and, thus, more interesting databases, the waveform and waveform21 + 19 databases, the learnt ENB models for data clustering outperform the obtained TANB models for data clustering in terms of effectiveness (L_{final} and L_{test}). In the third database, however, the results address a better behavior of the induced TANB models for data clustering over the obtained ENB models for data clustering. Thus, ENB models for data clustering appear to be more suitable than TANB models for data clustering for the two largest real-world databases considered, while in the third one the latter models behave in a better way than the former models.

According to the P-values that Table 7 reports for the Mann–Whitney statistical tests performed for the results obtained, there exist statistically significant differences between the distributions of L_{test} for the TANB models

Table 6
Performance achieved when learning TANB models and ENB models for data clustering from the three real-world databases

Database	Model	$L_{\text{initial}} \pm$ S.D.	$L_{\text{final}} \pm$ S.D.	$L_{\text{test}} \pm$ S.D.	Time $\pm$ S.D.
Waveform	TANB	-72069 ± 1765	-67531 ± 3	-14.21 ± 0.00	651 ± 265
	ENB	-72729 ± 2518	-67255 ± 7	-14.03 ± 0.01	479 ± 72
Waveform21 + 19	TANB	-127109 ± 2032	-120981 ± 0	-25.90 ± 0.00	672 ± 79
	ENB	-127259 ± 1791	-120713 ± 0	-25.72 ± 0.00	1455 ± 111
Pima	TANB	-8372 ± 65	-7706 ± 393	-11.56 ± 0.91	9 ± 3
	ENB	-8324 ± 52	-7946 ± 317	-12.10 ± 0.72	10 ± 2

All the average and standard deviation values are over ten independent runs.

Table 7
P-values for the different Mann–Whitney statistical tests performed when comparing TANB models and ENB models for data clustering according to the differences in the performance criteria values obtained from the three real-world databases

	$L_{final} - L_{initial}$	L_{test}	Time
Waveform	0.326	0.000	0.199
Waveform21 + 19	0.496	0.000	0.000
Pima	0.070	0.067	0.496

and ENB models for data clustering learnt from the waveform and waveform21 + 19 databases via the BS-EM algorithm. Also, statistically significant difference between the distributions of the cost of the unsupervised model induction process when learning TANB models and ENB models for data clustering from the waveform21 + 19 database is found. Summarizing, the increase in the predictive ability reached by the elicited ENB models for data clustering with respect to the predictive ability achieved by the induced-TANB models for data clustering for the waveform and waveform21 + 19 databases is statistically significant. Also the increase of the cost of learning ENB models for data clustering with respect to the cost of learning TANB models for data clustering from the waveform21 + 19 database is statistically significant. Thus, there are statistically significant differences between the effectiveness and efficiency of the TANB models and ENB models for data clustering elicited from these databases. Remember our comments about the different expressivity of TANB models and ENB models for data clustering that appear in Section 3.2.

As done in the case of the synthetic databases, it seems interesting to inspect the standard deviation values for the different performance criteria that we use for the real-world databases. Contrary to what happened with the synthetic domains, the standard deviations reported in Table 6 have very small values, which reflects the robustness of the BS-EM algorithm in these databases.

5. Conclusions

We have proposed two classes of compromise CGNs for data clustering: TANB models and ENB models for data clustering. Some of their advantages have been addressed: trade-off between efficiency and effectiveness, computational advantages, readability, interpretability, closeness to human intuition, etc. Moreover, an experimental evaluation and comparison of both classes of compromise CGNs for data clustering on synthetic and real-world databases have been carried out. The results obtained on the synthetic databases have shown the modeling power of TANB models and ENB models for data

clustering as, most of the times, the underlying model was completely discovered while the cost of the unsupervised model learning process was kept at a reasonable level. From the results achieved on real-world databases, it is noticeable the slight outperformance of the ENB models for data clustering learnt from the two largest databases over the obtained TANB models for data clustering.

In addition to this, we have provided the reader with insight into the performance of the BS-EM algorithm for learning the classes of compromise CGNs for data clustering that we have studied in this work. This has been achieved by performing several non-parametric statistical tests in order to show statistically significant (in)dependence between the performance criteria considered and several factors.

Some lines of future research could be the study of some other advantages of TANB models and ENB models for data clustering (probabilistic inference, simulation, marginalization, etc.), a proper study of the influence of the initial conditions, and/or the evaluation of the performance of TANB models and ENB models for data clustering with respect to some other factors (number of predictive attributes, number of clusters, number of cases, etc.).

Acknowledgements

The authors would like to thank the anonymous referees for their useful comments related to this work. This work was supported by the Spanish Ministry of Education and Culture (Ministerio de Educación y Cultura) under AP97 44673053 grant.

References

[1] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, New York, 1973.
[2] J. Banfield, A. Raftery, Model-based Gaussian and non-Gaussian clustering, Biometrics 49 (1993) 803–821.
[3] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth International Group, Belmont, CA, 1984.
[4] G. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (1992) 309–347.
[5] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, D.J. Spiegelhalter, Probabilistic Networks and Expert Systems, Springer, New York, 1999.
[6] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B 39 (1977) 1–38.
[7] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[8] D. Fisher, Knowledge acquisition via incremental conceptual clustering, Mach. Learn. 2 (1987) 139–172.

[9] N. Friedman, The Bayesian structural EM algorithm, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1998, pp. 129–138.
[10] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.
[11] N. Friedman, M. Goldszmidt, Building classifiers using Bayesian networks, in: Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1996, pp. 1277–1284.
[12] N. Friedman, M. Goldszmidt, T. Lee, Bayesian network classification with continuous attributes: getting the best of both discretization and parametric fitting, in: Proceedings of the Fifteenth National Conference on Machine Learning, 1998.
[13] D. Geiger, D. Heckerman, Learning Gaussian networks, Technical report MSR-TR-94-10, Microsoft Research, Redmond, WA, 1994.
[14] D. Geiger, D. Heckerman, Learning Gaussian networks, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, Seattle, WA, 1995, pp. 235–243.
[15] I. Good, Rational decisions, J. R. Stat. Soc. B 14 (1952) 107–114.
[16] J.A. Hartigan, Clustering Algorithms, Wiley, Canada, 1975.
[17] D. Heckerman, D. Geiger, Likelihoods and parameter priors for Bayesian networks, Technical report MSR-TR-95-54, Microsoft Research, Redmond, WA, 1995.
[18] L. Kaufman, P. Rousseeuw, Finding Groups in Data, Wiley, New York, 1990.
[19] E.J. Keogh, M.J. Pazzani, Learning augmented Bayesian classifiers: a comparison of distribution-based and classification-based approaches, in: Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, 1999, pp. 225–230.
[20] S.L. Lauritzen, Propagation of probabilities, means and variances in mixed graphical association models, J. Am. Stat. Assoc. 87 (1992) 1098–1108.
[21] S.L. Lauritzen, Graphical Models, Clarendon Press, Oxford, 1996.
[22] S.L. Lauritzen, N. Wermuth, Graphical models for associations between variables, some of which are qualitative and some quantitative, Ann. Stat. 17 (1989) 31–57.
[23] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley, New York, 1997.
[24] M. Meilă, Learning with Mixtures of Trees, Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1999.
[25] M. Meilă, M.I. Jordan, Estimating dependency structure as a hidden variable, Neural Inf. Process. Syst. 10 (1997) 584–590.
[26] C. Merz, P. Murphy, D. Aha, UCI repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 1997. http://www.ics.uci.edu/~mlearn/MLRepository.html.
[27] M.J. Pazzani, Constructive induction of cartesian product attributes, Inf. Stat. Induction Sci. (1996).
[28] M.J. Pazzani, Searching for dependencies in Bayesian classifiers, Learning from Data: Artificial Intelligence and Statistics V, Springer, New York, 1996, pp. 239–248.
[29] J.M. Peña, I. Izarzugaza, J.A. Lozano, E. Aldasoro, P. Larrañaga, Geographical clustering of cancer incidence by means of Bayesian networks and conditional Gaussian networks, in: Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics, Morgan Kaufmann, San Francisco, CA, 2001, pp. 266–271.
[30] J.M. Peña, J.A. Lozano, P. Larrañaga, Learning Bayesian networks for clustering by means of constructive induction, Pattern Recognition Lett. 20 (11–13) (1999) 1219–1230.
[31] J.M. Peña, J.A. Lozano, P. Larrañaga, An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering, Pattern Recognition Lett. 21 (8) (2000) 779–786.

[32] J.M. Peña, J.A. Lozano, P. Larrañaga, Learning recursive Bayesian multinets for data clustering by means of constructive induction, Mach. Learn. (2001) to appear.
[33] J.M. Peña, J.A. Lozano, P. Larrañaga, I. Inza, Dimensionality reduction in unsupervised learning of conditional Gaussian networks, IEEE Trans. Pattern Anal. Machine Intell. 23 (6) (2001) 590–603.
[34] M. Peot, Geometric implications of the Naive Bayes assumption, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1996, pp. 414–419.
[35] R. Shachter, C. Kenley, Gaussian influence diagrams, Manage. Sci. 35 (1989) 527–550.
[36] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, R.S. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the Symposium on Computer Applications and Medical Care, IEEE Computer Society Press, Silver Spring, MD, 1988, pp. 261–265.
[37] B. Thiesson, C. Meek, D.M. Chickering, D. Heckerman, Learning mixtures of DAG models, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1998, pp. 504–513.