UNSUPERVISED LEARNING OF BAYESIAN NETWORKS VIA ESTIMATION OF DISTRIBUTION ALGORITHMS: AN APPLICATION TO GENE EXPRESSION DATA CLUSTERING

J. M. PEÑA

Department of Computer Science, Aalborg University
Fredrik Bajers Vej 7E, DK-9220 Aalborg, Denmark
jmp@cs.auc.dk

J. A. LOZANO

Department of Computer Science and Artificial Intelligence, University of the Basque Country
Paseo Manuel de Lardizábal 1, E-20018 Donostia-San Sebastián, Spain
ccploalj@si.ehu.es

P. LARRAÑAGA

Department of Computer Science and Artificial Intelligence, University of the Basque Country
Paseo Manuel de Lardizábal 1, E-20018 Donostia-San Sebastián, Spain
ccplamup@si.ehu.es

This paper proposes using estimation of distribution algorithms for unsupervised learning of Bayesian networks, directly as well as within the framework of the Bayesian structural EM algorithm. Both approaches are empirically evaluated in synthetic and real data. Specifically, the evaluation in real data consists in the application of this paper's proposals to gene expression data clustering, i.e., the identification of clusters of genes with similar expression profiles across samples, for the leukemia database. The validation of the clusters of genes that are identified suggests that these may be biologically meaningful.

Keywords: Unsupervised learning; Bayesian networks; estimation of distribution algorithms; gene expression data analysis.

1. Introduction

One of the main problems that arises in a great variety of fields, including artificial intelligence, machine learning, and statistics, is the so-called *data clustering problem.* Given some data in the form of a set of instances with an underlying group-structure, data clustering may be roughly defined as the search for the best description of the underlying group-structure according to a certain criterion, when the true group membership of every instance is unknown. Each of the groups that exist in the data at hand is called a *cluster.*

Among the different interpretations and expectations that the term data clustering gives rise to, this paper is limited to data clustering problems that are basically defined by the following assumptions:

- A database $\boldsymbol{d}$ containing N instances or cases, i.e., $\boldsymbol{d} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, is available. The l-th case of $\boldsymbol{d}$ is represented by an $(n+1)$-dimensional discrete vector $\boldsymbol{x}_l = (x_{l1}, \ldots, x_{ln+1})$ that is partitioned as $\boldsymbol{x}_l = (c_l, \boldsymbol{y}_l)$ for all l. c_l is the unknown cluster membership of $\boldsymbol{x}_l$, and $\boldsymbol{y}_l = (y_{l1}, \ldots, y_{ln})$ is the n-dimensional discrete vector of observations or *predictive attributes* of $\boldsymbol{x}_l$ for all l.
- The number of clusters in the underlying group-structure of $\boldsymbol{d}$, in the forthcoming referred to as K, is known.
- Each of the K clusters underlying $\boldsymbol{d}$ corresponds to a physical process that is defined by an unknown joint probability distribution. Then, every case in $\boldsymbol{d}$ may be seen as sampled from exactly one of these K unknown joint probability distributions. This corresponds to assuming the existence of an $(n+1)$-dimensional discrete random variable $\boldsymbol{X} = (X_1, \ldots, X_{n+1})$ that is partitioned as $\boldsymbol{X} = (C, \boldsymbol{Y})$. C is a unidimensional discrete hidden random variable that represents the unknown cluster membership, i.e., the *cluster random variable.* $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ is an n-dimensional discrete random variable that represents the set of predictive attributes, i.e., the *predictive random variable.* Moreover, it is usual to assume that the mechanism that generated $\boldsymbol{d}$ works in two stages: First, one of the physical processes that are associated with the K clusters that exist in $\boldsymbol{d}$ is somehow selected according to a probability distribution for C and, then, an instance is somehow generated according to the joint probability distribution for $\boldsymbol{Y}$ that defines the selected physical process. The existence of a random variable C whose entries in $\boldsymbol{d}$ are unknown makes data clustering be also referred to as *learning from unlabelled data* or, simply, as *unsupervised learning.*
- The parametric forms of the joint probability distributions that govern the mechanism that generated $\boldsymbol{d}$ are all known to be multinomial.

Under these assumptions, data clustering is usually approached from the *probabilistic* or *model-based* perspective: The description of the K clusters underlying $\boldsymbol{d}$ is accomplished through the probabilistic modelling of the mechanism that generated $\boldsymbol{d}$. Consequently, probabilistic data clustering reduces to learning a joint probability distribution for $\boldsymbol{X}$ from $\boldsymbol{d}$. When the aim is to represent a joint probability distribution in general and for probabilistic data clustering in particular, one of the paradigms that can be helpful is the Bayesian network paradigm [1,2,3]. This paper is concerned with unsupervised learning of Bayesian networks [4,5,6] as a means to solve probabilistic data clustering problems.

Specifically, this paper proposes the use of a relatively novel family of evolutionary algorithms, called estimation of distribution algorithms [7,8,9], for unsupervised learning of Bayesian networks, both directly and within the Bayesian structural EM algorithm framework [10]. These two approaches are empirically evaluated in synthetic as well as in real data. The results that are reported for synthetic data

confirm the ability of this paper's proposals to induce models that perform satisfactorily when compared to the original models. The experimental evaluation in real data consists in the application of this paper's proposals to gene expression data clustering, i.e., the identification of clusters of genes with similar expression profiles across samples, for the leukemia database [11]. The clusters of genes that are obtained are described and validated. The validation shows that the clusters are homogeneous and have a natural interpretation. This suggests that they may be biologically meaningful.

The remainder of this paper is structured as follows. Section 2 reviews unsupervised learning of Bayesian networks. Section 3 introduces estimation of distribution algorithms. Section 4 compiles and discusses the experimental results for synthetic as well as for real data. Finally, Section 5 closes with some conclusions.

2. Bayesian Networks for Data Clustering

Let $\boldsymbol{X}$ be a random variable as stated above, i.e., an $(n+1)$-dimensional discrete random variable $\boldsymbol{X} = (X_1, \ldots, X_{n+1})$ that is partitioned as $\boldsymbol{X} = (C, \boldsymbol{Y})$ into a unidimensional discrete hidden cluster random variable C and an n-dimensional discrete predictive random variable $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$. A *Bayesian network (BN)* for (probabilistic) data clustering for $\boldsymbol{X}$ [4,5,6] consists of (i) a directed acyclic graph (DAG) whose nodes correspond to the unidimensional random variables of $\boldsymbol{X}$, i.e., the *model structure*, and (ii) a set of local probability distributions, one for each node of the model structure conditioned on each state of its parents [a]. The model structure encodes a set of conditional (in)dependencies between the random variables of $\boldsymbol{X}$, and it is usually constrained so that every Y_i is a child of C. This restriction is imposed by the assumption about how the generative mechanism underlying the domain works (recall Section 1). A BN for data clustering for $\boldsymbol{X}$ represents a graphical factorization of a joint probability distribution for $\boldsymbol{X}$ as follows:

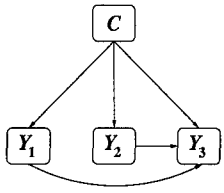
$$\begin{aligned} p(\boldsymbol{x} \mid \boldsymbol{\theta_s}, s^h) &= p(c \mid \boldsymbol{\theta_s}, s^h) p(\boldsymbol{y} \mid c, \boldsymbol{\theta_s}, s^h) \\ &= p(c \mid \boldsymbol{\theta}_C, s^h) \prod_{i=1}^{n} p(y_i \mid c, \boldsymbol{pa}(s^{\boldsymbol{Y}})_i, \boldsymbol{\theta}_i, s^h) \end{aligned} \tag{1}$$

where s is the model structure and $\boldsymbol{pa}(s^{\boldsymbol{Y}})_i$, with $s^{\boldsymbol{Y}}$ the subgraph of s that is induced by $\boldsymbol{Y}$, denotes the state of those parents of Y_i that correspond to predictive random variables, $\boldsymbol{Pa}(s^{\boldsymbol{Y}})_i$, for all i. The local probability distributions of the BN for data clustering for $\boldsymbol{X}$ are those induced by the terms in Eq. (1), and they are univariate multinomial distributions that depend on a finite set of parameters $\boldsymbol{\theta_s} = (\boldsymbol{\theta}_C, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$, i.e., the *model parameters*. Moreover, s^h denotes the hypothesis that the true joint probability distribution for $\boldsymbol{X}$ can be graphically factorized according to the conditional independencies in s.

As K clusters exist, C can take K distinct values that are denoted by $c^1, \ldots, c^K$. Then, the univariate multinomial distribution $p(c \mid \boldsymbol{\theta}_C, s^h)$ consists of a set of

[a] Throughout the text, the terms node and random variable are interchangeably used.

Table 1. Structure (left), parameters (middle), and local probability distributions (right) of a BN for data clustering for $\boldsymbol{X} = (C, \boldsymbol{Y}) = (C, Y_1, Y_2, Y_3)$, where C and every Y_i are binary.



$\boldsymbol{\theta}_C = (\theta^1, \theta^2)$	$p(c \mid \boldsymbol{\theta}_C, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_1 = (\boldsymbol{\theta}_1^1, \boldsymbol{\theta}_1^2)$	
$\boldsymbol{\theta}_1^1 = (\theta_1^{1\text{-}1}, \theta_1^{1\text{-}2})$	$p(y_1 \mid c^1, \boldsymbol{\theta}_1, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_1^2 = (\theta_1^{2\text{-}1}, \theta_1^{2\text{-}2})$	$p(y_1 \mid c^2, \boldsymbol{\theta}_1, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_2 = (\boldsymbol{\theta}_2^1, \boldsymbol{\theta}_2^2)$	
$\boldsymbol{\theta}_2^1 = (\theta_2^{1\text{-}1}, \theta_2^{1\text{-}2})$	$p(y_2 \mid c^1, \boldsymbol{\theta}_2, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_2^2 = (\theta_2^{2\text{-}1}, \theta_2^{2\text{-}2})$	$p(y_2 \mid c^2, \boldsymbol{\theta}_2, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3 = (\boldsymbol{\theta}_3^1, \boldsymbol{\theta}_3^2)$	
$\boldsymbol{\theta}_3^1 = (\boldsymbol{\theta}_3^{11}, \boldsymbol{\theta}_3^{12}, \boldsymbol{\theta}_3^{13}, \boldsymbol{\theta}_3^{14})$	
$\boldsymbol{\theta}_3^{11} = (\theta_3^{111}, \theta_3^{112})$	$p(y_3 \mid c^1, y_1^1, y_2^1, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3^{12} = (\theta_3^{121}, \theta_3^{122})$	$p(y_3 \mid c^1, y_1^1, y_2^2, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3^{13} = (\theta_3^{131}, \theta_3^{132})$	$p(y_3 \mid c^1, y_1^2, y_2^1, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3^{14} = (\theta_3^{141}, \theta_3^{142})$	$p(y_3 \mid c^1, y_1^2, y_2^2, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3^2 = (\boldsymbol{\theta}_3^{21}, \boldsymbol{\theta}_3^{22}, \boldsymbol{\theta}_3^{23}, \boldsymbol{\theta}_3^{24})$	
$\boldsymbol{\theta}_3^{21} = (\theta_3^{211}, \theta_3^{212})$	$p(y_3 \mid c^2, y_1^1, y_2^1, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3^{22} = (\theta_3^{221}, \theta_3^{222})$	$p(y_3 \mid c^2, y_1^1, y_2^2, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3^{23} = (\theta_3^{231}, \theta_3^{232})$	$p(y_3 \mid c^2, y_1^2, y_2^1, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$
$\boldsymbol{\theta}_3^{24} = (\theta_3^{241}, \theta_3^{242})$	$p(y_3 \mid c^2, y_1^2, y_2^2, \boldsymbol{\theta}_3, \boldsymbol{s}^h)$

probabilities of the form

$$p(c^g \mid \boldsymbol{\theta}_C, \boldsymbol{s}^h) = \theta_{c^g} = \theta^g > 0 \tag{2}$$

representing the probability that C takes its g-th state for all g. Furthermore, $\sum_{g=1}^{K} \theta^g = 1$. Consequently, the parameters of the local probability distribution for C are given by $\boldsymbol{\theta}_C = (\theta^1, \ldots, \theta^K)$. Besides, let $y_i^1, \ldots, y_i^{r_i}$ denote the r_i distinct values that Y_i can take, and let $\boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^1, \ldots, \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^{q_i}$ denote the q_i distinct states that $\boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i$ can have, with $q_i = \prod_{Y_e \in \boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i} r_e$ for all i. Then, the univariate multinomial distribution $p(y_i \mid c^g, \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^j, \boldsymbol{\theta}_i, \boldsymbol{s}^h)$ for all g, i, and j consists of a set of probabilities of the form

$$p(y_i^k \mid c^g, \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^j, \boldsymbol{\theta}_i, \boldsymbol{s}^h) = \theta^g_{y_i^k \mid \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^j} = \theta_i^{gjk} > 0 \tag{3}$$

representing the conditional probability that Y_i takes its k-th state given that C takes its g-th value and $\boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i$ takes its j-th value for all k. Furthermore, $\sum_{k=1}^{r_i} \theta_i^{gjk} = 1$ for all g, i, and j. Consequently, the parameters of the local probability distributions for every Y_i are given by $\boldsymbol{\theta}_i = (\boldsymbol{\theta}_i^g)_{g=1}^K$ with $\boldsymbol{\theta}_i^g = (\boldsymbol{\theta}_i^{gj})_{j=1}^{q_i}$ and $\boldsymbol{\theta}_i^{gj} = (\theta_i^{gjk})_{k=1}^{r_i}$ for all g and j. Table 1 shows a BN for data clustering.

As mentioned in Section 1, a BN for data clustering for $\boldsymbol{X}$ induced from an unlabelled database $\boldsymbol{d}$ encodes a description of the K clusters that exist in $\boldsymbol{d}$, by modelling the joint probability distribution for $\boldsymbol{X}$ that defines the mechanism that generated $\boldsymbol{d}$. Moreover, this is an effective and efficient description, because a BN for data clustering for $\boldsymbol{X}$ explicitly reflects the existing conditional (in)dependencies between the random variables of $\boldsymbol{X}$ (model structure) as well as the strictly necessary

parameters to be estimated (model parameters). As seen in Eq. (1), the description of the K clusters underlying $\boldsymbol{d}$ actually consists of (i) $p(c \mid \boldsymbol{\theta}_C, \boldsymbol{s}^h)$ modelling how one of the physical processes that are associated with the clusters was selected by the mechanism that generated $\boldsymbol{d}$, and (ii) $p(\boldsymbol{y} \mid c^g, \boldsymbol{\theta}_{\boldsymbol{S}}, \boldsymbol{s}^h)$ for all g modelling how the generative mechanism caused every instance that is summarized in $\boldsymbol{d}$, after one cluster was selected. Note that $p(\boldsymbol{y} \mid c^g, \boldsymbol{\theta}_{\boldsymbol{S}}, \boldsymbol{s}^h)$ for all g graphically factorize further according to the conditional independencies that are encoded in $\boldsymbol{s}$ (recall Eq. (1)). Once a BN for data clustering for $\boldsymbol{X}$ has been induced from $\boldsymbol{d}$, it constitutes an effective device for reasoning under uncertainty. However, learning such a model is challenging in general. As a matter of fact, it has been proven [12] that the identification of the BN structure with the highest Bayesian Dirichlet equivalent score [13] among all the BN structures in which every node has no more than t parents is an NP-hard optimization problem for $t > 1$. It is usually assumed that this hardness holds for other common scores as well, though there is not yet a formal proof [14]. These results also apply to unsupervised learning of BNs. This paper interprets unsupervised learning of BNs as an optimization problem where the search space, the objective function, and the search strategy are as follows.

As search space, this paper considers the space of structures of BNs for data clustering. This space can be restricted to the space of DAGs for $\boldsymbol{Y}$, due to the fact that every Y_i is a child of C. Alternative search spaces include the space of equivalence classes of structures of BNs for data clustering and the space of ancestral orderings of structures of BNs for data clustering. Note that, as usually, model parameter fitting is considered a secondary optimization problem: Given a BN structure for data clustering, maximum likelihood (ML) or maximum a posteriori model parameter estimates can be effectively obtained via approximation techniques such as the EM algorithm, Gibbs sampling, or gradient descent methods.

As objective function, this paper considers the *Bayesian information criterion (BIC)* [15], which can be expressed as follows:

$$Sc(\boldsymbol{s}, \boldsymbol{d}) = \log L(\boldsymbol{d} \mid \widehat{\boldsymbol{\theta}}_{\boldsymbol{S}}, \boldsymbol{s}^h) - \frac{1}{2} \log N dim(\boldsymbol{s}) \tag{4}$$

where $\boldsymbol{s}$ is the model structure being evaluated, $\widehat{\boldsymbol{\theta}}_{\boldsymbol{S}}$ are the ML model parameter estimates for $\boldsymbol{s}$, and $dim(\boldsymbol{s}) = (K-1) + \sum_{i=1}^{n}[(r_i - 1)K \prod_{Y_e \in \boldsymbol{Pa}(\boldsymbol{s}\boldsymbol{Y}_{)_i}} r_e]$ is the dimension of $\boldsymbol{s}$. Then, the goal of the problem optimization process is maximization. Other scores that can serve as objective function include Bayesian scores and information theory scores.

As search strategy, this paper proposes and evaluates a relatively novel family of evolutionary algorithms, known as estimation of distribution algorithms [7,8,9]. In addition to the direct application of estimation of distribution algorithms to unsupervised learning of BNs, this paper also studies the benefits of fitting them into the framework of the *Bayesian structural EM (BSEM) algorithm* [10] as follows. Table 2 shows a pseudocode of the generic BSEM algorithm for unsupervised learning of BNs. As can be seen, the generic BSEM algorithm iterates between two main steps.

Table 2. Pseudocode of the generic BSEM algorithm for unsupervised learning of BNs.

1.	Let s_1 be the initial model structure
2.	**for** $u = 1, 2, \ldots$ **do**
3.	Run the EM algorithm in order to approximate the ML parameters $\widehat{\theta}_{s_u}$ for s_u
4.	Perform a search over model structures, evaluating each one by $Sc(s : s_u, d) = E[Sc(s, d) \mid d^{Y}, \widehat{\theta}_{s_u}, s_u^h] = \sum_{d^C} Sc(s, (d^C, d^{Y}))L(d^C \mid d^{Y}, \widehat{\theta}_{s_u}, s_u^h)$
5.	Let s_{u+1} be the model structure with the highest score among those visited in step 4
6.	**if** $Sc(s_{u+1} : s_u, d) = Sc(s_u : s_u, d)$ **then**
7.	Return $(s_u, \widehat{\theta}_{s_u})$

The first step (step 3 in Table 2) approximates the ML parameters for the current model structure given the observed data, usually via the EM algorithm [16,17]. On the other hand, the second step (step 4 in Table 2) performs a search for the model structure with the highest expected score with respect to the observed data and the best model found so far. That is, the score that guides the structural search at the u-th iteration of the generic BSEM algorithm is as follows for all u:

$$\begin{aligned} Sc(s : s_u, d) &= E[Sc(s, d) \mid d^{Y}, \widehat{\theta}_{s_u}, s_u^h] \\ &= \sum_{d^C} Sc(s, (d^C, d^{Y}))L(d^C \mid d^{Y}, \widehat{\theta}_{s_u}, s_u^h) \end{aligned} \tag{5}$$

where d^C and d^{Y} denote d restricted to the missing entries for the cluster random variable C and to the values for the predictive random variable Y, respectively. In this paper, $Sc(s, d)$ in Eq. (5) corresponds to the BIC (recall Eq. (4)). In principle, any search strategy can be used to solve the structural search step at each iteration of the generic BSEM algorithm, being greedy hill-climbing the most common choice. This paper proposes applying estimation of distribution algorithms. The next section provides the reader with an introduction to them.

3. Estimation of Distribution Algorithms

Among stochastic heuristic search strategies for problem optimization, *evolutionary algorithms (EAs)* [18,19] are well known for their good performance and wide applicability. Classical examples of EAs are genetic algorithms, evolutionary programming, and evolution strategies. The main feature that is shared by all the instances of the EA paradigm is the fact of being inspired by Darwinian natural evolution. That is why much of the nomenclature of EAs is borrowed from this field. For instance, one talks about *populations* to refer to sets of solutions to an optimization problem, each solution is called an *individual*, and each basic component of an individual is named a *gene*. The main components of most EA instances are: An initial population of individuals, a *selection method*, a set of *random operators*, and a *replacement method*. Basically, all the EAs work in the same iterative way: At each iteration or *generation* some individuals of the current population are selected according to the selection method and modified by the random operators in order to create new individuals and, consequently, a new population via the replacement method. The

Table 3. Pseudocode of the generic EDA.

1.	Let $\boldsymbol{po}_1$ be a population composed of Q randomly generated individuals
2.	Evaluate the individuals in $\boldsymbol{po}_1$
3.	$u = 1$
4.	**while** the stopping criterion is not met **do**
5.	Let $\boldsymbol{d}_u$ group M individuals selected from $\boldsymbol{po}_u$ via the selection method
6.	Let $p_u(\boldsymbol{z})$ be the joint probability distribution for $\boldsymbol{Z}$ learnt from $\boldsymbol{d}_u$
7.	Let $\boldsymbol{of}_u$ be the offspring population composed of R individuals sampled from $p_u(\boldsymbol{z})$
8.	Evaluate the individuals in $\boldsymbol{of}_u$
9.	Let $\boldsymbol{po}_{u+1}$ be the population created from $\boldsymbol{po}_u$ and $\boldsymbol{of}_u$ via the replacement method
10.	$u++$
11.	Return the best individual found so far

objective of this iterative process is to evolve the population of individuals towards promising zones of the search space of the optimization problem at hand.

Recently, a novel class of EAs, known as *estimation of distribution algorithms (EDAs)* [7,8,9], has been proposed. The main characteristic of EDAs is the non-existence of random operators. Instead, the offspring population is generated from the current one at each iteration by learning and subsequent simulation of a joint probability distribution for a database conformed with those individuals that are selected from the current population by means of the selection method. This results in two important advantages of EDAs over classical EAs: The sometimes necessary design of random operators tailored to the particular optimization problem at hand is avoided, and the number of parameters to be assessed by the user is reduced. A further advantage of EDAs over classical EAs is that the relationships between the genes of the individuals that are selected at each generation can be exploited through the joint probability distribution that is learnt from those individuals. Besides, EDAs keep the main strengths of classical EAs: Wide applicability and good performance [7].

As any other class of EAs, EDAs are based on detecting promising zones of the search space of the optimization problem at hand by evolving a population of individuals. For this purpose, the generic EDA iterates between three main steps, after the individuals of the initial population $\boldsymbol{po}_1$ have been generated, usually at random, and evaluated. The iterative process ends when the stopping criterion is met, e.g., performance of a maximum number of generations, uniformity in the current population, or no improvement with regard to the best individual of the previous generation. This causes the best solution found so far being returned. The three main steps of the u-th iteration of the generic EDA are as follows for all u. First, M of the Q individuals of the current population $\boldsymbol{po}_u$ are selected by means of the selection method. Then, these individuals are used to construct a learning database, denoted by $\boldsymbol{d}_u$, from which a joint probability distribution for $\boldsymbol{Z}$, $p_u(\boldsymbol{z})$, is induced. $\boldsymbol{Z} = (Z_1, \ldots, Z_m)$ denotes an m-dimensional discrete random variable, where each Z_i is associated with one of the m genes of every individual in $\boldsymbol{d}_u$. Finally, R individuals are sampled from $p_u(\boldsymbol{z})$ and evaluated in order to create the

offspring population $\boldsymbol{of}_u$ which, then, is used to generate the new population $\boldsymbol{po}_{u+1}$ by replacing some individuals of $\boldsymbol{po}_u$ via the replacement method. A schematic of the generic EDA is shown in Table 3.

Learning $p_u(\boldsymbol{z})$ from $\boldsymbol{d}_u$ constitutes the main bottleneck of the u-th iteration of the generic EDA for all u. Obviously, the computation of all the parameters that are needed to specify this joint probability distribution in the standard representation is often impractical. For this reason, several families of EDAs have arisen where $p_u(\boldsymbol{z})$ is assumed to factorize according to a certain class of probabilistic models for all u [7,9]. For the purpose of this paper, it suffices to consider one of the simplest instances of the generic EDA, namely the *univariate marginal distribution algorithm (UMDA)* [7,9,20], which is based on the assumption that $p_u(\boldsymbol{z})$ factorizes as

$$p_u(\boldsymbol{z}) = \prod_{i=1}^{m} p_u(z_i) \tag{6}$$

and $p_u(z_i)$ is restricted to be a univariate multinomial distribution whose parameters are estimated from $\boldsymbol{d}_u$ according to the ML criterion for all i and u. Obviously, the assumption behind Eq. (6) may not hold in practice, as relationships between the unidimensional random variables of $\boldsymbol{Z}$ may exist in the optimization problem at hand. However, this assumption simplifies learning the probabilistic model for the factorization of $p_u(\boldsymbol{z})$ from $\boldsymbol{d}_u$ for all u, because this process reduces to parameter fitting. Furthermore, the UMDA has proven to work successfully in many domains and has received much attention in the literature [7,20].

4. Empirical Evaluation

This section is devoted to empirically evaluate the performance of EDAs for unsupervised learning of BNs, both directly and within the generic BSEM algorithm (see Section 2), in synthetic as well as in real data (gene expression data). The evaluation is limited to the UMDA, as this is one of the simplest but most widely used and studied instances of the generic EDA (recall Section 3). In the forthcoming, the direct application of the UMDA to unsupervised learning of BNs is simply referred to as the UMDA, and the incorporation of the UMDA into the generic BSEM algorithm for unsupervised learning of BNs is denoted as the BSEM-UMDA. Recall from Section 2 that both the UMDA and the BSEM-UMDA search for the best BN for data clustering in the space of DAGs for $\boldsymbol{Y}$. However, whereas the BIC (see Eq. (4)) is the objective function in the case of the UMDA, the expected BIC with respect to the observed data and the best model found so far (see Eq. (5)) is the score that guides the structural search at each iteration of the BSEM-UMDA.

This section starts by describing the experimental setup. Then, the performance of the UMDA and the BSEM-UMDA in synthetic data is discussed. Finally, the UMDA and the BSEM-UMDA are validated in a real-world domain: Gene expression data clustering for the leukemia database [11].

4.1. *Evaluation setup*

The representation that is considered in the UMDA and the BSEM-UMDA for every solution $\boldsymbol{s}^{\boldsymbol{Y}}$ in the search space uses an $n \times n$ adjacency matrix $\boldsymbol{a} = (a_{ij})$, such that (i) $a_{ij} = 2$ if $Y_j \in \boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i$, (ii) $a_{ij} = 1$ if $Y_i \in \boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_j$, and (iii) $a_{ij} = 0$ otherwise for all i and j. Therefore, every solution in the search space can be represented by an m-dimensional individual $\boldsymbol{z} = (z_1, \ldots, z_m)$, where $m = (n^2 - n)/2$, consisting only of the elements of $\boldsymbol{a}$ either above or below the diagonal.

It should be noted that the creation of $\boldsymbol{po}_1$ and $\boldsymbol{of}_u$ for all u is not closed with respect to the DAG property. Thus, individuals representing invalid solutions may appear during the problem optimization process. Invalid solutions need to be repaired before they are evaluated. A simple randomized repair operator is used in the UMDA and the BSEM-UMDA: An invalid solution is repaired by, iteratively, removing a randomly chosen directed edge that invalidates the DAG property until a DAG is obtained. Note that the repair operator does not modify the individuals but the invalid solutions that are represented by them.

The selection and the replacement methods of the UMDA and the BSEM-UMDA are as follows. The most fitted individuals in $\boldsymbol{po}_u$ are selected to conform $\boldsymbol{d}_u$ for all u. On the other hand, $\boldsymbol{po}_{u+1}$ is obtained as the result of replacing the least fitted individuals in $\boldsymbol{po}_u$ by $\boldsymbol{of}_u$ for all u. Moreover, the size of the population, Q, the number of selected individuals, M, and the size of the offspring population, R, are set to 75, 25, and 50, respectively, for the UMDA. For the BSEM-UMDA, $Q = 7500$, $M = 2500$, and $R = 5000$. The UMDA halts after 50 generations. The UMDA that is run at each iteration of the BSEM-UMDA stops after 50 generations as well. Preliminary experiments confirmed that these parameters are well suited and that they do not favor any of the two techniques over the other.

For ML model parameter estimation when computing the BIC, a multiple-restart version of the EM algorithm is employed. The convergence criterion for the EM algorithm is satisfied when the relative difference between successive values for $\log L(\boldsymbol{d} \mid \boldsymbol{\theta}_{\boldsymbol{S}}, \boldsymbol{s}^h)$ is less than 10^{-6}.

For comparison purposes, the most common instance of the generic BSEM algorithm is used as benchmark. This, referred to as the BSEM-HC in the forthcoming, reduces the structural search step at each iteration of the generic BSEM algorithm to a greedy hill-climbing search that, having the naive Bayes model as initial model, considers all the possible additions, removals, and non-covered reversals of a single directed edge at each point in the search. The score that guides the structural search steps of the BSEM-HC is the same as in the BSEM-UMDA, i.e., Eq. (5).

The performance of the UMDA, the BSEM-UMDA, and the BSEM-HC is assessed according to their capacity for obtaining BNs for data clustering that show satisfactory (i) ability to summarize the learning data, (ii) ability to generalize the learning data to previously unseen data, and (iii) structural similarity to the true model underlying the learning data. The BIC values that are scored by the induced models serve for assessing the first ability. The second ability can be measured by

calculating the log likelihood of some hold-out data given the corresponding elicited models. Finally, the third ability can be assessed as follows [21]. First, the completed partially directed acyclic graphs (CPDAGs) representing the equivalence classes of the structures of each learnt model and the corresponding original model are generated and, then, the number of edges that are different in these two graphs is reported. In the gene expression database, CPDAG distances cannot be computed. Instead, the interpretability and the homogeneity of the gene clusterings that the induced BNs for data clustering represent are extensively validated, as part of the evaluation of the UMDA, the BSEM-UMDA, and the BSEM-HC.

4.2. *Results: Synthetic data*

The first part of the evaluation of the UMDA and the BSEM-UMDA is carried out in three synthetic databases that were obtained by sampling three BNs for data clustering of increasing complexity. The three original models involved a binary cluster random variable C and a 9-dimensional predictive random variable $\boldsymbol{Y} = (Y_1, \ldots, Y_9)$, with Y_i binary for all i. The number of directed edges between unidimensional predictive random variables in each of these models was 10, 15, and 20. These directed edges were uniformly generated, as far as no directed cycle was created. Note that each of the three original models had nine additional directed edges, due to the fact that every Y_i was a child of C. The parameters for each of the original models were generated at random. Finally, 5000 cases were sampled from each of the three original models. Each case consisted only of a state for $\boldsymbol{Y}$, i.e., all the entries for C in the samples were missing. In the forthcoming, the samples are referred to as $\boldsymbol{d}_{10}$, $\boldsymbol{d}_{15}$, and $\boldsymbol{d}_{20}$, where the subscript indicates the number of directed edges between unidimensional predictive random variables in the generative model. In the experiments below, the first 4000 cases of each sample are used as learning data, and the last 1000 cases are set aside and used as testing data. Moreover, the number of clusters is fixed to the true number, i.e., $K = 2$.

Table 4 summarizes the performance of the BNs for data clustering that are induced by the UMDA, the BSEM-UMDA, and the BSEM-HC from $\boldsymbol{d}_{10}$, $\boldsymbol{d}_{15}$, and $\boldsymbol{d}_{20}$. All the performance criteria values in the table are given in terms of averages and standard deviations over five independent runs for the UMDA, and over ten independent runs for the BSEM-UMDA and the BSEM-HC. The performance criteria values of the original models are also given for comparison purposes.

The first conclusion that can be made from Table 4 is that the UMDA and the BSEM-UMDA behave satisfactorily in terms of all the performance measures that are considered in the evaluation, no matter the complexity of the learning database. Moreover, the UMDA and the BSEM-UMDA clearly outperform the BSEM-HC for the three criteria and the three databases. The results in the table also show the superiority of the UMDA over the BSEM-UMDA. As a matter of fact, the UMDA is able to identify models that score BIC values for the learning databases and log likelihood values for the hold-out databases that are very close to those of the

Table 4. Performance of the BNs for data clustering induced by the UMDA, the BSEM-UMDA, and the BSEM-HC from $\boldsymbol{d}_{10}$, $\boldsymbol{d}_{15}$, and $\boldsymbol{d}_{20}$.

		BIC		Log likelihood		CPDAG distance	
		Initial	Final	Initial	Final	Initial	Final
$\boldsymbol{d}_{10}$	Original	—	-8709	—	-2156	—	—
	BSEM-HC	-10372±0	-8732±26	-2203±6	-2159±3	28±0	3±3
	BSEM-UMDA	-10372±0	-8726±18	-2204±6	-2158±1	28±0	3±3
	UMDA	-8935±37	-8714±5	-2191±19	-2158±2	17±3	2±2
$\boldsymbol{d}_{15}$	Original	—	-8898	—	-2189	—	—
	BSEM-HC	-10502±0	-8971±66	-2250±7	-2197±9	32±0	9±6
	BSEM-UMDA	-10502±0	-8930±59	-2249±5	-2195±10	32±0	6±5
	UMDA	-9202±28	-8913±35	-2249±7	-2195±11	22±2	4±2
$\boldsymbol{d}_{20}$	Original	—	-9094	—	-2232	—	—
	BSEM-HC	-10658±0	-9145±45	-2298±8	-2249±11	31±0	10±5
	BSEM-UMDA	-10658±0	-9127±11	-2294±8	-2248±4	31±0	9±1
	UMDA	-9368±37	-9107±19	-2302±11	-2241±4	21±3	9±3

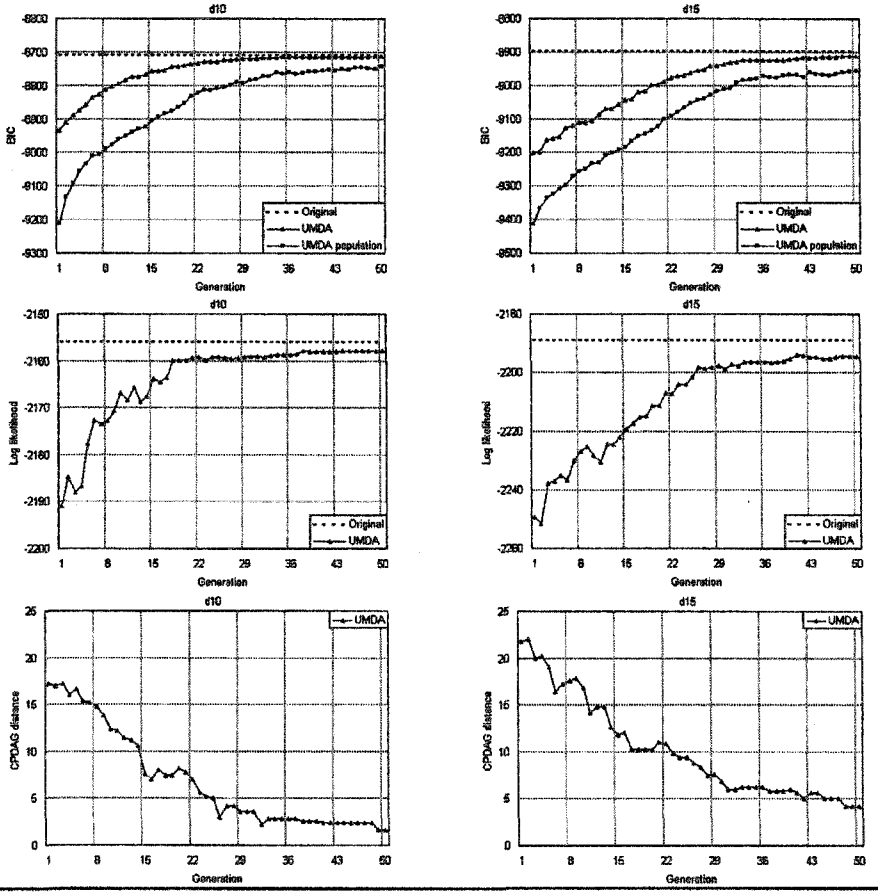
original models.

The fact that the models that are selected by both the UMDA and the BSEM-UMDA enjoy satisfactory log likelihood values for the hold-out databases and CPDAG distances to the generative models confirms that the (expected) BIC is an appropriate objective function to guide the search towards models that, in addition to summarize well the learning data, generalize well to previously unseen data, and encode conditional (in)dependence models fairly similar to those of the original models. As expected, the performance of the induced models with respect to these two criteria slightly degrades as the complexity of the generative model increases and the amount of learning data available remains the same (4000 cases).

The reliability of the UMDA and the BSEM-UMDA to recover the structures of the true models underlying $\boldsymbol{d}_{10}$, $\boldsymbol{d}_{15}$, and $\boldsymbol{d}_{20}$ can be appreciated as follows. Table 4 summarizes the average number of relationships, i.e., non-edges, undirected edges, and directed edges with any orientation, that are different in the CPDAGs corresponding to the equivalence classes of the original and the induced models, out of the 36 pairwise combinations of unidimensional predictive random variables. Then, the number of relationships that coincide in the CPDAGs corresponding to the equivalence classes of the original model and the ones that are learnt by the UMDA is, on average, 34 (94 %) for $\boldsymbol{d}_{10}$, 32 (89 %) for $\boldsymbol{d}_{15}$, and 27 (75 %) for $\boldsymbol{d}_{20}$. The models that are induced by the BSEM-UMDA score, on average, 33 (92 %) for $\boldsymbol{d}_{10}$, 30 (83 %) for $\boldsymbol{d}_{15}$, and 27 (75 %) for $\boldsymbol{d}_{20}$. As discussed above, the models that are selected by the BSEM-HC have larger CPDAG distances than those of the models that are obtained by means of the UMDA and the BSEM-UMDA: They score, on average, 33 (92 %) for $\boldsymbol{d}_{10}$, 27 (75 %) for $\boldsymbol{d}_{15}$, and 26 (72 %) for $\boldsymbol{d}_{20}$.

The graphs in Table 5 complement Table 4 with the dynamics of the UMDA for $\boldsymbol{d}_{10}$ (left) and $\boldsymbol{d}_{15}$ (right). All the performance criteria values in the graphs are

Table 5. Performance of the BNs for data clustering induced by the UMDA from $\boldsymbol{d}_{10}$ (left) and $\boldsymbol{d}_{15}$ (right), as a function of the number of generations.



averaged over five independent runs. The performance criteria values of the original models are also given for comparison purposes. Due to space restrictions, the graphs corresponding to the dynamics of the UMDA for $\boldsymbol{d}_{20}$ are not shown. These curves have essentially the same shapes as those in Table 5 and, thus, they do not provide additional information [22].

It can be clearly appreciated in the graphs in the first row of Table 5 that, as the number of generations of the UMDA increases, the curves corresponding to the BIC values of the population averages (indicated as UMDA population in Table 5) get closer to the curves corresponding to the BIC values of the best models found so far (indicated as UMDA in Table 5). This observation reflects the good behavior of the experiments regarding convergence of the UMDA. This fact together with the fairly flat shape of the curves corresponding to the BIC values of the best models found so far during the final generations indicate that further improvements are

unlike to occur, if more generations of the UMDA are considered in the experiments. Therefore, the stopping criterion used, i.e., 50 generations, seems to be a sensible choice for the databases in the evaluation. This makes the performance of the UMDA specially satisfactory: For the three databases in the evaluation, the UMDA identifies final models that score similar BIC values to those of the original models, by evaluating only 2525 solutions out of the approximately 1.2×10^{15} different solutions in the search space. Furthermore, the graphs in the second and the third rows of Table 5 indicate that, as the problem optimization process progresses, the best models found so far (indicated as UMDA in Table 5) increase their ability to generalize the learning data to previously unseen data as well as their closeness to the true model underlying the learning data. This supports the claim made previously that the (expected) BIC is a suitable objective function to optimize.

Finally, it must be said against the UMDA that it is typically more time consuming than the BSEM-UMDA and the BSEM-HC, despite the latter two evaluate a considerably larger number of solutions than the former. The reason is that every evaluation of a solution in the UMDA implies running the EM algorithm. This somehow bounds the scalability of the UMDA to domains of higher dimension than the ones in the evaluation. Unlike the UMDA, the BSEM-UMDA enjoys an interesting trade-off between effectiveness and efficiency, i.e., a trade-off between the quality of the final models and the computational cost of the unsupervised model learning process. The reason is in the generic BSEM algorithm: Treating expected data as real data makes possible the use of sophisticated search strategies, like EDAs, in order to solve the structural search step at each iteration effectively, and without compromising the efficiency of the whole unsupervised model learning process. Consequently, the BSEM-UMDA is a realistic approach, i.e., effective and scalable, to unsupervised learning of BNs.

4.3. *Results: Real data*

Answering biological questions through gene expression data analysis has been taken to a new level by the relatively recent development of *DNA microarray experiments*, which enable to monitor the expression levels of many genes simultaneously. This explains, at least partially, the fast-growing popularity and relevance that disciplines like, for instance, bioinformatics and biostatistics enjoy nowadays.

For the purpose of this paper, a DNA microarray experiment can be seen as a sequence of complex laboratory and computer related steps, whose output is usually presented in the form of a matrix with as many rows as samples, e.g., tissues, and as many columns as genes in the experiment, or vice versa. Each entry in this matrix measures the expression level of the corresponding gene in the corresponding sample. In this scenario, data clustering can help to identify clusters of samples sharing the same gene expression profile [11] and/or clusters of genes sharing the same expression profile across samples [23].

This section evaluates the UMDA and the BSEM-UMDA for gene expression

Table 6. Performance of the BNs for data clustering induced by the UMDA, the BSEM-UMDA, and the BSEM-HC from $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$.

		BIC		Log likelihood		Directed edges	
		Initial	Final	Initial	Final	Initial	Final
$\boldsymbol{d}_{ALL}$	BSEM-HC	-20656±0	-20292±0	-8795±0	-8590±0	0±0	8±0
	BSEM-UMDA	-20656±0	-20299±0	-8795±0	-8549±0	0±0	9±0
	UMDA	-20678±49	-20276±13	-8651±30	-8563±13	14±1	10±1
$\boldsymbol{d}_{AML}$	BSEM-HC	-21181±0	-21105±7	-9032±0	-8979±6	0±0	3±1
	BSEM-UMDA	-21181±0	-21105±7	-9032±0	-8979±6	0±0	3±1
	UMDA	-21437±9	-21090±4	-8991±23	-8952±11	12±1	5±1

data clustering, with the purpose of identifying clusters of genes with similar expression profiles across the samples in the leukemia database [11]. This database has become pretty much of a standard test bed for gene expression data analysis techniques [23,24]. It consists of 72 samples from leukemia patients, with each sample being characterized by the expression levels of 7129 genes. Besides, each sample is labelled with either acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML) meaning the specific type of acute leukemia the patient suffers from. Out of the 72 samples in the database, 47 are labelled as ALL and 25 as AML.

The preprocessing of the leukemia database was kept at minimum possible. This means that the original data were only discretized. Following the most common approach in the literature, gene expression levels were discretized into three states, corresponding to the concepts of a gene being either underexpressed, or baseline, or overexpressed with respect to its control expression level [25]. The discretization method that was used is based on information theory [24,26]. The resulting discretized database is referred to as $\boldsymbol{d}_{leukemia}$ in the forthcoming. From this, two auxiliary databases were created in order to analyze separately ALL and AML patients, due to their different gene expression profiles [11]. The first database grouped the first ten ALL patients in $\boldsymbol{d}_{leukemia}$. On the other hand, the second database contained the first ten AML patients in $\boldsymbol{d}_{leukemia}$. These two databases were then transposed, so that the 7129 genes were the cases and the measurements for the corresponding ten patients were the predictive attributes. The resulting learning databases are denoted $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$ in the forthcoming, where the subscript indicates the label of the corresponding patients. In the experiments below, the first 5000 cases of each database are used as learning data, and the last 2129 cases are set aside and used as testing data. Preliminary experiments with different numbers of clusters indicated that $K = 3$ is well suited for both $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$. Therefore, $K = 3$ in the experiments below. Finally, it should be mentioned that the cases in $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$ are treated as independent and identically distributed, although some genes may be co-regulated and, therefore, some cases may be correlated. This simplifies the analysis and may not change the essence of the results. In fact, this approach is taken in many other gene expression data analysis applications [23].

Table 7. Descriptions of the clusters of genes encoded by BN_{ALL} (left) and BN_{AML} (right).

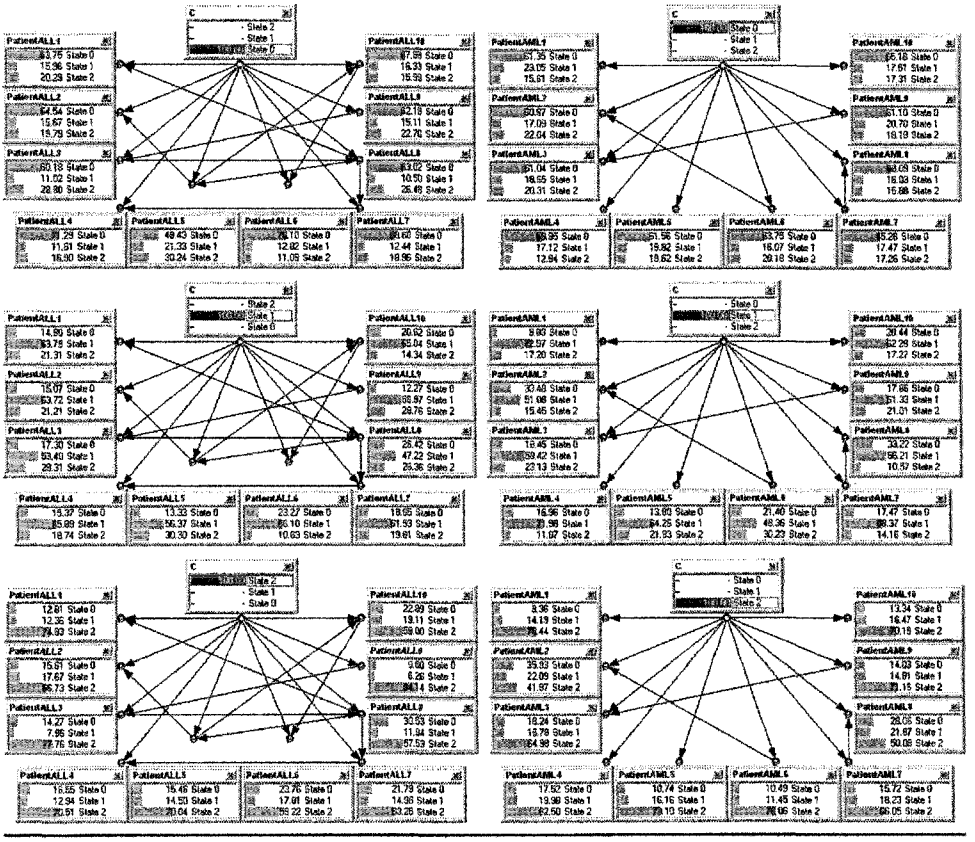


Table 6 summarizes the BIC and the log likelihood values corresponding to the BNs for data clustering that are induced by the UMDA, the BSEM-UMDA, and the BSEM-HC from $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$. The numbers of directed edges between unidimensional predictive random variables in the final models are also reported. All the performance criteria values in the table are given in terms of averages and standard deviations over five independent runs for the UMDA, and over ten independent runs for the BSEM-UMDA and the BSEM-HC. It can be appreciated from the table that, while no algorithm is clearly superior to the rest for $\boldsymbol{d}_{ALL}$, the UMDA outperforms the BSEM-UMDA and the BSEM-HC for $\boldsymbol{d}_{AML}$. In this last scenario, the BSEM-UMDA behaves as effectively as the BSEM-HC.

The remainder of this section provides more evidence about the effectiveness of the UMDA and the BSEM-UMDA. Specifically, the forthcoming paragraphs validate the interpretability and the homogeneity of the gene clusterings that are encoded by two of the models that are induced by the BSEM-UMDA, one from $\boldsymbol{d}_{ALL}$, BN_{ALL} for short, and one from $\boldsymbol{d}_{AML}$, BN_{AML} for short. Similar conclusions to

Table 8. Homogeneity of the clusters of genes encoded by BN_{ALL}.

	$\boldsymbol{d}_{ALL}$	$c_{ALL}^{underexpressed}$	$c_{ALL}^{baseline}$	$c_{ALL}^{overexpressed}$
Genes	7129	2265	2063	2801
$p(c^*)$	0.34±0.05	0.31±0.00	0.29±0.00	0.39±0.00
$p(c^* \mid \boldsymbol{y})$	0.90±0.14	0.89±0.14	0.89±0.15	0.90±0.14
$p(c^* \mid \boldsymbol{y})/p(c^*)$	2.68±0.55	2.85±0.45	3.04±0.51	2.29±0.35
$H(C)$	1.57±0.00	1.57±0.00	1.57±0.00	1.57±0.00
$H(C \mid \boldsymbol{y})$	0.37±0.40	0.38±0.39	0.39±0.41	0.36±0.38
$H(C)/H(C \mid \boldsymbol{y})$	76.85±223.38	41.77±81.65	152.87±379.22	49.22±98.55

Table 9. Homogeneity of the clusters of genes encoded by BN_{AML}.

	$\boldsymbol{d}_{AML}$	$c_{AML}^{underexpressed}$	$c_{AML}^{baseline}$	$c_{AML}^{overexpressed}$
Genes	7129	2678	1713	2738
$p(c^*)$	0.35±0.05	0.37±0.00	0.25±0.00	0.38±0.00
$p(c^* \mid \boldsymbol{y})$	0.91±0.14	0.91±0.14	0.89±0.15	0.92±0.13
$p(c^* \mid \boldsymbol{y})/p(c^*)$	2.71±0.62	2.46±0.37	3.52±0.59	2.46±0.35
$H(C)$	1.56±0.00	1.56±0.00	1.56±0.00	1.56±0.00
$H(C \mid \boldsymbol{y})$	0.32±0.40	0.31±0.40	0.39±0.41	0.28±0.38
$H(C)/H(C \mid \boldsymbol{y})$	555.27±1753.93	754.62±2305.34	209.88±650.83	576.38±1560.30

those below can be achieved by considering any other pair of models that are learnt via the UMDA and the BSEM-UMDA from $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$. It should be noticed that all the 7129 cases, i.e., genes, in $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$ are involved in the analysis below, and not only those 5000 that were used for learning BN_{ALL} and BN_{AML}.

Table 7 outlines the descriptions of the clusters of genes that BN_{ALL} and BN_{AML} represent. Concretely, each component of Table 7 illustrates the marginal probability distributions of the unidimensional predictive random variables, i.e., gene expression levels corresponding to patients, conditioned on one of the states of the cluster random variable. These figures suggest that, in both BN_{ALL} and BN_{AML}, cluster 0 exhibits a tendency towards all the predictive random variables being in state 0 (underexpressed), cluster 1 towards being in state 1 (baseline) and, finally, cluster 2 towards being in state 2 (overexpressed). It seems sensible and believable that the generative mechanisms underlying $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$ consist of three physical processes each, with each physical process being governed by a joint probability distribution that tends towards genes being either underexpressed, or baseline, or overexpressed for all the patients. This description of the generative mechanism underlying $\boldsymbol{d}_{ALL}$ ($\boldsymbol{d}_{AML}$) seems specially convincing if one takes into account that the patients in $\boldsymbol{d}_{ALL}$ ($\boldsymbol{d}_{AML}$) all suffer from the same type of acute leukemia.

Given a BN for data clustering and a case with predictive attributes $\boldsymbol{y}$, let $p(c)$ and $p(c \mid \boldsymbol{y})$ denote, respectively, the prior and the posterior probability distributions for the cluster random variable C. Also, let $H(C)$ and $H(C \mid \boldsymbol{y})$ represent,

Table 10. Error rates of the NB, the IB1-1, and the IB1-3 for $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$.

	NB	IB1-1	IB1-3
$\boldsymbol{d}_{ALL}$	5.54±0.25	6.06±0.22	4.74±0.20
$\boldsymbol{d}_{AML}$	2.85±0.22	7.60±0.27	3.31±0.18

respectively, the entropy of $p(c)$ and $p(c \mid \boldsymbol{y})$. A straightforward way of assessing the homogeneity of the clusters of genes that BN_{ALL} (BN_{AML}) encodes is by averaging over the genes in $\boldsymbol{d}_{ALL}$ ($\boldsymbol{d}_{AML}$) the values of $p(c^* \mid \boldsymbol{y})$, $p(c^* \mid \boldsymbol{y})/p(c^*)$, $H(C \mid \boldsymbol{y})$, and $H(C)/H(C \mid \boldsymbol{y})$, where $c^* = arg\, max_{C=c} p(c \mid \boldsymbol{y})$. The higher the value of the first, the second, and the fourth averages the more homogeneous the clusters. On the other hand, the lower the value of the third average the better. Additionally, the homogeneity of each cluster alone can be assessed by averaging the previous criteria only over the genes belonging to that particular cluster. Note that for this analysis, genes need to be hard-assigned to clusters. Consequently, every case in $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$ is completed with c^*. For the sake of readability, clusters 0, 1, and 2 in Table 7 are denoted $c_{ALL}^{underexpressed}$, $c_{ALL}^{baseline}$, and $c_{ALL}^{overexpressed}$, respectively, for $\boldsymbol{d}_{ALL}$, and $c_{AML}^{underexpressed}$, $c_{AML}^{baseline}$, and $c_{AML}^{overexpressed}$, respectively, for $\boldsymbol{d}_{AML}$.

Table 8 and Table 9 compile the values of the homogeneity criteria for the whole $\boldsymbol{d}_{ALL}$ and $\boldsymbol{d}_{AML}$ as well as for each of the clusters underlying them alone. The size of each cluster, after hard-assigning cases to clusters, is also reported. The results in the tables confirm that the six clusters of genes are homogeneous, being the clusters underlying $\boldsymbol{d}_{AML}$ slightly more homogeneous than those underlying $\boldsymbol{d}_{ALL}$. Specifically, the high values of $p(c^* \mid \boldsymbol{y})$ and the low values of $H(C \mid \boldsymbol{y})$ for all the clusters indicate that cases can be hard-assigned to clusters with little uncertainty. The same conclusion can be reached by looking at the ratios $p(c^* \mid \boldsymbol{y})/p(c^*)$ and $H(C)/H(C \mid \boldsymbol{y})$: In all the cases, there is a significant reduction in uncertainty when comparing the prior and the posterior probability distributions. These are clear signs of homogeneity.

More evidence about the homogeneity of the clusters of genes that are represented by BN_{ALL} and BN_{AML} can be given through supervised classification. If the clusters are homogeneous then, after hard-assigning genes to them, supervised classifiers should do well at predicting the cluster membership of the genes. Table 10 reports the error rates for this task, in terms of averages and standard deviations estimated via 10-fold cross-validation, of three well known supervised classifiers: The naive Bayes classifier [27], NB for short, and the IB1 classifier [28] with one and three neighbors, IB1-1 and IB1-3, respectively, for short. The error rates in the table are rather low and, therefore, support the claim made above that the clusters of genes are homogeneous.

It is known that the accuracy of supervised classifiers is not monotonic with respect to the inclusion of predictive attributes [29], i.e., the inclusion of irrelevant

Table 11. Error rates of the NB, the IB1-1, and the IB1-3 for $\boldsymbol{d}_{leukemia}$, when different sets of genes are considered as predictive attributes.

	Genes	NB	IB1-1	IB1-3
All	7129	0.00±0.00	0.00±0.00	1.39±1.39
All differentially expressed	3160	1.39±1.39	0.00±0.00	0.00±0.00
$c_{ALL}^{underexpressed} \cap c_{AML}^{baseline}$	352	2.78±1.95	8.33±3.28	6.94±3.02
$c_{ALL}^{underexpressed} \cap c_{AML}^{overexpressed}$	407	8.33±3.28	5.56±2.72	6.94±3.02
$c_{ALL}^{baseline} \cap c_{AML}^{underexpressed}$	557	4.17±2.37	2.78±1.95	1.39±1.39
$c_{ALL}^{baseline} \cap c_{AML}^{overexpressed}$	687	5.56±2.72	1.39±1.39	0.00±0.00
$c_{ALL}^{overexpressed} \cap c_{AML}^{underexpressed}$	615	2.78±1.95	5.56±2.72	6.94±3.02
$c_{ALL}^{overexpressed} \cap c_{AML}^{baseline}$	542	2.78±1.95	4.17±2.37	2.78±1.95

and/or redundant predictive attributes may degrade their accuracy. For this reason, supervised classifiers aiming at discriminating among known classes of samples in gene expression data analysis usually focus on what are called *differentially expressed genes* [11,23,24], i.e., genes whose expression levels vary significantly from one class of samples to another. Based on these observations, the homogeneity of the clusters of genes that BN_{ALL} and BN_{AML} represent can be further validated as follows. Under the assumption that these clusters are homogeneous, differentially expressed genes for $\boldsymbol{d}_{leukemia}$ can be easily detected: After hard-assigning genes to clusters, genes belonging to any of the intersections of clusters $c_{ALL}^{labelALL} \cap c_{AML}^{labelAML}$ for $labelALL, labelAML = underexpressed, baseline$, and $overexpressed$ such that $labelALL \neq labelAML$ can be deemed differentially expressed. Table 11 illustrates the error rates, in terms of averages and standard deviations estimated via leave-one-out cross-validation, of the NB, the IB1-1, and the IB1-3 for predicting the specific type of acute leukemia, i.e., either ALL or AML, of the 72 patients in $\boldsymbol{d}_{leukemia}$. The table reports results for the original database $\boldsymbol{d}_{leukemia}$, i.e., 7129 genes characterize each sample, for $\boldsymbol{d}_{leukemia}$ restricted to all the differentially expressed genes, and for $\boldsymbol{d}_{leukemia}$ restricted to those genes in each of the intersections of clusters $c_{ALL}^{labelALL} \cap c_{AML}^{labelAML}$ for $labelALL, labelAML = underexpressed, baseline$, and $overexpressed$ such that $labelALL \neq labelAML$. Table 11 also shows the number of genes that are included in the supervised classifiers. From the results in the table, it can be concluded that the supervised classifiers that are induced from $\boldsymbol{d}_{leukemia}$ restricted to the 3160 differentially expressed genes perform as well as those that involve all the 7129 genes. Furthermore, the supervised classifiers that are learnt from each of the intersections of pairs of clusters under study are very accurate in general, despite the significant reduction in the number of genes characterizing each case in the learning data. Therefore, the homogeneity of the clusters of genes that are identified in this work is again confirmed.

As summary, it can be said that the gene clusterings that BN_{ALL} and BN_{AML} encode have natural and sensible interpretations and conform homogeneous group-

structures. This suggests that these gene clusterings may be meaningful for the biologist. Further validation using biological knowledge is required to confirm this.

5. Conclusions

The contribution of this paper has been twofold. First, the proposal and empirical evaluation of EDAs for unsupervised learning of BNs, both directly and within the framework of the BSEM algorithm. Second, the application of this paper's proposals to gene expression data analysis, and in particular to gene expression data clustering, which is one of the most challenging research areas nowadays.

The evaluation has been limited to one of the simplest EDAs, namely the UMDA. Both the UMDA and the BSEM-UMDA have behaved effectively in synthetic and real data. However, only the BSEM-UMDA seems to scale well to high-dimensional domains. This trade-off between effectiveness and efficiency makes the BSEM-UMDA attractive. An issue for further research may be the evaluation of EDAs more sophisticated than the UMDA within the BSEM algorithm framework. Regarding the application to gene expression data clustering, both approaches have identified similar gene clusterings. The extensive validation of these clusters of genes has indicated that these may be biologically meaningful.

Acknowledgements

The authors thank I. Inza and R. Blanco for making the discretized leukemia database available. The authors are also indebted to I. Inza for his help in using the MLC++ software library. J. A. Lozano and P. Larrañaga were supported by the Spanish Ministry of Science and Technology under grant TIC2001-2973-C05-03.

References

1. E. Castillo, J. M. Gutiérrez and A. S. Hadi, *Expert Systems and Probabilistic Network Models* (Springer-Verlag, 1997).
2. F. V. Jensen, *Bayesian Networks and Decision Graphs* (Springer-Verlag, 2001).
3. J. Pearl, *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufmann Publishers, 1988).
4. J. M. Peña, J. A. Lozano and P. Larrañaga, "Learning Bayesian Networks for Clustering by Means of Constructive Induction", *Pattern Recognition Letters* **20** (1999) 1219–1230.
5. J. M. Peña, J. A. Lozano and P. Larrañaga, "An Improved Bayesian Structural EM Algorithm for Learning Bayesian Networks for Clustering", *Pattern Recognition Letters* **21** (2000) 779–786.
6. J. M. Peña, J. A. Lozano and P. Larrañaga, "Learning Recursive Bayesian Multinets for Data Clustering by Means of Constructive Induction", *Machine Learning* **47** (2002) 63–89.
7. P. Larrañaga and J. A. Lozano (eds.), *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation* (Kluwer Academic Publishers, 2001).
8. H. Mühlenbein and G. Paaß, "From Recombination of Genes to the Estimation of Distributions I. Binary Parameters", in *Proceedings of Parallel Problem Solving from Nature IV* (1996) pp. 178–187.

9. M. Pelikan, D. E. Goldberg and F. G. Lobo, "A Survey of Optimization by Building and Using Probabilistic Models", *Computational Optimization and Applications* **21** (2002) 5–20.
10. N. Friedman, "The Bayesian Structural EM Algorithm", in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (1998) pp. 129–138.
11. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield and E. S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring", *Science* **286** (1999) 531–537.
12. D. M. Chickering, "Learning Bayesian Networks is NP-Complete", in *Learning from Data: Artificial Inteligence and Statistics V* (1996) pp. 121–130.
13. D. Heckerman, D. Geiger and D. M. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data", *Machine Learning* **20** (1995) 197–243.
14. D. M. Chickering, "Learning Equivalence Classes of Bayesian-Network Structures", *Journal of Machine Learning Research* **2** (2002) 445–498.
15. G. Schwarz, "Estimating the Dimension of a Model", *Annals of Statistics* **6** (1978) 461–464.
16. A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the EM Algorithm", *Journal of the Royal Statistical Society B* **39** (1977) 1–38.
17. G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions* (John Wiley and Sons, 1997).
18. T. Bäck, *Evolutionary Algorithms in Theory and Practice* (Oxford University Press, 1996).
19. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, 1989).
20. H. Mühlenbein, "The Equation for Response to Selection and Its Use for Prediction", *Evolutionary Computation* **5** (1997) 303–346.
21. D. M. Chickering, "Learning Equivalence Classes of Bayesian Network Structures", in *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence* (1996) pp. 150–157.
22. J. M. Peña, J. A. Lozano and P. Larrañaga, "Unsupervised Learning of Bayesian Networks Via Estimation of Distribution Algorithms", in *Proceedings of the First European Workshop on Probabilistic Graphical Models* (2002) pp. 144–151.
23. A. Ben-Dor, N. Friedman and Z. Yakhini, "Class Discovery in Gene Expression Data", in *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology* (2001) pp. 31–38.
24. I. Inza, P. Larrañaga, R. Blanco and A. J. Cerrolaza, "Filter Versus Wrapper Gene Selection Approaches in DNA Microarray Domains", submitted (2003).
25. N. Friedman, M. Linial, I. Nachman and D. Pe'er, "Using Bayesian Networks to Analyze Expression Data", *Journal of Computational Biology* **7** (2000) 601–620.
26. M. Beibel, "Selection of Informative Genes in Gene Expression Based Diagnosis: A Nonparametric Approach", in *Proceedings of the First International Symposium in Medical Data Analysis* (2000) pp. 300–307.
27. B. Cestnik, "Estimating Probabilities: A Crucial Task in Machine Learning", in *Proceedings of the European Conference on Artificial Intelligence* (1990) pp. 147–149.
28. D. W. Aha, D. Kibler and M. K. Albert, "Instance-Based Learning Algorithms", *Machine Learning* **6** (1991) 37–66.
29. R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection", *Artificial Intelligence* **97** (1997) 273–324.