Dimensionality Reduction in Unsupervised Learning of Conditional Gaussian Networks

Jose Manuel Peña, Jose Antonio Lozano, Pedro Larrañaga, and Iñaki Inza

Abstract—This paper introduces a novel enhancement for unsupervised learning of conditional Gaussian networks that benefits from feature selection. Our proposal is based on the assumption that, in the absence of labels reflecting the cluster membership of each case of the database, those features that exhibit low correlation with the rest of the features can be considered irrelevant for the learning process. Thus, we suggest performing this process using only the relevant features. Then, every irrelevant feature is added to the learned model to obtain an explanatory model for the original database which is our primary goal. A simple and, thus, efficient measure to assess the relevance of the features for the learning process is presented. Additionally, the form of this measure allows us to calculate a relevance threshold to automatically identify the relevant features. The experimental results reported for synthetic and real-world databases show the ability of our proposal to distinguish between relevant and irrelevant features and to accelerate learning; however, still obtaining good explanatory models for the original database.

Index Terms—Data clustering, conditional Gaussian networks, feature selection, edge exclusion tests.

1 Introduction

One of the basic problems that arises in a great variety of fields, including pattern recognition, machine learning, and statistics, is the so-called *data clustering problem* [1], [2], [10], [11], [18], [22]. Despite the different interpretations and expectations it gives rise to, the generic data clustering problem involves the assumption that, in addition to the observed variables (also referred to as predictive attributes or, simply, features), there is a *hidden* variable. This last unobserved variable would reflect the cluster membership for every case in the database. Thus, the data clustering problem is also referred to as an example of learning from *incomplete data* due to the existence of such a hidden variable. Incomplete data represents a special case of *missing data* where all the missing entries are concentrated in a single variable: The hidden cluster variable. That is, we refer to a given database as incomplete when all the cases are unlabeled.

From the point of view adopted in this paper, the data clustering problem may be defined as the inference of the generalized joint probability density function for a given database. Concretely, we focus on learning *conditional Gaussian networks* for data clustering [25], [26], [27], [36], [37]. Roughly speaking, a conditional Gaussian network is a graphical model that encodes a *conditional Gaussian distribution* [25], [26], [27] for the variables of the domain. Then when applied to data clustering, it encodes a multivariate normal distribution for the observed variables conditioned on each state of the cluster variable.

As we aim to automatically recover the generalized joint probability density function from a given incomplete database by learning a conditional Gaussian network, this paper is concerned with the understanding of data clustering as a *description* task rather than a *prediction* task. Thus, in order to encode a description of the original database, the learned model must involve all the original features instead of a subset of them. When unsupervised learning algorithms focus on prediction tasks, *feature selection* has proven to be a valuable technique to increase the predictive ability of the elicited models. In this paper, we demonstrate that, even when focusing on description, feature selection (also known as *dimensionality reduction*) can be a profitable tool for improving the performance of unsupervised learning.

The general framework that we propose to show how unsupervised learning of conditional Gaussian networks can benefit from feature selection is straightforward and consists of three steps: 1) identification of the relevant features for learning, 2) unsupervised learning of a conditional Gaussian network from the database restricted to the relevant features, and 3) addition of the irrelevant features to the learned network to obtain an explanatory model for the original database. According to this framework, feature selection is considered a preprocessing step that should be accompanied by a postprocessing step to fulfill our objective. This postprocessing step consists of the addition of every irrelevant feature to the learned model to have a final model that encodes the generalized joint probability density function for the original data.

To completely define the framework, one should decide on the automatic dimensionality reduction scheme to identify the relevant features for learning. This paper introduces a simple *relevance measure* to assess the relevance of the features for the learning process in order to select a subset of them containing the most salient ones. Additionally, we propose a heuristic method to automatically qualify every feature as

- *The authors are with the Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, P.O. Box 649, E-20080 Donostia-San Sebastián, Spain. E-mail: ccbpepaj@si.ehu.es.*

Manuscript received 11 Oct. 2000; revised 31 Jan. 2001; accepted 6 Feb. 2001. Recommended for acceptance by P. Meer.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112968.

0162-8828/01/$10.00 © 2001 IEEE

completely relevant or irrelevant for the learning process. This is carried out by the automatic calculation of a *relevance threshold*. Those features with relevance measure values higher than the relevance threshold are considered relevant for the learning process, whereas the rest are qualified as irrelevant.

The experimental results reported in this paper show that the framework depicted above provides us with good explanatory models for the original database reducing the cost of the learning process as only relevant features are used in this process. In addition to its effectiveness, the simplicity of the automatic dimensionality reduction scheme that we propose represents a valuable advantage as it allows the framework to reduce the dimensionality of the database where to perform learning very efficiently. Besides, our scheme is not tied to any particular learning algorithm and, therefore, it can be adapted to most of them.

The remainder of this paper is organized as follows: In Section 2, we introduce conditional Gaussian networks for data clustering. Section 3 is dedicated to explaining in detail our automatic dimensionality reduction scheme. We present a new relevance measure as well as how to automatically discover the relevant and irrelevant features through the calculation of a relevance threshold. This section also presents how to fit our proposal into the unsupervised learning of conditional Gaussian networks under the framework already outlined. Some experimental results showing the ability of our proposal to identify the relevant features and to accelerate the learning process are compiled in Section 4. Finally, we draw conclusions in Section 5.

2 Conditional Gaussian Networks for Data Clustering

This section starts introducing the notation used throughout this paper. Then, we give a formal definition of conditional Gaussian networks. We also present the *Bayesian Structural EM algorithm* [13], which is used for explanatory purposes as well as in our experiments presented in Section 4 due to its good performance in unsupervised learning of conditional Gaussian networks.

2.1 Notation

We follow the usual convention of denoting 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 set of variables and the same boldface lowercase letter or letters to denote an assignment of a state to each variable in a given set. The generalized joint probability density function of $\mathbf{X}$ is represented as $\rho(\mathbf{x})$. Additionally, $\rho(\mathbf{x} \mid \mathbf{y})$ denotes the generalized conditional probability density function of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$. If all the variables in $\mathbf{X}$ are discrete, then $\rho(\mathbf{x}) = p(\mathbf{x})$ is the joint probability mass function of $\mathbf{X}$. Thus, $p(\mathbf{x} \mid \mathbf{y})$ denotes the conditional probability mass function of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$. On the other hand, if all the variables in $\mathbf{X}$ are continuous, then $\rho(\mathbf{x}) = f(\mathbf{x})$ is the joint probability density function of $\mathbf{X}$. Thus, $f(\mathbf{x} \mid \mathbf{y})$ denotes the conditional probability density function of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$.

2.2 Conditional Gaussian Networks

As we have already mentioned, when facing a data clustering problem we assume the existence of a random variable $\mathbf{X}$ partitioned as $\mathbf{X} = (\mathbf{Y}, C) = (Y_1, \ldots, Y_n, C)$ into a n-dimensional continuous variable $\mathbf{Y}$ and a unidimensional discrete hidden cluster variable C. $\mathbf{X}$ is said to have a conditional Gaussian distribution [25], [26], [27] if the distribution of $\mathbf{Y}$, conditioned on each state of C, is a multivariate normal distribution. That is,

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

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 $\mathbf{X}$ as a graphical model that encodes a conditional Gaussian distribution for $\mathbf{X}$ [25], [26], [27], [36], [37]. Essentially, CGNs belong to a class of mixed graphical models introduced for the first time by Lauritzen and Wermuth [27] and further developed in [25], [26]. This class groups models in which both discrete and continuous variables can be present and for which the conditional distribution of the continuous variables given the discrete variables is restricted to be multivariate Gaussian. More recently, CGNs have been successfully applied to data clustering [36], [37].

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

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

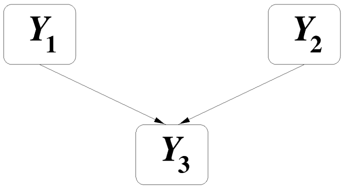
where $\mathbf{pa}(\mathbf{s})_i$ denotes the configuration of the parents of Y_i, $\mathbf{Pa}(\mathbf{s})_i$, consistent with $\mathbf{x}$. The local probability density functions and the multinomial distribution are those in the previous equation and we assume that they depend on a finite set of parameters $\boldsymbol{\theta}_{\mathbf{s}} \in \boldsymbol{\Theta}_{\mathbf{s}}$. Therefore, (2) can be rewritten as follows:

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

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

If $\mathbf{s}^h$ denotes the hypothesis that the conditional (in)-dependence assertions implied by s hold in the true generalized joint probability density function of $\mathbf{X}$, then we obtain from (3) that:

- **Model structure**



- **Multinomial distribution**

$p(c_1 \mid \boldsymbol{\theta}_{\mathbf{s}}, \mathbf{s}^h)$

$p(c_2 \mid \boldsymbol{\theta}_{\mathbf{s}}, \mathbf{s}^h) = 1 - p(c_1 \mid \boldsymbol{\theta}_{\mathbf{s}}, \mathbf{s}^h)$

- **Local probability density functions**

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

Fig. 1. Structure, local probability density functions, and multinomial distribution for a CGN with three continuous variables and one binary cluster variable.

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

In order to encode a conditional Gaussian distribution for $\mathbf{X}$, each local probability density function of a CGN should be a linear-regression model. Thus, when $C = c$:

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

where $\mathcal{N}(y; \mu, \sigma^2)$ is a univariate normal distribution with mean μ and standard deviation σ ($\sigma > 0$). Given this form, a missing arc from Y_j to Y_i implies that $b_{ji}^c = 0$ in the linear-regression model. When $C = c$, the local parameters are $\boldsymbol{\theta}_i^c = (m_i^c, \mathbf{b}_i^c, v_i^c)$, $i = 1, \dots, n$, where $\mathbf{b}_i^c = (b_{1i}^c, \dots, b_{i-1i}^c)^t$ is a column vector.

The interpretation of the components of the local parameters $\boldsymbol{\theta}_i^c$, $i = 1, \dots, n$, is as follows: Given $C = c$, m_i^c is the unconditional mean of Y_i, v_i^c is the conditional variance of Y_i given $\mathbf{Pa}(\mathbf{s})_i$, and b_{ji}^c, $j = 1, \dots, i-1$, is a linear coefficient reflecting the strength of the relationship between Y_j and Y_i. See Fig. 1 for an example of a CGN with three continuous variables and one binary cluster variable.

Note that the model structure is independent of the value of the cluster variable C, thus the model structure is the same for all the values of C. However, the parameters of the local probability density functions do depend on the value of C and they may differ from the distinct values of the variable C.

2.3 Learning CGNs from Incomplete Data

One of the methods for learning CGNs from incomplete data is the well-known Bayesian Structural EM (BS-EM) algorithm developed by Friedman in [13]. Due to its good performance, this algorithm has received special attention in the literature and has motivated several variants of itself [32], [34], [35], [41]. 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, $\mathbf{d} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, where every case is represented by an assignment to the n observed variables of the $n+1$ variables involved in the problem domain. So, there are $(n+1)N$ random variables that describe the database. Let $\mathbf{O}$ denote the set of observed variables, that is, the nN variables that have assigned values. Similarly, let $\mathbf{H}$ denote the set of hidden or unobserved variables, that is, the N variables that reflect the unknown cluster membership of each case of $\mathbf{d}$.

For learning CGNs from incomplete data, the BS-EM algorithm performs a search over the space of CGNs based on the well-known *EM algorithm* [7], [29] and direct optimization of the Bayesian score. As shown in Fig. 2, the

```
loop $l = 0, 1, \ldots$
  1. Run the EM algorithm to compute the MAP parameters $\widehat{\boldsymbol{\theta}}_{\mathbf{s}_l}$ for $\mathbf{s}_l$ given $\mathbf{o}$
  2. Perform search over model structures, evaluating each model structure by
     $Score(\mathbf{s} : \mathbf{s}_l) = E[\log \rho(\mathbf{h}, \mathbf{o}, \mathbf{s}^h) \mid \mathbf{o}, \widehat{\boldsymbol{\theta}}_{\mathbf{s}_l}, \mathbf{s}_l^h] = \sum_{\mathbf{h}} p(\mathbf{h} \mid \mathbf{o}, \widehat{\boldsymbol{\theta}}_{\mathbf{s}_l}, \mathbf{s}_l^h) \log \rho(\mathbf{h}, \mathbf{o}, \mathbf{s}^h)$
  3. Let $\mathbf{s}_{l+1}$ be the model structure with the highest score among these encountered in the search
  4. if $Score(\mathbf{s}_l : \mathbf{s}_l) = Score(\mathbf{s}_{l+1} : \mathbf{s}_l)$ then return $(\mathbf{s}_l, \widehat{\boldsymbol{\theta}}_{\mathbf{s}_l})$
```

Fig. 2. A schematic of the BS-EM algorithm.

BS-EM algorithm is comprised of two steps: An optimization of the CGN parameters and a structural search for model selection. Concretely, the BS-EM algorithm alternates between a step that finds the maximum a posteriori (MAP) parameters for the current CGN structure usually by means of the EM algorithm, and a step that searches over CGN structures. At each iteration, the BS-EM algorithm attempts to maximize the expected Bayesian score instead of the true Bayesian score.

As we are interested in solving data clustering problems of considerable size, the direct application of the BS-EM algorithm as it appears in Fig. 2 may be an unrealistic and inefficient solution. In our opinion, the reason for this possible inefficiency is that the computation of $Score(\mathbf{s} : \mathbf{s}_l)$ implies a huge computational expense as it takes account of every possible completion of the database. It is common to use a relaxed version of the presented BS-EM algorithm that just considers the most likely completion of the database to compute $Score(\mathbf{s} : \mathbf{s}_l)$ instead of considering every possible completion. Thus, this relaxed version of the BS-EM algorithm is comprised of the iteration of a parametric optimization for the current model and a structural search once the database has been completed with the most likely completion by using the best estimate of the generalized joint probability density function of the data so far (current model). That is, the posterior probability distribution of the cluster variable C for each case of the database, $p(c \mid \mathbf{y}_i, \widehat{\boldsymbol{\theta}}_{\mathbf{s}_l}, \mathbf{s}_l^h)$, is calculated. Then, the case is assigned to the cluster where the maximum of the posterior probability distribution of C is reached. We use this relaxed version in our experiments of Section 4.

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 considering all 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 and the factorization properties of the Bayesian score for complete data. However, any structural search procedure that exploits these properties can be used.

The log *marginal likelihood* of the expected complete data, $\log \rho(\mathbf{d} \mid \mathbf{s}^h)$, is usually chosen as the score to guide the structural search. We make use of it in our experiments. According to [15], under the assumptions that 1) the database restricted to the cluster variable C, $\mathbf{d}^C$, is a multinomial sample, 2) the database $\mathbf{d}$ is complete, and 3) the parameters of the multinomial distribution of C are independent and follow a Dirichlet distribution; we have that:

$$\begin{aligned}\rho(\mathbf{d} \mid \mathbf{s}^h) &= \prod_{l=1}^{N} \rho(\mathbf{x}_l \mid \mathbf{x}_1, \ldots, \mathbf{x}_{l-1}, \mathbf{s}^h) \\ &= \prod_{l=1}^{N} p(c_l \mid \mathbf{x}_1, \ldots, \mathbf{x}_{l-1}, \mathbf{s}^h) f(\mathbf{y}_l \mid c_l, \mathbf{x}_1, \ldots, \mathbf{x}_{l-1}, \mathbf{s}^h) \\ &= p(\mathbf{d}^C \mid \mathbf{s}^h) \prod_{l=1}^{N} f(\mathbf{y}_l \mid c_l, \mathbf{y}_1, \ldots, \mathbf{y}_{l-1}, \mathbf{s}^h) \\ &= p(\mathbf{d}^C \mid \mathbf{s}^h) \prod_{c \in Val(C)} f_c(\mathbf{d}^{\mathbf{Y},\,c} \mid \mathbf{s}^h),\end{aligned} \tag{6}$$

where $\mathbf{d}^{\mathbf{Y},\,c}$ is the database $\mathbf{d}$ restricted to the continuous variables $\mathbf{Y}$ and to cases where $C = c$, and $Val(C)$ is the set of values that the cluster variable C can take. The term $p(\mathbf{d}^C \mid \mathbf{s}^h)$ 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 [5]. Moreover, each term of the form $f_c(\mathbf{d}^{\mathbf{Y},\,c} \mid \mathbf{s}^h)$, for all $c \in Val(C)$, represents the marginal likelihood of a domain containing only continuous variables under the assumption that the continuous data is sampled from a multivariate normal distribution. Then, these terms can be evaluated in factorable closed form under some reasonable assumptions according to [15], [16], [19].

3 Automatic Dimensionality Reduction in Unsupervised Learning of CGNs

This section is devoted to the detailed presentation of a new automatic dimensionality reduction scheme applied to unsupervised learning of CGNs. The section starts with an introductory revision on the general problem of feature selection and a brief discussion on some of the problems that appear when adapting supervised feature selection to the unsupervised paradigm.

3.1 From Supervised to Unsupervised Feature Selection

In many data analysis applications, the size of the data can be large. The largeness can be due to an excessive number of features, the huge number of instances, or both. For learning algorithms to work efficiently and even sometimes effectively, one may need to reduce the data size. Feature selection has proven to be a valuable technique to achieve such a reduction of the dimensionality of the data by selecting a subset of features on which to focus the attention in the subsequent learning process.

In its general form, feature selection is considered a problem of searching for an optimal subset of the original features according to a certain criterion [3], [23], [28]. The criterion specifies the details of measuring the goodness of feature subsets as well as the relevance of each feature. The choice of a criterion is influenced by the purpose of feature selection. However, what is shared by the different purposes is the desire of improving the performance of the subsequent learning algorithm usually in terms of the speed of learning, the predictive ability of the learned models, and/or the comprehensibility of the learned models.

Roughly speaking, feature selection involves an algorithm to explore the space of potential feature subsets and an evaluation function to measure the quality of these feature subsets. Since the space of all feature subsets of n features has size 2^n, feature selection mechanisms typically perform a nonexhaustive search. One of the most popular techniques is the use of a simple hill-climbing search known as *sequential selection* which may be either *forward* or *backward* [3], [23], [28]. In the former, the search starts with an empty set of selected features and, at each time, it adds the best feature among unselected ones according to the evaluation function. The process stops when no further improvement can be made. Similarly, backward sequential selection begins with the full set of features and, at each time, it removes the worst feature based on the evaluation function until no improvement is found. As it is addressed by Doak [9], feature selection mechanisms based on sequential selection can require a great deal of processing time in databases with a large number of features. Also, more complex and effective search algorithms can be used to explore the space of potential feature subsets. The main advantage of these algorithms over sequential selection is that they avoid getting stuck in local maxima by means of randomness. However, these approaches usually involve a huge computational effort. One of the recent works in the field is reported in [20]. In this paper, the authors propose exploring the space of feature subsets according to an evolutionary, population-based, randomized search algorithm which represents an instance of the Estimation of Distribution Algorithm (EDA) approach [24].

In [23], the authors distinguish two approaches to the evaluation function for feature selection: *wrapper* and *filter*. The wrapper approach implies a search for an optimal feature subset tailored to the performance function of the subsequent learning algorithm. That is, it considers feedback from the performance function of the particular subsequent learning algorithm as part of the function to evaluate feature subsets. On the other hand, the filter approach relies on intrinsic properties of the data that are presumed to affect the performance of the learning algorithm but they are not a direct function of its performance. Then, the filter approach tries to assess the merits of the different feature subsets from the data, ignoring the subsequent learning algorithm.

When applied to supervised learning, the main objective of feature selection is the improvement of the classification accuracy or class label predictive accuracy of the models elicited by the subsequent learning algorithm considering only the relevant features for the task. Independently of the approach used, both filter and wrapper approaches require the class labels to be present in the data in order to carry out feature selection. Filter approaches evaluate feature subsets usually by assessing the correlation of every feature with the class label by using different measures [3], [28]. On the other hand, wrapper approaches rely on the performance of the learning algorithm itself by measuring the classification accuracy on a validation set to evaluate the goodness of the different feature subsets [3], [23], [28]. There is some evidence from supervised feature selection research that wrapper approaches outperform filter approaches [21].

Although feature selection is a central problem in data analysis as suggested by the growing amount of research in this area, the vast majority of the research has been carried out under the supervised learning paradigm (*supervised* feature selection), paying little attention to unsupervised learning (*unsupervised* feature selection). Only a few works exist addressing the latter problem. In [6], the authors present a method to rank features according to an unsupervised entropy measure. Their algorithm works as a filter approach plus a backward sequential selection search. Devaney and Ram [8] propose a wrapper approach combined with either a forward or a backward sequential selection search to perform conceptual clustering. In [39], Talavera introduces a filter approach combined with a search in one step and a wrapper approach combined with either a forward or a backward sequential selection search as feature selection mechanisms in hierarchical clustering of symbolic data. The filter approach uses the feature dependence measure defined by Fisher [11]. Whereas the performance criterion considered in [39] is the *multiple predictive accuracy* measured by the average accuracy of predicting the values of each feature present in the testing data, [40] applies the mechanism comprised of a filter approach and a search in one step presented in [39] to feature selection in conceptual clustering of symbolic data considering the class label predictive accuracy as performance criterion.

In our opinion, two are the main problems to translate supervised feature selection into unsupervised feature selection. First, the absence of class labels reflecting the membership for every case in the database that is inherent to the unsupervised paradigm makes impossible the use of the same evaluation functions as in supervised feature selection. Second, there is not a standard accepted performance task for unsupervised learning. Due to this lack of a unified performance criterion, the meaning of optimal feature subset may vary from task to task. A natural solution to both problems is proposed in [39] by interpreting the performance task of unsupervised learning as the multiple predictive accuracy. This seems a reasonable approach because it extends the standard accepted performance task for supervised learning to unsupervised learning. Whereas the former learning comprises the prediction of only one feature, the class, from the knowledge of many, the latter aims the prediction of many features from the knowledge of many [12]. On the other hand, [6], [8], [40] evaluate their unsupervised feature selection mechanisms by measuring the class label predictive accuracy of the learned models over the cases of a testing set after having performed learning in a training set where the class labels were masked out. The speed of learning and the comprehensibility of the learned models are also studied in [8], [39], although they are considered less important performance criteria.

3.2 How Learning CGNs for Data Clustering Benefits from Feature Selection

Our motivation to perform unsupervised feature selection differs from the motivation of the previously referred papers due to our distinct point of view over the data clustering problem. When the learned models for data clustering are primarily evaluated regarding their multiple or class label predictive accuracy, as it occurs in [6], [8], [39], [40], feature selection has proven to be a valuable technique for reducing the dimensionality of the database where learning is performed. This usually pursues an improvement of the performance of the learned models considering only the relevant features for the task. However, when the main goal of data clustering, as it happens in this paper, is description rather than prediction, the learned models must involve all the features that the original database has in order to encode a description of this database.

It is well-known that unsupervised learning of CGNs for solving data clustering problems is a difficult and time consuming task, even more so when focusing on description as all the original features are usually considered in the learning process. With the aim to solve these handicaps, we propose a framework where learning CGNs for data clustering benefits from feature selection. The framework is straightforward and consists of three steps: 1) identification of the relevant features for learning, 2) unsupervised learning of a CGN from the database restricted to the relevant features, and 3) addition of the irrelevant features to the learned CGN for obtaining an explanatory model for the original database. Thus, feature selection is considered a preprocessing step that should be accompanied by a postprocessing step to achieve our objective. The postprocessing step consists of the addition of every irrelevant feature to the elicited model as conditionally independent of the rest given the cluster variable.

To make the framework applicable for unsupervised learning of CGNs, we should define relevance. However, the meaning of relevance depends on the particular purpose of dimensionality reduction due to the lack of a unified performance criterion for data clustering. In our concrete case, the objective of reducing the dimensionality of the databases when learning CGNs for data clustering is to decrease the cost of the learning process while still obtaining good explanatory models for the original data. The achievement of such a goal can be assessed by comparing, in terms of explanatory power and runtime of the learning process, a CGN learned from the given original database and a CGN elicited when using dimensionality reduction in the learning process.

Such an assessment of the achievement of our objective leads us to make the following assumption on the consideration of a feature as either relevant or irrelevant for the learning process: In the absence of labels reflecting the cluster membership of each case of the database, those features that exhibit low correlation with the rest of the features can be considered irrelevant for the learning process. Implicitly, this assumption defines relevance according to our purpose to perform dimensionality reduction. It is important to note that the assumption is independent of any clustering of the data, so, it can be readily applied without requiring a previous clustering of the database.

The justification of the previous assumption is straightforward. Features low correlated with the rest are likely to remain conditionally independent of the rest of the features given the cluster variable when learning a CGN from the original database. Thus, a CGN elicited from the original database restricted to features highly correlated with the rest is likely to encode the same set of conditional dependence assertions as a CGN learned from the original database. The parameters for the local probability density functions of the features that appear in both CGNs should be similar as well. Furthermore, if low correlated features are added to that CGN elicited from the restricted database as conditionally independent of the rest given the cluster variable, then this final CGN is likely to encode the same set of conditional dependence and independence assertions as the CGN learned from the original data. Thus, the explanatory power of both CGNs should be almost the same as the models are likely to be very similar.

Some other works that have successfully made use of a similar assumption are [11], [39], [40]. Although the three works present the assumption in its general form, they only validate it for conceptual clustering of symbolic data. Our paper is the first, to our knowledge, that verifies it for continuous domains.

3.2.1 Relevance Measure

In order to assess the relevance of Y_i, $i = 1, \ldots, n$, for learning, we propose evaluating the following simple and, thus, efficient relevance measure:

$$\sum_{j=1,\, j \neq i}^{n} \frac{-N \log(1 - r_{ij|rest}^2)}{n-1}, \tag{7}$$

where n is the number of features in the database, N is the number of cases in the database, and $r_{ij|rest}$ is the sample partial correlation of Y_i and Y_j adjusted for the remainder variables. This last quantity can be expressed in terms of the maximum-likelihood estimates of the elements of the inverse variance matrix as $r_{ij|rest} = -\hat{w}_{ij}(\hat{w}_{ii}\hat{w}_{jj})^{-\frac{1}{2}}$ [43].

Then, the relevance measure value for each feature Y_i, $i = 1, \ldots, n$, is calculated as the average likelihood ratio test statistic for excluding an edge between Y_i and any other feature in a graphical Gaussian model [38]. This means that those features likely to remain conditionally independent of the rest given the cluster variable as learning progresses receive low relevance measure values. Thus, this measure shows a reasonable behavior according to our definition of relevance.

3.2.2 Relevance Threshold

After having calculated the relevance measure value for every feature of the database, a decreasing relevance

```
Evaluate the relevance measure for each feature $Y_i$, $i = 1, \dots, n$
Calculate the relevance threshold
Let $\mathbf{Y}^{Rel}$ be the feature subset containing only the relevant features
loop $l = 0, 1, \dots$
    1. Run the EM algorithm to compute the MAP parameters $\widehat{\boldsymbol{\theta}}_{\mathbf{s}_l^{Rel}}$ for $\mathbf{s}_l^{Rel}$ given $\mathbf{o}^{Rel}$
    2. Perform search over model structures, evaluating each model structure by
       $Score(\mathbf{s}^{Rel} : \mathbf{s}_l^{Rel}) = E[\log \rho(\mathbf{h}, \mathbf{o}^{Rel}, \mathbf{s}^{Rel^h}) \mid \mathbf{o}^{Rel}, \widehat{\boldsymbol{\theta}}_{\mathbf{s}_l^{Rel}}, \mathbf{s}_l^{Rel^h}]$
       $= \sum_{\mathbf{h}} p(\mathbf{h} \mid \mathbf{o}^{Rel}, \widehat{\boldsymbol{\theta}}_{\mathbf{s}_l^{Rel}}, \mathbf{s}_l^{Rel^h}) \log \rho(\mathbf{h}, \mathbf{o}^{Rel}, \mathbf{s}^{Rel^h})$
    3. Let $\mathbf{s}_{l+1}^{Rel}$ be the model structure with the highest score among these encountered in the search
    4. exit loop when $Score(\mathbf{s}_l^{Rel} : \mathbf{s}_l^{Rel}) = Score(\mathbf{s}_{l+1}^{Rel} : \mathbf{s}_l^{Rel})$
Let $\mathbf{s}_{final}$ be the final model obtained after adding the irrelevant features to $\mathbf{s}_l^{Rel}$
Calculate the MAP parameters $\widehat{\boldsymbol{\theta}}_{\mathbf{s}_{final}}$ for $\mathbf{s}_{final}$
Return $(\mathbf{s}_{final}, \widehat{\boldsymbol{\theta}}_{\mathbf{s}_{final}})$
```

Fig. 3. A schematic of how to fit our automatic dimensionality reduction scheme into the BS-EM algorithm under the framework presented.

ranking of the features can be obtained. Now, we would like to know how many of them are needed to perform learning appropriately, that is, we would like to identify, in the relevance ranking, the relevant features for the learning process. If we knew that only k features were needed, we could simply choose the first k features in our relevance ranking, namely, those k features with the highest relevance measure values. However, to have this kind of knowledge is not at all usual. We propose a novel and automatic solution for this problem.

The relevance measure value for each feature Y_i, $i = 1, \dots, n$, can be interpreted as the average value of the likelihood ratio test statistic for excluding a single edge between Y_i and any other feature in a graphical Gaussian model. Thus, we propose the following heuristic: The relevance threshold is calculated as the rejection region boundary for an edge exclusion test in a graphical Gaussian model for the likelihood ratio test statistic (see [38] for details). This heuristic agrees with our purpose to perform dimensionality reduction as it qualifies as irrelevant those features likely to remain conditionally independent of the rest given the cluster variable as learning progresses. As shown in [38], the distribution function of the likelihood ratio test statistic is as follows:

$$F(x) = G_{\mathcal{X}}(x) - \frac{1}{2}(2n+1)x\frac{1}{\sqrt{2\pi}}x^{-\frac{1}{2}}e^{-\frac{1}{2}x}N^{-1}, \qquad (8)$$

where $G_{\mathcal{X}}(x)$ is the distribution function of a $\mathcal{X}_1^2$ random variable. Thus, for a 5 percent test, the rejection region boundary (which is considered our relevance threshold) is given by the resolution of the following equation:

$$0.95 = G_{\mathcal{X}}(x) - \frac{1}{2}(2n+1)x\frac{1}{\sqrt{2\pi}}x^{-\frac{1}{2}}e^{-\frac{1}{2}x}N^{-1}. \qquad (9)$$

By a simple manipulation, the resolution of the previous equation turns into finding the root of an equation. The Newton-Raphson method, used in our experiments, is only an example of suitable methods for solving the equation. Only those features that exhibit relevance measure values higher than the relevance threshold are qualified as relevant for the learning process. The rest of the features are treated as irrelevant.

3.2.3 Fitting Automatic Dimensionality Reduction into Learning

In this section, we present how to fit our automatic dimensionality reduction scheme into the BS-EM algorithm under the general framework previously introduced. However, it should be noticed that our scheme is not coupled to any particular learning algorithm and it could be adapted to most of them.

Fig. 3 shows that, after the preprocessing step that consists of our automatic dimensionality reduction scheme, the BS-EM algorithm is applied as usual but restricting the original database to the relevant features, $\mathbf{Y}^{Rel}$, and the hidden cluster variable C. That is, the database where learning is performed consists of N cases, $\mathbf{d}^{Rel} = \{\mathbf{x}_1^{Rel}, \dots, \mathbf{x}_N^{Rel}\}$, where every case is represented by an assignment to the relevant features. So, there are $(r+1)N$ random variables that describe the database, where r is the number of relevant features ($r = |\mathbf{Y}^{Rel}|$). We denote the set of observed variables restricted to the relevant features and the set of hidden variables restricted to the relevant features by $\mathbf{O}^{Rel}$ ($|\mathbf{O}^{Rel}| = rN$) and $\mathbf{H}$ ($|\mathbf{H}| = N$), respectively. Obviously, in Fig. 3, $\mathbf{s}_l^{Rel}$ represents the model structure only when the relevant features are considered in the learning process, and $\mathbf{s}_l^{Rel^h}$ denotes the hypothesis that the conditional (in)dependence assertions implied by $\mathbf{s}_l^{Rel}$ hold in the true joint probability density function of $\mathbf{Y}^{Rel}$.

Learning ends with the postprocessing step that comprises the addition of every irrelevant feature to the model returned by the BS-EM algorithm as conditionally independent of the rest given the cluster variable. This results in an explanatory model for the original database. The local parameters for those nodes of the final model associated to the irrelevant features can be easily estimated after completing the original database $\mathbf{d}$ with the last completion of the restricted database $\mathbf{d}^{Rel}$.

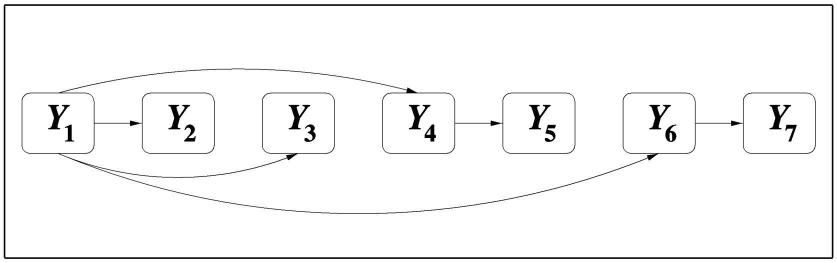


Fig. 4. Example of a TANB model structure with seven predictive attributes.

4 Experimental Evaluation

This section is dedicated to showing the ability of our proposal to perform an automatic dimensionality reduction that accelerates unsupervised learning of CGNs without degrading the explanatory power of the final models. In order to reach such a conclusion, we perform two sorts of experiments in synthetic and real-world databases. The first evaluates the relevance measure introduced in Section 3.2.1 as a means to assess the relevance of the features for the learning process. The second evaluates the ability of the relevance threshold calculated as it appears in Section 3.2.2 to automatically distinguish between relevant and irrelevant features for learning.

As we have addressed, we use the BS-EM algorithm as our unsupervised learning algorithm. In the current experiments, we limit the BS-EM algorithm to learning *Tree Augmented Naive Bayes (TANB) models* [14], [30], [36]. This is a sensible and usual decision to reduce the otherwise large search space of CGNs. Moreover, this allows us to efficiently solve data clustering problems of considerable size as it is well-known the difficulty involved in learning densely connected CGNs from large databases and the painfully slow probabilistic inference when working with these.

TANB models constitute a class of compromise CGNs defined by the following condition: Predictive attributes may have, at the most, one other predictive attribute as a parent. Fig. 4 shows an example of a TANB model structure. TANB models are CGNs where an interesting trade-off between efficiency and effectiveness is achieved, that is, a balance between the cost of the learning process and the quality of the learned CGNs [36].

4.1 Databases Involved

Two synthetic and two real-world databases are involved in our experimental evaluation. The knowledge of the CGNs used to generate the synthetic databases allows us to assess accurately the achievement of our objectives. Besides, the real-world databases provide us with a more realistic evaluation framework.

To obtain the two synthetic databases, we constructed two TANB models of different complexity to be sampled. The first TANB model involved 25 predictive continuous attributes and one three-valued cluster variable. The first 15 of the 25 predictive attributes were relevant and the rest irrelevant. The 14 arcs between the relevant attributes were randomly chosen. The unconditional mean of every relevant attribute was fixed to zero for the first value of the cluster variable, four for the second, and eight for the third. The linear coefficients were randomly generated in the interval $[-1, 1]$ and the conditional variances were fixed to one (see (5)). The multinomial distribution for the cluster variable C was uniform. Every irrelevant attribute followed a univariate normal distribution with mean zero and variance one for each of the three values of the cluster variable.

The second TANB model involved 30 predictive continuous attributes and one three-valued cluster variable. The first 15 of the 30 predictive attributes were relevant and the rest irrelevant. The 14 arcs between the relevant attributes were randomly chosen. The unconditional mean of every relevant attribute was fixed to zero for the first value of the cluster variable, four for the second, and eight for the third. The linear coefficients were randomly generated in the interval $[-1, 1]$, and the conditional variances were fixed to two (see (5)). The multinomial distribution for the cluster variable C was uniform. Every irrelevant attribute followed a univariate normal distribution with mean zero and variance five for each of the three values of the cluster variable. This second model was considered more complex than the first due to the higher degree of overlapping between the probability density functions of each of the clusters and the higher number of irrelevant attributes.

From each of these two TANB models, we sampled 4,000 cases for the learning databases and 1,000 cases for the testing databases. In the forthcoming, the learning databases sampled from these two TANB models will be referred to as synthetic1 and synthetic2, respectively. Obviously, we discarded all the entries corresponding to the cluster variable for the two learning databases and the two testing databases.

Another source of data for our evaluation consisted of two well-known real-world databases from the UCI repository of Machine Learning databases [33]:

- Waveform which is an artificial database consisting of 40 predictive features. The last 19 predictive attributes are noise attributes which turn out to be irrelevant for describing the underlying three clusters. We used the data set generator from the UCI repository to obtain 4,000 cases for learning and 1,000 cases for testing.
- Pima which is a real database containing 768 cases and eight predictive features. There are two clusters. We used the first 700 cases for learning and the last 68 cases for testing.

The first database was chosen due to our interest in working with databases of considerable size (thousands of cases and tens of features). In addition to this, it represented an opportunity to evaluate the effectiveness of our approach as the true irrelevant features were known beforehand. The second database, considerably shorter in both the number of cases and the number of features, was chosen to get feedback on the scalability of our dimensionality reduction scheme. Obviously, we deleted all the cluster entries for the two learning databases and the two testing databases.

4.2 Performance Criteria

There exist two essential purposes for focusing on the explanatory power or *generalizability* of the learned models. The first purpose is to summarize the given databases into the learned models. The second purpose is to elicit models which are able to predict unseen instances [28]. Thus, the explanatory power of the learned CGNs should be assessed by evaluating the achievement of both purposes. The $\log$ marginal likelihood, sc_final, and the multiple predictive accuracy, L(test), of the learned CGNs seem to be sensible performance measures for the first and the second purpose, respectively. The multiple predictive accuracy is measured as the logarithmic scoring rule of Good [17]:

$$\mathrm{L(test)} = \frac{1}{|\mathbf{d}_{test}|} \sum_{\mathbf{y} \in \mathbf{d}_{test}} \log f(\mathbf{y} \mid \boldsymbol{\theta}_{\mathbf{s}}, \mathbf{s}^h), \qquad (10)$$

where $\mathbf{d}_{test}$ is a set of test cases and $|\mathbf{d}_{test}|$ is the number of test cases. The higher the value for this criterion, the higher the multiple predictive accuracy of the learned CGNs. Note that L(test) is not the primary performance measure but one of the two measures to assess the explanatory power of the learned CGNs. When focusing on description, L(test) is extremely necessary to detect models that, suffering from overfitting, have high sc_final values although they are not able to generalize the learning data to unseen instances.

It should be noted that (10) represents a kind of probabilistic approach to the standard multiple predictive accuracy understanding the latter as the average accuracy of predicting the value of each feature present in the testing data. When the data clustering problem is considered as the inference of a generalized joint probability density function from the learning data via unsupervised learning of a CGN, the probabilistic approach presented in (10) is more appropriate than the standard multiple predictive accuracy. This can be illustrated with a simple example. Let us imagine two different CGNs that exhibit the same standard multiple predictive accuracy but different multiple predictive accuracy measured as the logarithmic scoring rule of Good. This would reflect that the generalized joint probability density functions encoded by the two CGNs are different. Moreover, this would imply that one of the two CGNs generalizes the learning data to unseen instances better (i.e., the likelihood of the unseen instances is higher) than the other, although their standard multiple predictive accuracy is the same. Thus, the standard multiple predictive accuracy would not be an appropriate performance criterion in this context as it would be unable to distinguish between these two models. Some other works that have made use of the logarithmic scoring rule of Good to assess the multiple predictive accuracy are [31], [34], [36], [37], [41].

The runtime of the overall learning process, runtime, is also considered as valuable information. Every runtime reported includes the runtimes of the preprocessing step (dimensionality reduction), learning algorithm, and post-processing step (addition of the irrelevant features).

All the results reported are averaged over 10 independent runs for the synthetic1, synthetic2, and waveform databases, and over 50 independent runs for the pima database due to its shorter size. The experiments are run on a Pentium 366 MHz computer.

4.3 Results: Relevance Ranking

Fig. 5 plots the relevance measure values for the features of each of the four databases considered. Additionally, it shows the relevance threshold (dashed line) for each database. In the case of the synthetic databases, the 10 true irrelevant features of the synthetic1 database and the 15 of the synthetic2 database clearly appear with the lowest relevance measure values.

In the case of the waveform database, it may be interesting to compare the graph of Fig. 5 with other graphs reported in [4], [40], [42] for the same database. Caution should be used as a detailed comparison is not advisable due to the fact that relevance is defined in different ways depending on the particular purpose of each of these works. Moreover, the work by Talavera [40] is limited to conceptual clustering of symbolic data, then, the original waveform database was previously discretized. However, it is noticeable that the 19 true irrelevant features appear plotted with low relevance values in the four graphs. Although the shape of the graphs restricted to the 21 relevant features varies for the three works reported ([4], [40], [42]), these agree with our graph and consider the first and last few of these relevant features less important than the rest of the 21. The shape of our graph is slightly closer to those that appear in [4], [42] than to the one plotted in [40].

Then, we can conclude that the relevance measure proposed exhibits a desirable behavior for the databases where the true irrelevant features are known as it clearly assigns low relevance values to them. The following section evaluates if these values are low enough to automatically distinguish between relevant and irrelevant features through the calculation of a relevance threshold.

Fig. 6 shows the $\log$ marginal likelihood (sc_final) and multiple predictive accuracy (L(test)) of the final CGNs for the four databases considered as functions of the number of features selected as relevant for learning. In addition to this, Fig. 7 reports on the runtime needed to learn the final CGNs as a function of the number of features selected as relevant for learning. The selection of k features as relevant means the selection of the k first features of the decreasing relevance ranking obtained for the features of each concrete database according to their relevance measure values. Thus, in this first part of the experimental evaluation, we do not perform an automatic dimensionality reduction. Instead, we aim to study performance as a function of the number of features involved in learning. This allows us to evaluate the ability of our relevance measure to assess the relevance of the features for the learning process.

In general terms, Fig. 6 confirms that our relevance measure is able to induce an effective decreasing relevance

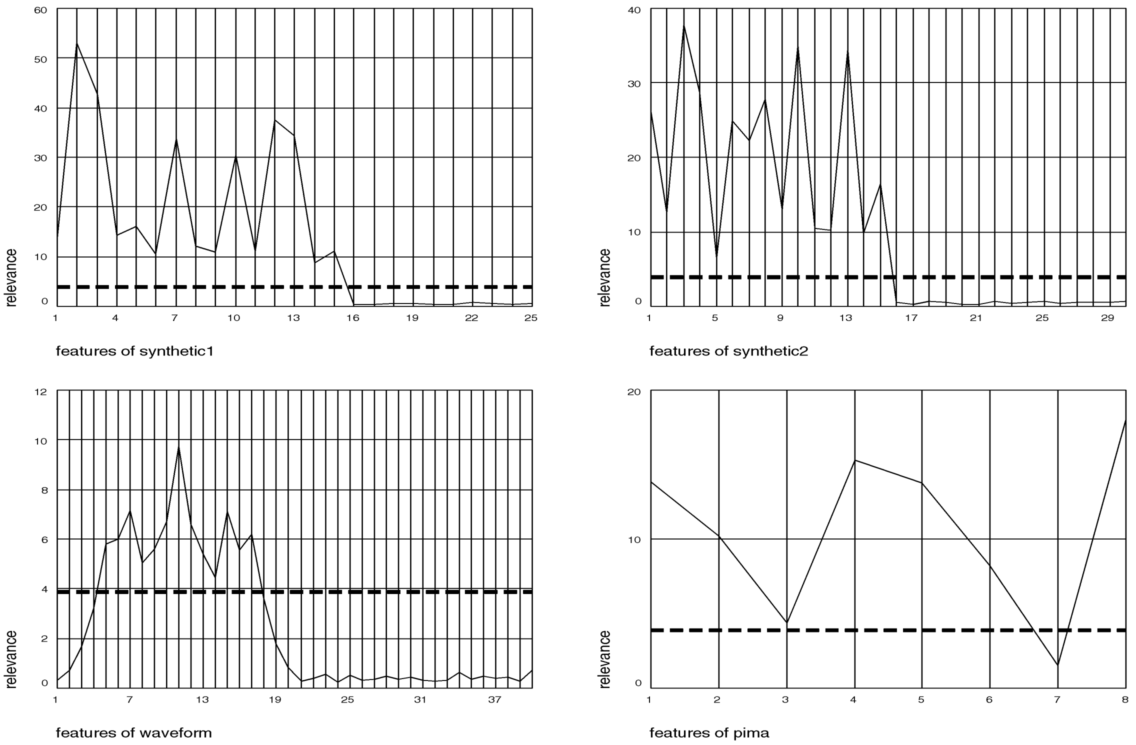


Fig. 5. Relevance measure values for the features of the databases used. The dashed lines correspond to the relevance thresholds.

ranking of the features of each database considered. That is, the addition of the features that have low relevance measure values (last features of the rankings) does not imply a significant increase in the quality of the final models, even in some cases, it hurts the explanatory power. Thus, this figure confirms that the assumption that low correlated features are irrelevant for the learning process works very well on the continuous domains considered. On the other hand, the addition of these irrelevant features tends to increase the cost of the learning process measured as runtime (see Fig. 7).

Particularly interesting are the results for the synthetic databases where the original models are known. The selection of true irrelevant features to take part in learning does not produce better models but increases the runtime of the learning process. Also, it is known that the last 19 of the 40 features of the waveform database are true irrelevant features. According to the relevance measure values for the features of the waveform database (see Fig. 5), all the 19 true irrelevant features would appear in the last 21 positions of the decreasing relevance ranking. Furthermore, it can be appreciated from Fig. 6 that the addition of these 19 irrelevant features does not significantly increase the explanatory power of the final CGNs. The results obtained for the pima database, where there is no knowledge on the existence of true irrelevant features, share the fact that using all the features in the learning process degrades the quality of the final models as well as makes the learning process slower. Thus, the explanatory power of the final CGNs appears to be not monotonic with respect to the addition of features as relevant for learning. Hence, the need for automatic tools for discovering irrelevant features that may degrade the effectiveness and enlarge the runtime of learning.

4.4 Results: Automatic Dimensionality Reduction

Fig. 5 shows the relevance threshold (dashed line) calculated as it appears in Section 3.2.2 for each of the databases considered. Only those features that exhibit relevance measure values higher than the relevance threshold are qualified as relevant. The rest of the features are considered irrelevant for learning.

It is interesting to notice that, for the two synthetic databases, all the true irrelevant features are identified independently of the complexity of the sampled model. It should be remembered that the synthetic2 database was sampled from a model more complex than the one used to generate the synthetic1 database. The results obtained for the waveform database are also specially appealing as the 19 true irrelevant features are correctly identified. Moreover, our scheme considers eight features of the remainder 21 features also as irrelevant. This appears to be a sensible decision as these eight features correspond to the first four and the last four of the 21 relevant features. Remember that [4], [40], [42] agree in this point: The first and last few of the 21 relevant features are less important than the rest of relevant features.

Table 1 compares, for the four databases considered, the performance achieved when no dimensionality reduction is carried out and the performance achieved when our automatic dimensionality reduction scheme is applied to learn CGNs. The column relevant indicates the number of relevant features automatically identified by our scheme for each database (see Fig. 5). It clearly appears from the table that our scheme is able to automatically set up a relevance threshold that induces a saving in runtime but still obtains good explanatory models. The application of our scheme as a

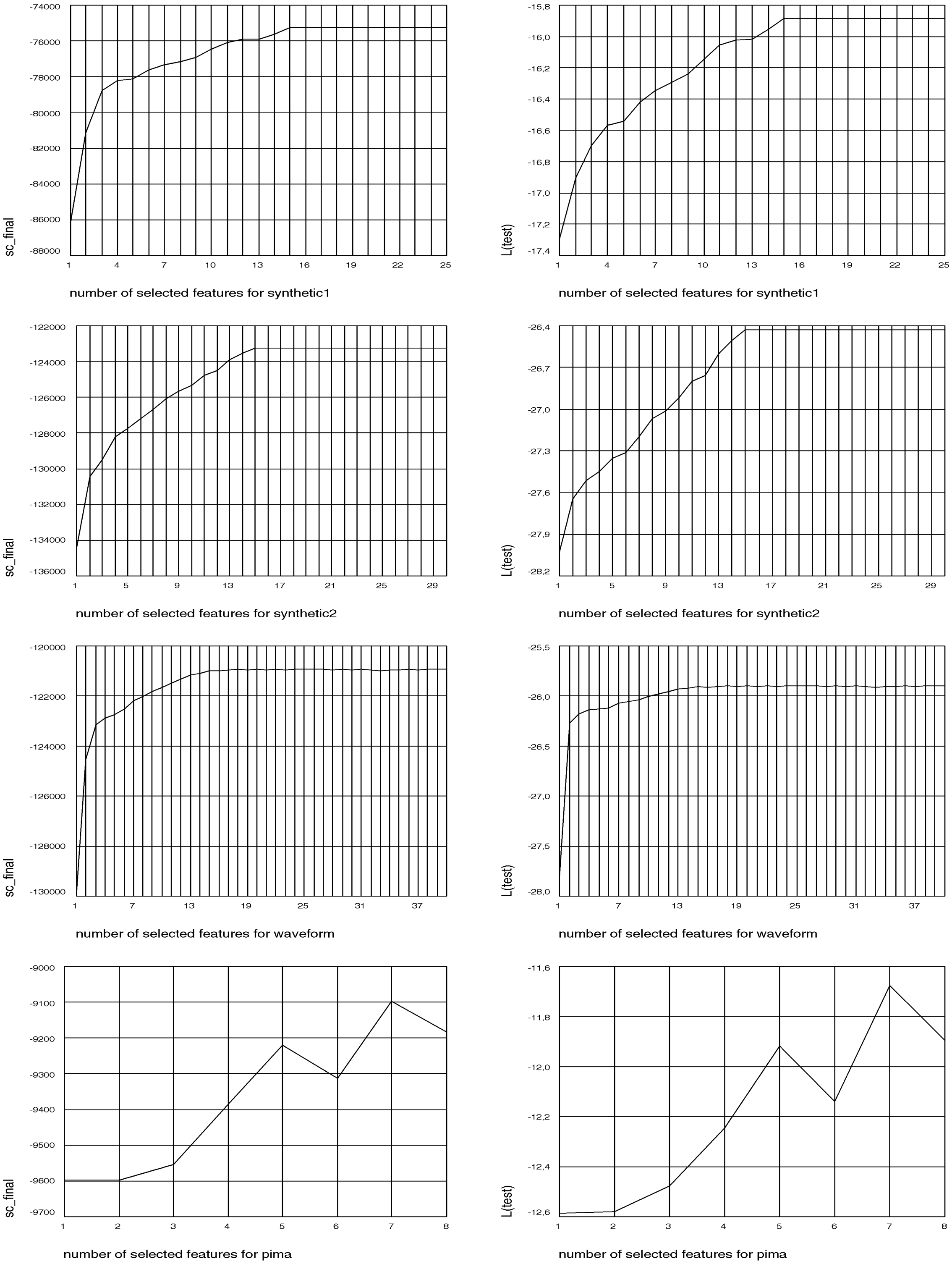


Fig. 6. log marginal likelihood (sc_final) and multiple predictive accuracy (L(test)) of the final CGNs for the databases used as functions of the number of features selected as relevant from a decreasing relevance ranking.

preprocessing step for the BS-EM algorithm (Fig. 3) provides us with a saving of runtime over the original BS-EM algorithm that achieves 22 percent for the synthetic1 database and 30 percent for the synthetic2 database. Moreover, the explanatory power of the CGNs elicited from the original synthetic databases and the CGNs obtained when using the automatic dimensionality reduction scheme is exactly the same.

For the waveform database, our automatic dimensionality reduction scheme proposes a reduction of the number of features of 68 percent: Only 13 out of the 40 original features are considered relevant. This reduction induces a gain in terms of runtime of 58 percent, whereas our scheme does not significantly hurt the quality of the learned models. On the other hand, the CGNs learned with the help of our automatic

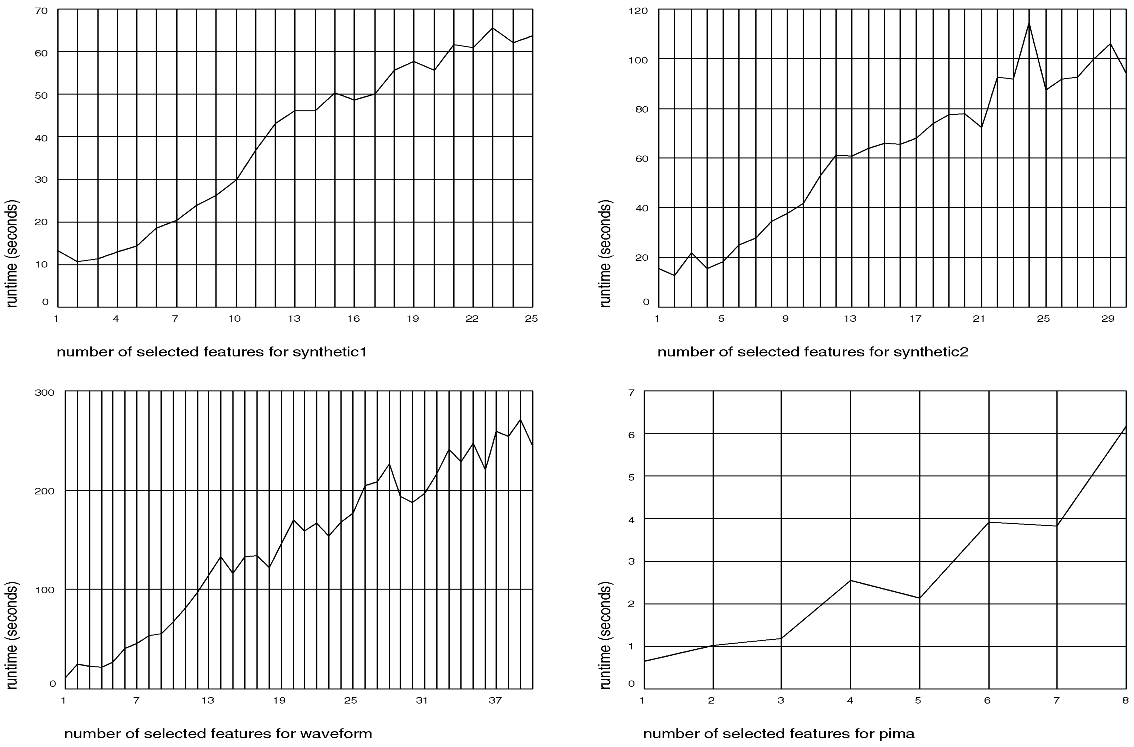


Fig. 7. Runtime needed to learn the final CGNs for the databases used as a function of the number of features selected as relevant from a decreasing relevance ranking.

dimensionality reduction scheme from the pima database exhibit, on average, a more desirable behavior than the CGNs elicited from the original pima database: Higher log marginal likelihood and multiple predictive accuracy, whereas the runtime of the learning process is shortened.

5 Conclusions

The main contribution of this paper is twofold. First, the proposal of a novel automatic scheme to perform unsupervised dimensionality reduction comprised of 1) a simple and efficient measure to assess the relevance of every feature for the learning process and 2) a heuristic to calculate a relevance threshold to automatically distinguish between relevant and irrelevant features. Second, to present a framework where unsupervised learning of CGNs benefits from our proposed scheme in order to obtain models that describe the original databases. This framework proposes performing learning taking into account only the relevant features identified by the automatic dimensionality reduction scheme presented. Then, every irrelevant feature is incorporated into the learned model in order to obtain an explanatory CGN for the original database.

Our experimental results for synthetic and real-world domains have suggested great advantages derived from the use of our automatic dimensionality reduction scheme in unsupervised learning of CGNs: A huge decrease of the runtime of the learning process and an achievement of final models that appear to be as good as and, sometimes, even better than the models obtained using all the features in the learning process. Additionally, the experimental results have proven that the assumption that we made, once relevance was defined according to our purpose to perform

TABLE 1
Comparison of the Performance Achieved when Learning CGNs from the Original Databases and when Our Automatic Dimensionality Reduction Scheme Is Applied

	features		original dimensionality			dimensionality reduction		
database	original	relevant	sc_final	L(test)	runtime	sc_final	L(test)	runtime
synthetic1	25	15	-75240	-15.89	64	-75240	-15.89	50
synthetic2	30	15	-123248	-26.43	94	-123248	-26.43	66
waveform	40	13	-120913	-25.90	245	-121154	-25.93	104
pima	8	7	-9182	-11.89	6	-9096	-11.68	4

dimensionality reduction, works fairly well in the continuous domains considered.

This paper has primarily focused on the gain in efficiency without degrading the explanatory power of the final models derived from the use of the referred scheme as a preprocessing for the learning process. However, it is worth noticing that the identification of the relevant and irrelevant features for the learning process allows us to reach a better comprehensibility and readability of the problem domains and the elicited models.

Few works have addressed the problem of unsupervised feature selection as a preprocessing step [6], [8], [39], [40]. However, all of them differ from our work. Whereas we focus on the description of the original database, [6], [8], [40] are interested in the class label predictive accuracy and [39] in the multiple predictive accuracy. This impossibilities a fair comparison between these different approaches. Moreover, our automatic dimensionality reduction scheme offers a series of advantages over the other existing mechanisms. In addition to its simplicity and efficiency, our scheme is not coupled to any particular learning algorithm and it could be adapted to most of them. On the other hand, the existing unsupervised feature selection mechanisms based on wrapper approaches are tailored to the performance criterion of the particular subsequent learning algorithm (see [8], [39]) and, thus, usually require a great deal of processing time for large databases. Furthermore, [6], [40] propose feature selection mechanisms based on filter approaches that only provide the user with a ranking of the features leaving open the problem of determining how many features should be used to perform a proper learning. Our scheme is able to automatically distinguish between relevant and irrelevant features in the relevance ranking. Then, one line of future research could be the extension of our current contribution to categorical data in order to overcome the problem of determining the number of features to be used by the subsequent learning algorithm.

We are aware that the contribution presented in this paper is unable to deal properly with domains where *redundant* features exist (i.e., features whose values can be exactly determined from the rest of the features). The reason is that the relevance measure introduced in Section 3.2.1 scores each feature separately instead of as groups of features. Thus, redundant features would be considered relevant although they would not provide the learning process with additional information over the true relevant features. To detect these features is necessary because they have an effect on the runtime of the learning process. One of the lines of research that we are currently exploring is concerned with the extension of the general framework depicted in this paper to the case where redundant features exist. Our current work is focused on the derivation of a new relevance measure to assess the gain in relevance of each feature in relation to the features considered relevant so far.

Acknowledgments

J.M. Peña would like to thank Dr. Steve Ellacott for his interest in this work and his useful comments. He also made it possible to visit the School of Computing and Mathematical Sciences at the University of Brighton, Brighton, United Kingdom. The authors would also like to thank the two anonymous reviewers whose useful comments to about previous version of this paper helped us to improve our manuscript. 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.* New York: Academic Press, 1973.

[2] J. Banfield and A. Raftery, "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics,* vol. 49, pp. 803-821, 1993.

[3] A.L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," *Artificial Intelligence,* vol. 97, pp. 245-271, 1997.

[4] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees.* Belmont, Calif.: Wadsworth Int'l Group, 1984.

[5] G. Cooper and E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Machine Learning,* vol. 9, pp. 309-347, 1992.

[6] M. Dash, H. Liu, and J. Yao, "Dimensionality Reduction of Unsupervised Data," *Proc. Ninth IEEE Int'l Conf. Tools with Artificial Intelligence,* pp. 532-539, 1997.

[7] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[8] M. Devaney and A. Ram, "Efficient Feature Selection in Conceptual Clustering," *Proc. 14th Int'l Conf. Machine Learning,* 1997.

[9] J. Doak, "An Evaluation of Feature Selection Methods and Their Application to Computer Security," Technical Report CSE-92-18, Dept. of Computer Science, Univ. of California at Davis, 1992.

[10] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons, 1973.

[11] D. Fisher, "Knowledge Acquisition via Incremental Conceptual Clustering," *Machine Learning,* vol. 2, pp. 139-172, 1987.

[12] D. Fisher and G. Hapanyengwi, "Database Management and Analysis Tools of Machine Induction," *J. Intelligent Information Systems,* vol. 2, pp. 5-38, 1993.

[13] N. Friedman, "The Bayesian Structural EM Algorithm," *Proc. 14th Conf. Uncertainty in Artificial Intelligence,* pp. 129-138, 1998.

[14] N. Friedman and M. Goldszmidt, "Building Classifiers Using Bayesian Networks," *Proc. 13th Nat'l Conf. Artificial Intelligence,* pp. 1277-1284, 1996.

[15] D. Geiger and D. Heckerman, "Learning Gaussian Networks," Technical Report MSR-TR-94-10, Microsoft Research, Redmond, Wash., 1994.

[16] D. Geiger and D. Heckerman, "Learning Gaussian Networks," *Proc. 10th Conf. Uncertainty in Artificial Intelligence* pp. 235-243, 1995.

[17] I. Good, "Rational Decisions," *J. Royal Statistical Soc. B,* vol. 14, pp. 107-114, 1952.

[18] J.A. Hartigan, *Clustering Algorithms.* New York: John Wiley & Sons, 1975.

[19] D. Heckerman and D. Geiger, "Likelihoods and Parameter Priors for Bayesian Networks," Technical Report MSR-TR-95-54, Microsoft Research, Redmond, Wash., 1995.

[20] I. Inza, P. Larrañaga, R. Etxeberria, and B. Sierra, "Feature Subset Selection by Bayesian Networks-Based Optimization," *Artificial Intelligence,* vol. 123, pp. 157-184, 2000.

[21] G.H. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," *Proc. 11th Int'l Conf. Machine Learning,* pp. 121-129, 1994.

[22] L. Kaufman and P. Rousseeuw, *Finding Groups in Data.* New York: John Wiley & Sons, 1990.

[23] R. Kohavi and G.H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* vol. 97, pp. 273-324, 1997.

[24] P. Larrañaga and J.A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation.* Kluwer Academic Publishers, 2001.

[25] S.L. Lauritzen, "Propagation of Probabilities, Means and Variances in Mixed Graphical Association Models," *J. Am. Statistical Assoc.,* vol. 87, pp. 1098-1108, 1992.

[26] S.L. Lauritzen, *Graphical Models.* Oxford, U.K.: Clarendon Press, 1996.

[27] S.L. Lauritzen and N. Wermuth, "Graphical Models for Associations between Variables, Some of which Are Qualitative and Some Quantitative," *The Annals of Statistics,* vol. 17, pp. 31-57, 1989.

[28] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Dordrecht, The Netherlands: Kluwer Academic, 1998.

[29] G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* New York: John Wiley & Sons, 1997.

[30] M. Meila, "Learning with Mixtures of Trees," PhD thesis, Dept. of Electrical Eng. and Computer Science, Massachusetts Inst. of Technology, Cambridge, Mass., 1999.

[31] M. Meila and D. Heckerman, "An Experimental Comparison of Several Clustering and Initialization Methods" *Proc. 14th Conf. Uncertainty in Artificial Intelligence,* pp. 386-395, 1998.

[32] M. Meila and M.I. Jordan, "Estimating Dependency Structure as a Hidden Variable," *Neural Information Processing Systems,* vol. 10, pp. 584-590, 1998.

[33] C. Merz, P. Murphy, and D. Aha, "UCI Repository of Machine Learning Databases," Dept. Information and Computer Science, Univ. of California, Irvine, Calif., 1997. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[34] J.M. Peña, J.A. Lozano, and P. Larrañaga, "An Improved Bayesian Structural EM Algorithm for Learning Bayesian Networks for Clustering," *Pattern Recognition Letters,* vol. 21, pp. 779-786, 2000.

[35] J.M. Peña, J.A. Lozano, and P. Larrañaga, "Learning Recursive Bayesian Multinets for Data Clustering by Means of Constructive Induction," *Machine Learning,* to appear in 2001.

[36] J.M. Peña, J.A. Lozano, and P. Larrañaga, "Performance Evaluation of Compromise Conditional Gaussian Networks for Data Clustering," *Int'l J. Approximate Reasoning,* to appear in 2001.

[37] J.M. Peña, J.A. Lozano, and P. Larrañaga, "Learning Conditional Gaussian Networks for Data Clustering via Edge Exclusion Tests," *Pattern Recognition Letters,* 2000.

[38] P.W.F. Smith and J. Whittaker, "Edge Exclusion Tests for Graphical Gaussian Models," *Learning in Graphical Models,* pp. 555-574, 1998.

[39] L. Talavera, "Feature Selection as a Preprocessing Step for Hierarchical Clustering," *Proc. 16th Int'l Conf. on Machine Learning,* pp. 389-397, 1999.

[40] L. Talavera, "Dependency-Based Feature Selection for Clustering Symbolic Data," *Intelligent Data Analysis,* vol. 4, pp. 19-28, 2000.

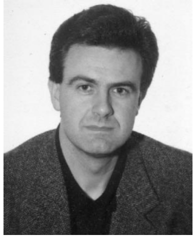
[41] B. Thiesson, C. Meek, D.M. Chickering, and D. Heckerman, "Learning Mixtures of DAG Models," *Proc. 14th Conf. Uncertainty in Artificial Intelligence,* pp. 504-513, 1998.

[42] D. Wettschereck and D. Aha, "Weighting Features," *Proc. First Int'l Conf. Case-Based Reasoning,* 1995.

[43] J. Whittaker, *Graphical Models in Applied Multivariate Statistics.* Chichester, U.K.: John Wiley & Sons, 1990.

Jose Manuel Peña received the computer science degree from the University of the Basque Country, Spain and the BSc degree in computer science from the University of Brighton, United Kingdom. He is currently pursuing the PhD degree in computer science in the Department of Computer Science and Artifical Intelligence at the University of the Basque Country. His research interests include data clustering via probabilistic graphical models and optimization.

Jose Antonio Lozano received the PhD degree in computer science and is an associate professor in the Department of Computer Science and Artifical Intelligence at the University of the Basque Country, Spain. Dr. Lozano's research interests include probabilistic graphical models, evolutionary algorithms, optimization, and machine learning.



Pedro Larrañaga received the PhD degree in computer science and is an associate professor in the Department of Computer Science and Artifical Intelligence at the University of the Basque Country, Spain where he leads the Intelligence Systems Group. His research interests include probabilistic graphical models, evolutionary algorithms, optimization, and machine learning.

Iñaki Inza is a lecturer at the University of the Basque Country, Spain, where he is currently pursuing the PhD degree in computer science. His research interests include machine learning, evolutionary algorithms, and Bayesian networks.