Unsupervised Learning of Bayesian Networks Via Estimation of Distribution Algorithms

J. M. Peña
jmp@cs.auc.dk
Aalborg University
Denmark

J. A. Lozano and P. Larrañaga
{ccploalj, ccplamup}@si.ehu.es
University of the Basque Country
Spain

Abstract

This paper proposes the use of estimation of distribution algorithms for unsupervised learning of Bayesian networks. The empirical results reported for one of the simplest estimation of distribution algorithms confirm the ability of this approach to induce models that show (i) similar fitness of the learning data to that of the original models, (ii) satisfactory generalization of the learning data to previously unseen data, and (iii) closeness to the original models in terms of conditional (in)dependence statements encoded.

1 Introduction

One of the main problems that arises in a great variety of fields, including artificial intelligence 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 finding 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 is called a *cluster*.

Among the different interpretations and expectations that the term data clustering gives rise to, this paper is limited to those data clustering problems defined basically 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. Furthermore, the l-th case of $\boldsymbol{d}$ is represented by an $(n+1)$-dimensional vector $\boldsymbol{x}_l = (x_{l1}, \ldots, x_{ln+1}) = (c_l, y_{l1}, \ldots, y_{ln})$, where 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 underlying clusters in $\boldsymbol{d}$ corresponds to a physical process defined by an unknown joint probability distribution. Then, every case of $\boldsymbol{d}$ may be seen as sampled from exactly one of these K unknown joint probability distributions. Consequently, it is assumed the existence of an $(n+1)$-dimensional random variable $\boldsymbol{X} = (X_1, \ldots, X_{n+1})$ partitioned as $\boldsymbol{X} = (C, \boldsymbol{Y})$ into a unidimensional discrete random variable C, i.e., the *cluster random variable*, that represents the cluster membership, and an n-dimensional discrete random variable $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$, i.e., the *predictive random variable*, that represents the set of predictive attributes. Moreover, it is usual to assume that the mechanism that generated $\boldsymbol{d}$ works in two stages: First, one of the physical processes 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 the selected physical process defines.
- The parametric forms of the joint probability distributions that govern the mechanism that generated $\boldsymbol{d}$ are all multinomial.

Under these assumptions, data clustering is usually approached from the *probabilistic* or *model-based* perspective: The description of the K underlying clusters of $\boldsymbol{d}$ is accomplished through the probabilistic modelling of the mechanism that generated $\boldsymbol{d}$. Consequently, solving a data clustering problem reduces to learning a joint probability distribution for $\boldsymbol{X}$ from $\boldsymbol{d}$. This paper approaches this task by learning a Bayesian network for data clustering for $\boldsymbol{X}$ from $\boldsymbol{d}$. In the forthcoming, *unsupervised learning* is used as a synonym of data clustering.

A *Bayesian network (BN) for data clustering* for $\boldsymbol{X}$ is a graphical factorization of a joint probability distribution for $\boldsymbol{X}$, $p(\boldsymbol{x})$. A BN for data clustering for $\boldsymbol{X}$ consists of (i) a directed acyclic graph (DAG) $\boldsymbol{s}$, whose nodes correspond to the unidimensional random variables of $\boldsymbol{X}$, determining the conditional (in)dependencies between the random variables of $\boldsymbol{X}$, i.e., the *model structure*, and (ii) a set of local probability distributions for $\boldsymbol{s}$. Moreover, it is common to constraint $\boldsymbol{s}$ so that every Y_i is a child of C. This structural constraint is somehow imposed by the assumption about how the mechanism that generated $\boldsymbol{d}$ works.

The joint probability distribution for $\boldsymbol{X}$ encoded by a BN for data clustering for $\boldsymbol{X}$ graphically factorizes as follows:

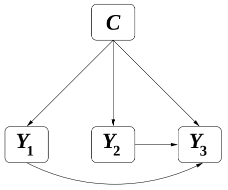
$$p(\boldsymbol{x} \mid \boldsymbol{\theta_s}, \boldsymbol{s}^h) = p(c \mid \boldsymbol{\theta_s}, \boldsymbol{s}^h) p(\boldsymbol{y} \mid c, \boldsymbol{\theta_s}, \boldsymbol{s}^h)$$

$$= p(c \mid \boldsymbol{\theta}_C, \boldsymbol{s}^h) \prod_{i=1}^{n} p(y_i \mid c, \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i, \boldsymbol{\theta}_i, \boldsymbol{s}^h) \quad (1)$$

where $\boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i$, with $\boldsymbol{s}^{\boldsymbol{Y}}$ the subgraph of $\boldsymbol{s}$ induced by $\boldsymbol{Y}$, denotes the state of those parents of Y_i that correspond to predictive random variables, $\boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i$, for all i. The local probability distributions of the BN for data clustering for $\boldsymbol{X}$ are those induced by the terms in Equation 1, and they 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, $\boldsymbol{s}^h$ denotes the hypothesis that the true joint probability distribution for $\boldsymbol{X}$ can be graphically factorized according to the conditional independencies encoded in $\boldsymbol{s}$.

Typically, the local probability distributions of a BN for data clustering for $\boldsymbol{X}$ are univariate multinomial distributions. As K clusters exist, C can take K distinct values denoted by $c^1, \ldots, c^K$. Then, $p(c \mid \boldsymbol{\theta}_C, \boldsymbol{s}^h)$ consists of a set of probabilities of the form $p(c^g \mid \boldsymbol{\theta}_C, \boldsymbol{s}^h) = \theta^g > 0$ for all g. Furthermore, $\sum_{g=1}^{K} \theta^g = 1$ and $\boldsymbol{\theta}_C = (\theta^1, \ldots, \theta^K)$. Besides, let us assume that Y_i can take r_i distinct values denoted by $y_i^1, \ldots, y_i^{r_i}$, and that $\boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i$ can have q_i distinct states denoted by $\boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^1, \ldots, \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^{q_i}$ with $q_i = \prod_{Y_e \in \boldsymbol{Pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i} r_e$ for all i. Then, $p(y_i \mid c^g, \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^j, \boldsymbol{\theta}_i, \boldsymbol{s}^h)$ for all g, i, and j consist of a set of probabilities of the form $p(y_i^k \mid c^g, \boldsymbol{pa}(\boldsymbol{s}^{\boldsymbol{Y}})_i^j, \boldsymbol{\theta}_i, \boldsymbol{s}^h) = \theta_i^{gjk} > 0$ for all k. Moreover, $\sum_{k=1}^{r_i} \theta_i^{gjk} = 1$, and $\boldsymbol{\theta}_i = (\boldsymbol{\theta}_i^g)_{g=1}^{K}$ 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, i, and j. Figure 1 shows an example of a BN for data clustering.

As seen in Equation 1, the description of the K clusters in $\boldsymbol{d}$ encoded by a BN for data clustering for $\boldsymbol{X}$ induced from $\boldsymbol{d}$ consists of (i) $p(c \mid \boldsymbol{\theta_s}, \boldsymbol{s}^h)$ which represents a probability distribution for C modelling how one of the clusters was selected by the mechanism that generated $\boldsymbol{d}$, and (ii) $p(\boldsymbol{y} \mid c^g, \boldsymbol{\theta_s}, \boldsymbol{s}^h)$ for all g which represent a set of conditional joint probability distributions for $\boldsymbol{Y}$ given a state of C modelling how the mechanism caused every instance summarized in $\boldsymbol{d}$, after a cluster was selected. Note that $p(\boldsymbol{y} \mid c^g, \boldsymbol{\theta_s}, \boldsymbol{s}^h)$ for all g graphically factorize further according to the conditional independencies encoded in $\boldsymbol{s}$ (see Equation 1). Therefore, 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 most cases. Among the different approaches to this task, this paper is concerned with the interpretation of unsupervised learning of BNs as an optimization problem, where the search space is the space of structures of BNs for data clustering, and the objective function assesses the quality of every solution in the search space with respect to the learning database $\boldsymbol{d}$ and, possibly, some prior knowledge. Note that the search space can be restricted to the space of DAGs for $\boldsymbol{Y}$, due to the structural constraint imposed on BNs for data clustering, i.e., every Y_i is a child of C. As the result reported by Chicker-



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

Figure 1: Model structure (left), model parameters (middle), and local probability distributions (right) of a BN for data clustering for $\boldsymbol{X} = (C, \boldsymbol{Y}) = (C, Y_1, Y_2, Y_3)$. It is assumed that two clusters exist and that all the unidimensional predictive random variables are binary.

ing (1996a) applies, this approach to unsupervised learning of BNs constitutes an NP-hard optimization problem, when the objective function is the Bayesian Dirichlet equivalent score. Moreover, it is assumed that this hardness holds for other common scores as well, though there is not yet a formal proof. These results justify the heuristic nature of the majority of algorithms for unsupervised learning BNs. To our knowledge, only deterministic heuristic search strategies have received attention for this task (e.g., (Peña et al., 1999; Peña et al., 2002)), mainly inspired by the Bayesian structural EM (BSEM) algorithm (Friedman, 1998). However, the complexity of unsupervised learning of BNs calls for the consideration of randomness in order to overcome some drawbacks of deterministic heuristic search strategies, such as local optimality and dependence on the initial solution.

The remainder of this paper is structured as follows. Section 2 introduces the class of stochastic heuristic search strategies that this paper proposes for unsupervised learning of BNs: A relatively novel family of evolutionary algorithms called estimation of distribution algorithms. Section 3 compiles some experimental results that confirm the effectiveness of one of the simplest estimation of distribution algorithms for unsupervised learning of BNs. Finally, some conclusions are drawn in Section 4.

2 Estimation of Distribution Algorithms

The *evolutionary algorithm (EA)* paradigm groups a set of stochastic heuristic search strategies for problem optimization whose main feature is that of being inspired by natural evolution of species. That is why much of the nomenclature of EAs is borrowed from the field of natural evolution: A *population* refers to a set of solutions, each solution is called an *individual*, and each basic component of an individual is named a *gene*. The main elements of most EAs are: An initial population, a *selection method* over individuals, a set of *random operators* over individuals, and a *replacement method* over individuals. 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 through the replacement method. The objective of this iterative process is to evolve the population towards promising zones of the search space of the optimization problem at hand.

The main advantages of EAs for problem optimization are their wide applicability and good performance. Unfortunately, EAs are not exempt of drawbacks. For instance, some optimization problems may require the design of specific random operators. However, the worst characteristic of most EAs is probably their dependence on a set of parameters (number of generations, size of the population, probabilities for applying the random operators, etc.) that have to be experimentally tuned for the optimization problem and the particular EA at hand. With the aim of overcoming some of these drawbacks, a novel family of EAs, known as *estimation of distribution algorithms (EDAs)*, has been recently proposed. EDAs do so by replacing the application of random operators at each iteration by learning and, subsequently, simulating a joint probability distribution for a database conformed with those individuals selected from the current population by means of the selection method. The generic EDA iterates between three main steps, after the individuals of the initial population $\boldsymbol{po}_1$ have been generated and evaluated. The iterative process ends when the stopping criterion is met. This causes the best solution found so far to be returned. The three main steps are as follows for the u-th iteration for all u. First, M of the Q individuals of the current population $\boldsymbol{po}_u$ are selected according to the selection method. Then, these individuals are used to construct a learning database $\boldsymbol{d}_u$ 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 $\boldsymbol{d}_u$. It should be noticed that $\boldsymbol{Z}$ is limited to be discrete due to the combinatorial nature of the optimization problem that this paper addresses, i.e., unsupervised induction of BNs. However, this can be fully generalized to deal with continuous optimization problems. 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 $\boldsymbol{po}_u$ according to the replacement method. See Figure 2 for a schematic of the generic EDA.

```
Create $\boldsymbol{po}_1$ by randomly generating $Q$ individuals
Evaluate the individuals in $\boldsymbol{po}_1$
$u$=1
while the stopping criterion is not met do
    Create $\boldsymbol{d}_u$ by selecting $M$ individuals from $\boldsymbol{po}_u$
    Learn $p_u(\boldsymbol{z})$ from $\boldsymbol{d}_u$
    Create $\boldsymbol{of}_u$ by sampling $R$ instances from $p_u(\boldsymbol{z})$
    Evaluate the individuals in $\boldsymbol{of}_u$
    Create $\boldsymbol{po}_{u+1}$ by merging $\boldsymbol{po}_u$ and $\boldsymbol{of}_u$
    $u$++
Return the best individual found so far
```

Figure 2: Pseudocode of the generic EDA.

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 needed to completely specify this joint probability distribution in the standard representation is often impractical. Therefore, 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. The reader is referred to the book by Larrañaga and Lozano (2001) for a review of EDAs for combinatorial and continuous problem optimization.

3 Empirical Evaluation

3.1 Evaluation Setup

The empirical evaluation of the approach proposed in this paper involves one of the simplest EDAs known as the *univariate marginal distribution algorithm (UMDA)* (Mühlenbein, 1997). The generic UMDA is based on the assumption that $p_u(\boldsymbol{z}) = \prod_{i=1}^{m} p_u(z_i)$ for all u. Moreover, $p_u(z_i)$ is restricted to be a univariate multinomial distribution whose parameters are estimated from $\boldsymbol{d}_u$ according to the maximum likelihood criterion for all i and u. In the forthcoming, the term UMDA refers to the problem specific instance of the generic UMDA employed in the evaluation.

The representation considered in the UMDA for every solution $\boldsymbol{s^Y}$ in the search space uses an $n \times n$ adjacency matrix $\boldsymbol{a} = (a_{ij})$, such that (i) $a_{ij} = 2$ if $Y_j \in \boldsymbol{Pa}(\boldsymbol{s^Y})_i$, (ii) $a_{ij} = 1$ if $Y_i \in \boldsymbol{Pa}(\boldsymbol{s^Y})_j$, 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. As the creation of $\boldsymbol{po}_1$ and $\boldsymbol{of}_u$ for all u is not closed with respect to the DAG property, individuals representing invalid solutions are likely to be produced by the UMDA. An invalid solution is repaired by, iteratively, removing a randomly chosen directed edge that invalidates the DAG property until a DAG is obtained. This decoding process, if needed, does not repair individuals but the solutions represented by them.

The objective function for the UMDA is the Bayesian information criterion (BIC). A multiple-restart version of the EM algorithm is employed to estimate the maximum likelihood model parameters for a given model structure.

The selection and the replacement methods of the 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 generated offspring R are set to 75, 25, and 50, respectively. The UMDA halts after 50 generations, i.e., after evaluating 2525 solutions.

The evaluation is carried out on 3 synthetic databases obtained by sampling 3 BNs for data clustering of increasing complexity. The knowledge of the 3 original models underlying the databases enables to assess the performance of the UMDA by comparing the original and the elicited models. The 3 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 the 3 original models was 10, 15, and 20. These directed edges were uniformly generated, as far as no cycle was created. Note that each of the 3 original models had 9 additional directed edges due to the structural constraint imposed on BNs for data clustering, i.e., every Y_i was a child of C. The model parameters for each of the 3 original models were generated at random. From each of the 3 original models 5000 cases were sampled. Each case consisted only of a state for $\boldsymbol{Y}$, i.e., all the entries for C in the samples were missing. The first 4000 cases of each sample were used as learning data and, the last 1000 cases were set aside and used as testing data. In the forthcoming, the 3 learning databases are referred to as $\boldsymbol{d}10$, $\boldsymbol{d}15$, and $\boldsymbol{d}20$, respectively.

In the evaluation, the performance of the UMDA is assessed according to its 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) similarity to the true model underlying the learning data. The BIC values scored by the induced models serve for assessing the first ability. The second ability can be measured by calculating the log likelihood of each hold-out database given the corresponding elicited model. Finally, the similarity between each learnt model and the corresponding original model should be assessed by measuring the closeness between the equivalence classes to which these models belong. For this purpose, the approach proposed by Chickering (1996b) is taken: First, the completed partially directed acyclic graphs (CPDAGs) for both equivalence classes are generated and, then, the number of edges that are different in these 2 graphs is reported.

It should be emphasized that the best indicators of the performance of the UMDA are the BIC values of the learnt BNs for data clustering, because the BIC is the objective function that guides the search. The other 2 performance measures considered are also relevant for the evaluation, though secondary.

For comparison purposes, 2 instances of the BSEM algorithm are used as benchmarks. The first instance, referred to as BSEM+HC, reduces the maximization step of each iteration of the generic BSEM algorithm to a 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 arc at each point in the search. The score that guides the search at each maximization step of the BSEM+HC is the expected BIC with respect to the best model found so far. On the other hand, the second instance, referred to as BSEM+UMDA, solves the maximization step via the UMDA but, in this case, the objective function for the UMDA is the expected BIC with respect to the best model found so far, and $Q = 7500$, $M = 2500$, and $R = 5000$.

3.2 Results

The graphs in Figure 3 show the performance of the BNs for data clustering induced by the UMDA from $\boldsymbol{d}10$, $\boldsymbol{d}15$, and $\boldsymbol{d}20$, as a function of the number of generations. The graphs in the first row report the BIC values for the learning databases, those in the second row measure the log likelihood of the hold-out databases, and those in the third row show the distances between the CPDAGs of the equivalence classes of the original and the induced models. Table 1 complements Figure 3 with a comparison of the performance of the models obtained via the BSEM+HC, the BSEM+UMDA, and the UMDA. All the performance criteria values are given in terms of average and standard deviation over 5 independent runs for the UMDA, and over 10 independent runs for the 2 benchmarks. The performance criteria values of the original models are also given.

The most important observation that can be gained from Figure 3 and Table 1 is that the UMDA is able to evolve the populations towards models that perform well with respect to the objective function, i.e., the BIC, independently of the complexity of the learning databases. Moreover, the final models considerably improve the initial models, and can be totally compared with the original models in terms of BIC values. The small standard deviation values reported in Table 1 for the BIC values reflect the robustness of the UMDA for the databases in the evaluation.

It is also interesting to note in the graphs of the first row of Figure 3 that, as the number of generations increases, the curves corresponding to the BIC values of the population averages (UMDA population in Figure 3) get closer to the curves corresponding to the BIC values of the best models found so far. This observation reflects the good behavior of the experiments regarding convergence. Furthermore, 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 a sensible choice for the databases in the evaluation. This makes the good performance of the UMDA especially satisfactory: For the 3 databases in the evaluation, the UMDA identifies final models that perform similarly well as the original models by evaluating only 2525 solutions out of the approximately 1.2×10^{15} different solutions in the search space. Similar conclusions are reported by Blanco et al. (2002) when the UMDA is applied to BN induction from complete data.

Regarding the 2 secondary performance measures, i.e., ability to generalize the learning data to previously unseen data and closeness to the true model underlying the learning data, it can be said that the approach proposed in this paper performs satisfactorily for $\boldsymbol{d}10$, $\boldsymbol{d}15$, and $\boldsymbol{d}20$: As the problem optimization process progresses, there is an increase in the values that the best models found so far by the UMDA score for both performance criteria (see the graphs of the second and the third rows of Figure 3). Therefore, these results confirm that the 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.

The reliability of the UMDA to recover the true models underlying $\boldsymbol{d}10$, $\boldsymbol{d}15$, and $\boldsymbol{d}20$ can be appreciated as follows. Table 1 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 unidi-

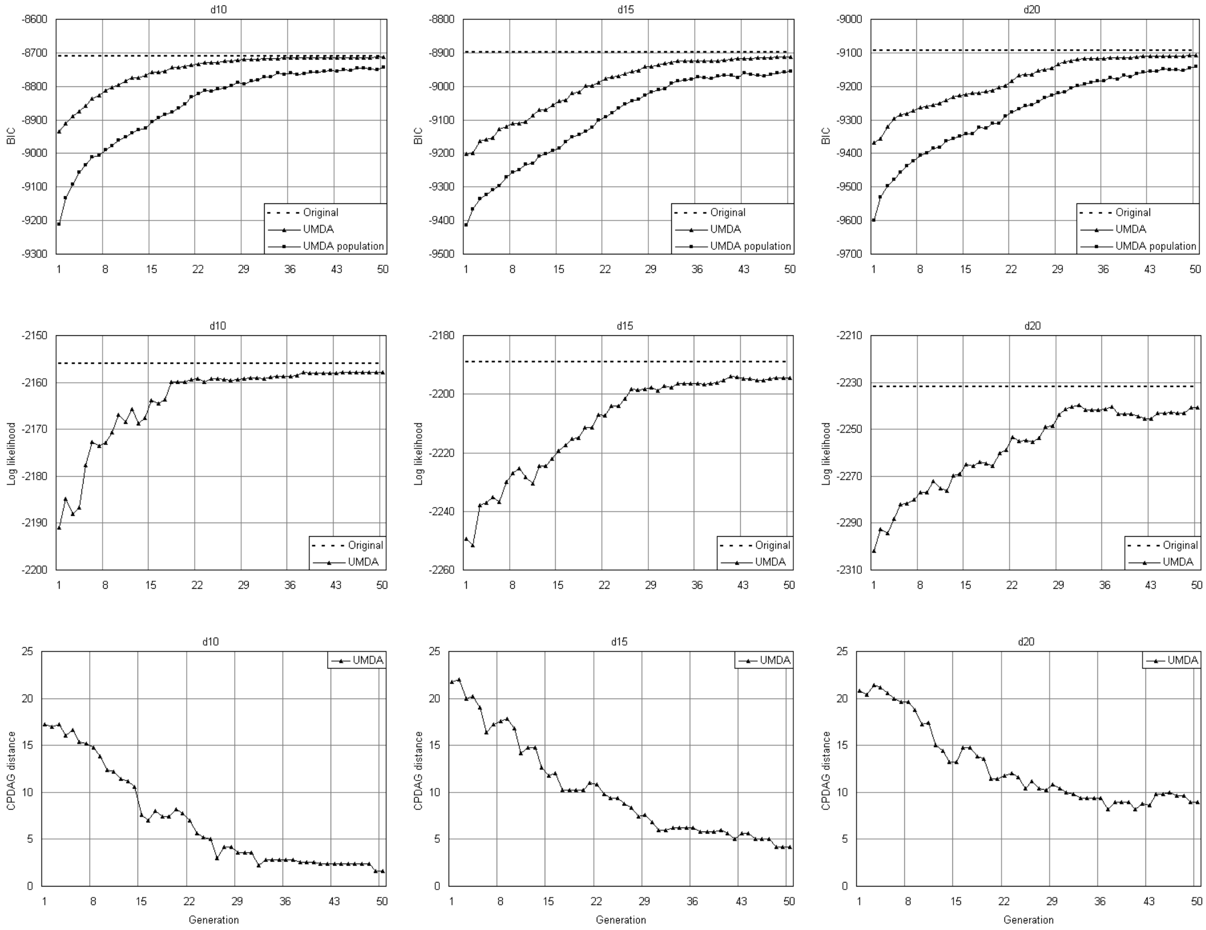


Figure 3: Performance of the BNs for data clustering induced by the UMDA from $\boldsymbol{d}10$, $\boldsymbol{d}15$, and $\boldsymbol{d}20$, as a function of the number of generations.

Table 1: Comparison of the performance of the BNs for data clustering induced by the BSEM+HC, the BSEM+UMDA, and the UMDA from $\boldsymbol{d}10$, $\boldsymbol{d}15$, and $\boldsymbol{d}20$.

		BIC		*Log likelihood*		*CPDAG distance*	
Data	*Model*	*Initial*	*Final*	*Initial*	*Final*	*Initial*	*Final*
$\boldsymbol{d}10$	Original	−8709	−8709	−2156	−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	−8898	−2189	−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	−9094	−2232	−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

mensional predictive random variables. Then, the number of relationships that coincide in the CPDAGs corresponding to the equivalence classes of the original model and the one learnt by the UMDA is, on average, 34 (94 %) for $\boldsymbol{d}10$, 32 (89 %) for $\boldsymbol{d}15$, and 27 (75 %) for $\boldsymbol{d}20$.

As expected, the performance of the induced models with respect to the 2 secondary performance measures degrades as the complexity of the models underlying the learning data increases. However, the UMDA is always able to learn models that perform very well in terms of the BIC, which is the primary performance criterion. It is an open question as to whether considering EDAs more sophisticated than the UMDA and/or larger learning databases would yield to better results for the 2 secondary performance measures, as the complexity of the underlying models increases.

Finally, the results compiled in Table 1 show that the UMDA clearly outperforms both benchmarks. Moreover, the UMDA also appears advantageous within the framework of the BSEM algorithm, as the BSEM+UMDA behaves more effectively than the BSEM+HC. In favor of both the BSEM+HC and the BSEM+UMDA, it must be said that they are typically less time consuming than the UMDA, despite they always evaluate a considerably larger number of solutions than the UMDA. The reason is that every evaluation performed by the UMDA implies running the EM algorithm. Nevertheless, the benefits of the UMDA for unsupervised learning of BNs are apparent.

4 Conclusions

The main contribution of this paper has been the proposal and empirical evaluation of EDAs for unsupervised learning of BNs. Specifically, the evaluation has been limited to one of the simplest EDAs, namely, the UMDA. The evaluation has been performed on synthetic databases in order to compare the learnt models with those underlying the learning databases. The evaluation has mainly focused on assessing the capacity of the UMDA for inducing BNs for data clustering that performed well with respect to (i) the ability to summarize the learning data, (ii) the ability to generalize the learning data to previously unseen data, and (iii) the closeness to the original models. The results reported are encouraging, as they show that the UMDA can be considered a reliable and robust technique for unsupervised learning of BNs. Due to space limitations, a thorough evaluation, including real data and EDAs more sophisticated than the UMDA, is deferred to an extended version of this paper. Finally, it should be noted that the work developed in this paper can be readily applied to BN induction from incomplete data in general. The validation of this approach may be an issue of further research.

References

R. Blanco, I. Inza, and P. Larrañaga. 2002. Learning Bayesian Networks in the Space of Structures by Estimation of Distribution Algorithms. *International Journal of Intelligent Systems*, accepted.

D. M. Chickering. 1996a. Learning Bayesian Networks is NP-Complete. In *Learning from Data: Artificial Intelligence and Statistics V*, pages 121–130. Springer-Verlag.

D. M. Chickering. 1996b. Learning Equivalence Classes of Bayesian Network Structures. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 150–157. Morgan Kaufmann Publishers.

N. Friedman. 1998. The Bayesian Structural EM Algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 129–138. Morgan Kaufmann Publishers.

P. Larrañaga and J. A. Lozano (eds.). 2001. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation.* Kluwer Academic Publishers.

H. Mühlenbein. 1997. The Equation for Response to Selection and Its Use for Prediction. *Evolutionary Computation*, 5(3):303–346.

J. M. Peña, J. A. Lozano, and P. Larrañaga. 1999. Learning Bayesian Networks for Clustering by Means of Constructive Induction. *Pattern Recognition Letters*, 20(11–13):1219–1230.

J. M. Peña, J. A. Lozano, and P. Larrañaga. 2002. Learning Recursive Bayesian Multinets for Data Clustering by Means of Constructive Induction. *Machine Learning*, 47(1):63–89.