Learning Bayesian Networks in the Space of Structures by Estimation of Distribution Algorithms

Rosa Blanco, Iñaki Inza, Pedro Larrañaga
Intelligent System Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, P.O. Box 649, 20080 Donostia—San Sebastián, Spain

The induction of the optimal Bayesian network structure is NP-hard, justifying the use of search heuristics. Two novel population-based stochastic search approaches, univariate marginal dis-tribution algorithm (UMDA) and population-based incremental learning (PBIL), are used to learn a Bayesian network structure from a database of cases in a score search framework. A comparison with a genetic algorithm (GA) approach is performed using three different scores: penalized maximum likelihood, marginal likelihood, and information-theory–based entropy. Experimental results show the interesting capabilities of both novel approaches with respect to the score value and the number of generations needed to converge.

1. INTRODUCTION

There has been a big growth in the use of the probability theory during the last 10 years as a formalism to reason under uncertainty in artificial intelligence. This resurgence of the probability theory in artificial intelligence has been principally motivated by Ref. 1, in which an algorithm for the evidence propagation in probabilistic graphical models is introduced.

Probabilistic graphical models are able to represent n-dimensional probability distributions by means of a directed acyclic graph and a set of marginal and conditional probability distributions drawn from the graph structure. This graph gathers a semantic related to the conditional independence concept.[2]

Among developed probabilistic graphical models, Bayesian networks are the most studied paradigm, resulting in a large number of applications. As each $X_1, \ldots, X_n$ random variable follows multinomial probability distributions in a Bayesian network, the joint probability distribution $p(x_1, \ldots, x_n)$ can be factorized by the formula $p(x_1, \ldots, x_n) = \Pi_{i=1}^{n} p(x_i|\mathbf{pa}(x_i))$, where x_i represents the value of the random variable X_i, and $\mathbf{pa}(x_i)$ represents a combination of the values of the random variable parents of X_i in the graphical structure. Excellent introductions to the Bayesian network paradigm can be found in Refs. 3, 4, 5, and 6.

The causality relations among the problem variables can be represented by a directed acyclic graph in several application fields (i.e., genetic domains). On the other hand, the help of an expert that constructs a list of conditional (in)dependences among the problem variables is needed in other domains. However, an automatic learning process that induces the Bayesian network structure from a database of cases is an interesting alternative. This automatic process should reflect the conditional (in)dependences that implicitly appear in the database. Although the first automatic approaches tried to produce a list of conditional (in)dependences by the use of statistical tests,[7] another automatic approach has strongly emerged in the last years: the score + search approach. The score + search approach is based on the idea of performing an intelligent search in a specific space (the space of network structures, orders, skeletons, or equivalence classes) and evaluating each proposed Bayesian network.

In this work, continuing within the score + search approach, an empirical comparison between three population-based, stochastic search paradigms is performed: genetic algorithms (GAs), univariate marginal distribution algorithms (UMDA), and population-based incremental learning (PBIL). Although GAs have been used in previous works to search for the optimal Bayesian network, as far as we know, this is the first work that uses UMDA and PBIL search strategies in the exposed task.

The rest of the work is organized as follows. In Section 2 the principal score + search contributions in the Bayesian network induction task are reviewed, ordering them according to the employed search strategy. A special emphasis is put on three score metrics that are used in the experimental part; Bayesian information criterion (BIC), K2, and entropy. Section 3 presents the search heuristics used in this work: GAs, UMDA, and PBIL. Section 4 shows the individual representations, and Section 5 collects the experimental results of three approaches over three networks previously used in the literature. We finish with a brief set of conclusions.

2. SCORE + SEARCH APPROACHES TO LEARNING BAYESIAN NETWORKS

In this section, the principal works that use a score + search mechanism for the induction of multiply connected Bayesian networks are reviewed. Although the principal score metrics are introduced, our review is focused in the way the search is performed and the nature of the space where this search is performed. Detailed

revisions on structure learning of Bayesian networks from data can be found in Refs. 8, 9, and 10.

2.1. Families of Scores

$\mathbf{X} = (X_1, \ldots, X_n)$ denotes an n-dimensional random variable and $\mathbf{x} = (x_1, \ldots, x_n)$ represents one of its possible instances. If the variable X_i has r_i possible values $x_i^1, \ldots, x_i^{r_i}$, the local distribution, $p(x_i|\mathbf{pa}_i^{j,S}, \boldsymbol{\theta}_i)$ is an unrestricted discrete distribution $p(x_i^k|\mathbf{pa}_i^{j,S}, \boldsymbol{\theta}_i) = \theta_{x_i^k|\mathrm{pa}_i^j} \equiv \theta_{ijk}$, where $\mathbf{pa}_i^{1,S}, \ldots, \mathbf{pa}_i^{q_i},S$ denotes the values of $\mathbf{Pa}_i^S$, the set of parents of the variable X_i in the structure S. The term q_i denotes the number of possible different instances of the parent variables of X_i. Thus, $q_i = \Pi_{X_g \in \mathbf{Pa}_i} r_g$. The local parameters are given by $\boldsymbol{\theta}_i = ((\theta_{ijk})_{k=1}^{r_i})_{j=1}^{q_i})$. In other words, the parameter θ_{ijk} represents the conditional probability of variable X_i being in its kth value, knowing that the set of its parent variables is in its jth value. We represent by $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ a database with N cases. The information contained in D is used to learn the Bayesian network structure S.

Using the maximum likelihood estimate for θ_{ijk} ($\hat{\theta}_{ijk} = N_{ijk}/N_{ij}$ where N_{ijk} denotes the number of cases in D in which the variable X_i has the value x_i^k and $\mathbf{Pa}_i$ has its jth value and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$), and incorporating some form of penalty model complexity into the maximized likelihood, we obtain a general formula for the *penalized maximum likelihood* score as

$$\sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - f(N) \sum_{i=1}^{n} q_i(r_i - 1)$$

where $f(N)$ is a nonnegative penalization function. A usual choice for it is the Jeffreys-Schwarz criterion, sometimes called the BIC,[11] where $f(N) = \frac{1}{2} \log N$.

In the Bayesian approach to the Bayesian network model induction from data, we express our uncertainty on the model (structure and parameters) by defining a variable in which its states correspond to the possible network structure hypothesis S^h and assessing the probability $p(S^h)$. In the approach known as Bayesian model selection we select the model in which its logarithm of the relative posterior probability, i.e., $\log p(S, D)$, is maximum. Taking into account that $\log p(S|D) \propto \log p(S, D) = \log p(S) + \log p(D|S)$ and under the assumption that the prior distribution over the structure is uniform, an equivalent criterion is the log *marginal likelihood* ($\log p(D|S)$) of the data given the structure. It is possible to compute the marginal likelihood efficiently and in a closed form under some general assumptions.[12,13] For instance, it is shown[12] that if the cases occur independently (there are no missing values) and the density of the parameters given the structure is uniform, then

$$p(D|S) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$$

This score is known as the K2 metric.

The literature includes several scores that, inspired in the information theory,[14] are able to calculate the *entropy* of a probability distribution represented by a Bayesian network. It is shown that the entropy of the distribution represented by a Bayesian network structure S is[15]

$$H_S = \sum_{i=1}^{n} \sum_{j=1}^{q_i} p(\mathbf{Pa}_i = j) H_{X_i|\mathbf{Pa}_i=j}$$

where $H_{X_i|\mathbf{Pa}_{i=j}} = \sum_{k=1}^{r_i} p(X_i = x_i^k|\mathbf{Pa}_i = j) \ln p(X_i = x_i^k|\mathbf{Pa}_i = j)$.

2.2. Search Heuristics

It is shown that the search problem of identifying an optimal Bayesian network structure is NP-hard.[16] A problem P is NP-hard when some problem N in NP can be reduced to P(N $\leq$ P). This result has been used to justify the use of large number of heuristics for the exposed problem. An organization of the works that propose different search heuristics to search for near-optimal Bayesian network models can be the following: *deterministic heuristics,* for instance hill climbing,[12,17] iterated hill climbing,[18] tabu search,[19] variable neighborhood search,[20] and branch and bound;[21] or *stochastic heuristics,* e.g., simulated annealing,[18] variable neighborhood search,[20] GAs,[22–24] evolutionary programming,[25] Markov Chain Monte Carlo,[26,27] and ant colonies.[28]

2.3. The Search Space

The most usual approach to perform the search of the Bayesian network model is to perform this search in the *space of directed acyclic graphs.* The search process in this space has difficulties when the search strategy has an evolutionary nature (See Section 3.4 for more details). The number of possible structures for a domain with n variables is given by a recursive formula presented in Ref. 29.

The Bayesian Dirichlet equivalent (BDe) metric[10] asseses with the same value two Bayesian networks reflecting the same set of conditional independences. In this way, the search also can be performed in the *space of equivalence classes* (classes that reflect the same set of conditional independences).[30] Recent works[31] note the relationship between the cardinality of Bayesian network structures and equivalence of classes spaces; this can be interpreted as a deceleration in the popularization of this promising approach. There are two basic reasons for this stop: the cardinality of this space is not largely reduced and the search process in this space has a large computational cost.

The literature also proposes to perform the search in the *space of skeletons.*[32] The advantage of this space for heuristics coming from evolutionary computation is that the operations that are performed with the old population to create a new one

```
AGA
   Make initial population at random
   while not stop do
         Select parents from the population
         Produce children from the selected parents
         Mutate the individuals
         Extend the population by adding the children to it
         Reduce the extended population
   end while
   Output the best individual found
```

Figure 1. The pseudocode of the AGA.

are closed: valid individuals are generated in the offspring without the need of repair operators.

Other authors[22,26,33,34] have proposed to perform the search in the *space of orders* of the n variables of the problem. The motivation for the birth of this approach is that several structure learning algorithms need the n variables ordered.

3. FROM GAs TO UMDA AND PBIL

3.1. GAs

Roughly, a GA works as follows. First, the initial population is chosen, and the quality of each of its individuals is determined. Next, in every iteration, parents are selected from the population. These parents produce children who are added to the population. For all newly created individuals a probability near zero exists that they mutate, i.e., they change their heriditary distinctions. After that, some individuals are removed from the population according to a selection criterion in order to reduce the population to its initial size. One iteration of the algorithm is referred to as a generation. The pseudocode of an abstract GA (AGA) is shown in Figure 1.

The operators that define the child production process and the mutation process are called the crossover operator and the mutation operator, respectively. Both operators are applied with different probabilities called the crossover probability and the mutation probability. Mutation and crossover play different roles in the GAs. Mutation is needed to explore new states and it helps the algorithm to avoid local optima. Crossover should increase the average quality of the population. In this work an elitist GA is used with a one-point crossover. This operator divides the parents into two parts and it combines the parts to the generated two new individuals. The probability of crossover is set to 1.0 and the mutation probability to 0.1 (these values are so common in the literature).

The major part of genetic combinatorial approaches has no mechanism for capturing the relationships among the variables of the problem. GAs try to capture implicitly these relationships by a semiblind process, concentrating samples on

EDA
$D_0 \leftarrow$ Generate M individuals (the initial population) randomly
repeat for $l = 1, 2, \ldots$ **until** a stop criterion is met
$D^s_{l-1} \leftarrow$ Select $N \leq M$ individuals from D_{l-1} according to a selection method
$p_l(\boldsymbol{x}) = p(\boldsymbol{x}|D^s_{l-1}) \leftarrow$ Estimate the joint probability of selected individuals
$D_l \leftarrow$ Sample M individuals (the new population) from $p_l(\boldsymbol{x})$

Figure 2. Main scheme of the EDA approach.

combinations of high-performance members of the current population through the use of the recombination (crossover and mutation) operators.

In GAs no explicit information is kept about which groups of variables jointly contribute to the quality of candidate solutions. As crossover and mutation operations are randomized, they could disrupt many of these desired relationships among the variables.[35] Although the search process could produce an individual that covers an optimal relation among a subset of variables, a crossover or mutation operator could break this. Therefore, most of the crossover and mutation operations yield unproductive results and the discovery of the global optima could be delayed. On the other hand, GAs are also criticized in the literature for three aspects[36]: (i) the large number of parameters and their associated refered optimal selection or tuning process; (ii) the extremely difficult prediction of the movements of the populations in the search space; (iii) their incapacity to solve the well-known deceptive problems.

3.2. UMDA

A different way to perform a population-based, stochastic search is to change the basic principle of recombination. One idea is to estimate the joint distribution of promising solutions and use this estimate in order to generate new individuals. A general scheme of the algorithms based on this principle is called the estimation of distribution algorithms (EDAs).[37,38] In EDAs (See Figure 2), there are no crossover or mutation operators, and the new population is sampled from a probability distribution that is estimated from the selected individuals. The initial M individuals are generated at random. These individuals constitute the initial population D_0, and each of them is evaluated. In a second step, a number N ($N \leqslant M$) of individuals is selected. In a third step the induction of the n-dimensional probabilistic model that reflects the relationships among the variables is carried out. In the fourth step, M new individuals, which form the new population, are obtained from simulation of the probabilistic distribution learned in the previous step. The previous three steps are repeated until a stopping criterion is met.

The main problem with EDAs is how the probability distribution $p_l(\mathbf{x})$ is estimated. Obviously, the computation of all the parameters needed to specify the probability distribution is impractical. This has led to several approximations where the probability distribution is assumed to factorize according to a probability model.

PBIL
Obtain an initial probability vector $p_0(\boldsymbol{x})$
while no convergence **do**
Using $p_l(\boldsymbol{x})$ obtain M individuals: $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$
Evaluate and rank $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$
Select the N ($N \leq M$) best individuals: $\boldsymbol{x}_{1:M}^l, \ldots, \boldsymbol{x}_{k:M}^l, \ldots, \boldsymbol{x}_{N:M}^l$
Update the probability vector $p_{l+1}(\boldsymbol{x}) = (p_{l+1}(x_1), \ldots, p_{l+1}(x_n))$
for $i = 1, \ldots n$ **do**
$p_{l+1}(x_i) = (1 - \alpha)p_l(x_i) + \alpha \frac{1}{N} \sum_{k=1}^{N} x_{i,k:M}^l$
end while

Figure 3. Pseudocode for the main PBIL algorithm.

In the case that the n-dimensional joint probability distribution factorizes as a product of n univariate and independent probability distributions, i.e., $p_l(\mathbf{x}) = \Pi_{i=1}^{n} p_l(x_i)$, we obtain the UMDA.[39] In this work, UMDA is used.

3.3. PBIL Algorithm

PBIL[40] is another paradigm that performs a population-based, stochastic search. Its objective is to obtain the optimum of a function defined in the binary space $\Omega = \{0, 1\}^n$ (the next explanations can be easily extended to nonbinary search spaces). In each generation, the population of individuals is represented by a vector of probabilities: $p_l(\mathbf{x}) = (p_l(x_1), \ldots, p_l(x_i), \ldots, p_l(x_n))$, where $p_l(x_i)$ refers to the probability of obtaining a value of 1 in the ith component of D_l, the population of individuals in the lth generation. The algorithm works as follows (See Figure 3). At each generation, using the probability vector $p_l(\mathbf{x})$, M individuals are obtained. Each of these M individuals are evaluated and the N best of them ($N \leq M$) are selected. We denote them by $\mathbf{x}_{1:M}^l, \ldots, \mathbf{x}_{i:M}^l, \ldots, \mathbf{x}_{N:M}^l$. These selected individuals are used to update the probability vector by using a Hebbian inspired rule: $p_{l+1}(\mathbf{x}) = (1 - \alpha)p_l(\mathbf{x}) + \alpha(1/N) \sum_{k=1}^{N} \mathbf{x}_{k:M}^l$, where $\alpha \in (0, 1]$ is a parameter of the algorithm. Note that the PBIL algorithm only belongs to the EDA approach in the case that $\alpha = 1$. In this case, PBIL coincides with UMDA. In our implementation of PBIL, α is fixed to 0.5. A theoretical study of PBIL can be consulted in Ref. 41.

4. INDIVIDUAL REPRESENTATION

To represent a Bayesian network structure, the same representative schema is used for three evaluation approaches (GAs, UMDA, and PBIL). In an n-dimensional domain, each Bayesian network structure is represented by a connectivity matrix $\mathbf{C} \in M(n, n)$, in which its elements c_{ij} verify that

$$c_{ij} = \begin{cases} 1 & \text{if } X_i \text{ is parent of } X_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

From this connectivity matrix two different individual representations can be proposed: (i) if an order of the variables is given, a node only can be parent of its following variables within the proposed ordering. The values of the connectivity matrix below the diagonal are zero. The array required to represent a network structure is given by the values of the upper triangular connectivity matrix

$$\mathbf{I} = (c_{12}, \ldots, c_{1n}, c_{23}, \ldots, c_{2n}, \ldots, c_{i(i+1)}, \ldots, c_{in}, \ldots, c_{(n-1)n})$$

(ii) if all the nodes of the network can be parents of the rest of the nodes, only the values of the c_{ii} elements of the connectivity matrix are zero. An $n^2 - n$-dimensional array is required to represent a network structure

$$\mathbf{I} = (c_{12}, \ldots, c_{1n}, \ldots, c_{i1}, \ldots, c_{i(i-1)}, c_{i(i+1)}, \ldots, c_{in}, \ldots, c_{n1}, \ldots, c_{n(n-1)})$$

It must be taken into account that the previous arrays represent a directed acyclic graph. Thus, neither genetic crossover and mutation operators nor the simulation of new individuals in UMDA and PBIL are closed operations with respect to the acyclicity when the ordering is not available; in the genetic recombination and in the simulation phase of new individuals of UMDA and PBIL, "not valid" individuals could be generated. In this way, we are forced to use a repairing operator to transform not valid individuals (solutions with cycles) into valid ones (directed acyclic graphs). In this work, a simple repairing operation is used: once a cycle is detected in the individual, one arc of the cycle is randomly deleted (this is repeated until a directed acyclic graph is achieved).

5. EXPERIMENTAL RESULTS

To compare the behavior of the three proposed search algorithms (GAs, UMDA, and PBIL), they are tested, in a score + search framework, for three scores (BIC, K2, and entropy) introduced in Section 2.1. We test our three algorithms over three databases of 10,000 cases generated from the *Asia,*[1] the *Alarm,*[42] and the *Water*[43] Bayesian networks. Alarm database is a subset of the 20,000 cases generated by probabilistic logic sampling,[44] and Asia and Water databases are generated using Hugin Expert software.

Three search techniques are tested with the same population size $10n$, where n is the number of variables of the problem (n is 8, 37, and 32 for Asia, Alarm, and Water networks, respectively). The presented algorithms in the previous sections are general schemes that can be modified. In this work an elitist scheme is used for three search strategies: the new population is formed from the best members of both the previous population and the offspring.

In the case of UMDA and PBID, half of the best individuals of the populations are selected to form the pool of "best individuals." In the case of GA, a rank-based proportional selection is used to select individuals for crossover. Ten independent runs are executed for each combination of score and search technique. When the ordering is taken into account, this ordering is consistent with the topology of the network and it is the same for the 10 independent runs.

Table 1. Results of the best scores and the number of generations required for convergence of Asia network.

	BIC		K2		Entropy	
Alg.	Score ± SD	Gener. ± SD	Score ± SD	Gener. ± SD	Score ± SD	Gener. ± SD
Order						
GA	−10947.1 ± 13.6†	21.1 ± 14.35†	−11086.4 ± 387.47†	25.6 ± 16.75†	1.09 ± 0†	26.1 ± 13.89†
UMDA	−9890.05 ± 2.51	13.3 ± 1.06	−9802.66 ± 0	15.0 ± 1.76	−1.00 ± 0	14.3 ± 1.34
PBIL	−9889.26 ± 0	18.5 ± 1.58†	−9802.66 ± 0	19.9 ± 1.28†	−1.00 ± 0	19.7 ± 1.57†
K2	−9889.26 ± —	— ± —	−9802.66 ± —	— ± —	−1.00 ± —	— ± —
No order						
GA	−9969.03 ± 46.67†	18.7 ± 3.40	−9813.09 ± 11.41	27.9 ± 9.07	−0.97 ± 0	25.7 ± 6.04
UMDA	−9917.61 ± 15.29	29.7 ± 6.20†	−9684.36 ± 343.36	34.7 ± 6.68	−0.97 ± 0	24.3 ± 3.13
PBIL	−9917.46 ± 15.06	30.7 ± 3.89†	−9793.72 ± 39.20	39.5 ± 6.62	−0.97 ± 0	26.8 ± 1.75
K2	−9968.66 ± 35.57	— ± —	−9804.25 ± 21.28	— ± —	−1.00 ± 0.19	— ± —

The real values of the BIC, K2, and entropy scores for the network are −9894.16, −9802.66, and −1.00, respectively.

With the purpose of comparing the obtained results with a "standard algorithm" to learn Bayesian networks, the results obtained with the well-known K2 algorithm[12] are shown. The K2 algorithm is only executed once when the order of the variables is supplied and $10n$ with random orders when the order is not available.

Tables 1, 2, and 3 show the obtained results for Asia, Alarm, and Water problems, respectively. For each combination of score + search technique, the average score and number of required generations for convergence are shown in tables. It is assumed that the search converges when the sum of the scores of the individuals of the previous population is the same as the sum of the scores of the current population. It must be noted that the maximization of the three scores is the objective.

A deeper analysis of the results is performed by means of statistical tests. The Mann-Whitney test is performed to determine the significance of the differences

Table 2. Results of the best scores and the number of generations required for convergence of the Alarm network.

	BIC		K2		Entropy	
Alg.	Score ± SD	Gener. ± SD	Score ± SD	Gener. ± SD	Score ± SD	Gener. ± SD
Order						
GA	−68525.0 ± 1009.9†	106.5 ± 27.24†	−79494.7 ± 1788.1†	155.1 ± 25.68†	−8.59 ± 0.03†	165.8 ± 34.65†
UMDA	−49430.4 ± 116.40	79.6 ± 5.43†	−47083.1 ± 15.41	73.3 ± 1.77	−6.52 ± 0.02	77.4 ± 3.31†
PBIL	−49466.2 ± 28.87	66.2 ± 1.69	−47081.6 ± 8.81	77.0 ± 1.94†	−6.52 ± 0.02	74.6 ± 1.07
K2	−49433.5 ± —	— ± —	−47103.5 ± —	— ± —	−6.53 ± —	— ± —
No order						
GA	−52331.8 ± 574.21†	174.7 ± 25.84†	−47744.4 ± 289.54†	272.2 ± 31.05†	−6.11 ± 0.08	208.9 ± 36.69†
UMDA	−51224.3 ± 129.47	149.2 ± 38.02†	−47083.3 ± 10.43	183.8 ± 26.17†	−6.12 ± 0.07	202.5 ± 44.42†
PBIL	−51250.8 ± 362.90	101.1 ± 7.82	−48271.0 ± 299.81†	117.7 ± 8.06	−6.13 ± 0.04	98.6 ± 5.89
K2	−52439.7 ± 883.70	— ± —	−49193.4 ± 462.18	— ± —	−6.68 ± 0.15	— ± —

The real values of the BIC, K2, and entropy scores for the network are −49687.55, −47086.57, and −6.52, respectively.

Table 3. Results of the best scores and the number of generations required for convergence of Water network.

	BIC		K2		Entropy	
Alg.	Score ± SD	Gener. ± SD	Score ± SD	Gener. ± SD	Score ± SD	Gener. ± SD
Order						
GA	−62429.3 ± 586.75†	77.6 ± 26.24	−71175.4 ± 800.08†	124.3 ± 27.40†	−10.34 ± 0.01†	45.44 ± 39.71†
UMDA	−57119.1 ± 3.11	167.3 ± 10.02†	−56251.4 ± 0	44.6 ± 1.35	−9.79 ± 0	36.2 ± 2.62
PBIL	−63030.1 ± 1705.3†	178.9 ± 6.71†	−56257.8 ± 8.20	53.9 ± 1.79†	−9.79 ± 0	45.9 ± 1.91†
K2	−57118.1 ± —	— ± —	−56251.4 ± —	— ± —	−9.79 ± —	— ± —
No order						
GA	−57967.5 ± 554.27†	91.0 ± 16.92†	−56423.3 ± 100.27	127.0 ± 28.34†	−8.89 ± 0.04	121.9 ± 23.90†
UMDA	−57141.2 ± 384.66	111.8 ± 21.95†	−56346.9 ± 110.15	155.0 ± 33.86†	−8.87 ± 0.03	112.5 ± 29.26
PBIL	−57131.9 ± 69.69	65.8 ± 6.37	−56370.4 ± 56.75	72.2 ± 2.86	−8.88 ± 0.02	72.2 ± 4.64
K2	−57577.7 ± 202.02	— ± —	−56760.0 ± 86.39	— ± —	−9.45 ± 0.19	— ± —

The real values of the BIC, K2, and entropy scores for the network are −120595.94, −56687.60, and −10.07, respectively.

shown in the score and in the number of generations. For each score, statistically significant differences with respect to the algorithm with the best score are noted in the table; the same test is performed relative to the algorithm with the lowest number of needed generations for convergence. The symbol † denotes a statistically significant difference with respect to the best search algorithm at the 0.05 confidence level in Tables 1, 2, and 3.

For Asia, Alarm, and Water networks, when the ordering is supplied, UMDA and PBIL algorithm obtain competitive results with respect to GA with a lowest number of generations. The results of UMDA and PBIL improve the real values of the networks and the value of the network learned by the K2 algorithm, except for Water with the BIC score. The number of generations required for convergence by PBIL and UMDA is, in all cases, lower than the number of generations required by GA, except for Water with the BIC score.

When the ordering is not taken into account, it must be noted that the results of the GA are competitive but UMDA always obtains the best results, except for the Alarm with the entropy score. These results improve those obtained by the K2 algorithm with random orders. PBIL needs the lowest number of generations in all cases, obtaining score values not significantly different to those obtained by UMDA.

It must be noted that in all cases, for the three metrics and the three networks, GA obtains better results if the ordering is ignored, i.e., better results when the problem is more complex than if the ordering is taken into account. This can mean that the stopping criterion is restrictive to GA, and the algorithm stops when the population is not uniform. Figure 7 shows that when the improvements of the UMDA and PBIL become stable, the improvement of GA is still growing slowly. GA could possibly obtain better results with other less restrictive stopping criterion when the ordering is available.

Comparisons between the structure of the networks with the best score values and the original network are also performed. Three types of differences are measured with respect to the original network: the Hamming distance, the number

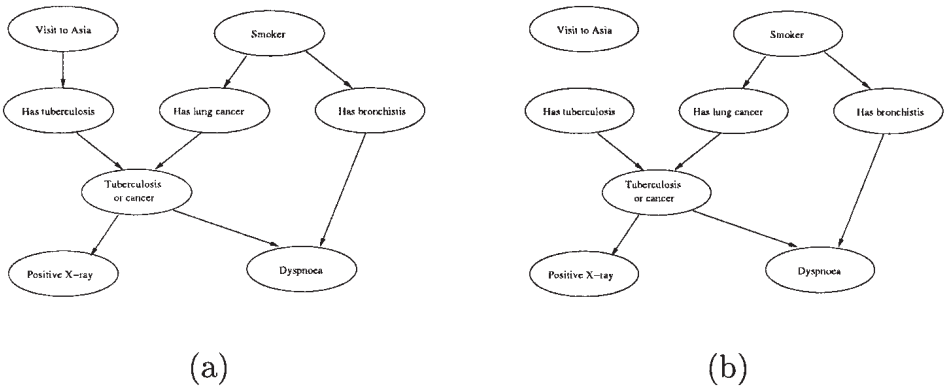


Figure 4. (a) Real Asia network with eight nodes and eight arcs. (b) Learned Asia network with only seven arcs. It can been seen that only the arc from the node visit to asia to has tuberculosis is missing.

of exceed arcs, and the number of missing arcs in the learned network. The more similar networks with respect to the original one are obtained when the ordering is taken into account. In Figures 4, 5, and 6, the closest networks to the original one are shown. It must be noted that the PBIL algorithm and UMDA with the K2 score obtain the original Asia network; in Figure 4 the network structure drawn is the structure obtained by PBIL and UMDA with BIC and entropy scores. In the case of Alarm network, the structure depicted in Figure 5 is obtained with the three metrics by the UMDA. In the case of the Water network, the structure shown in Figure 6 is the most common structure into the set of structures obtained by the three algorithms, and it is obtained with the UMDA using the metric K2. If the ordering is ignored, the learned structures are different in a large degree with respect to the original network.

Figures 7 and 8 show the evolution of the best values found with respect to the number of evaluations in the search process in a typical run for the Alarm network. In Figure 7 the ordering is available, and in Figure 8 it is ignored.

Figure 7 shows how UMDA and PBIL obtain a considerable improvement in the first 10,000 evaluations and in further evaluations this gain is maintained. We can assume that the best values found by UMDA and PBIL increase logarithmi-

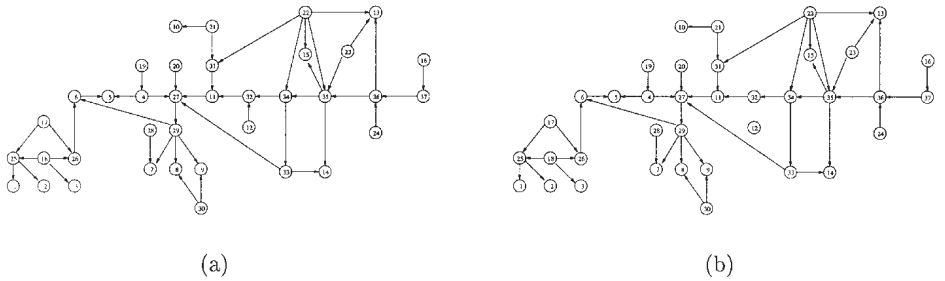
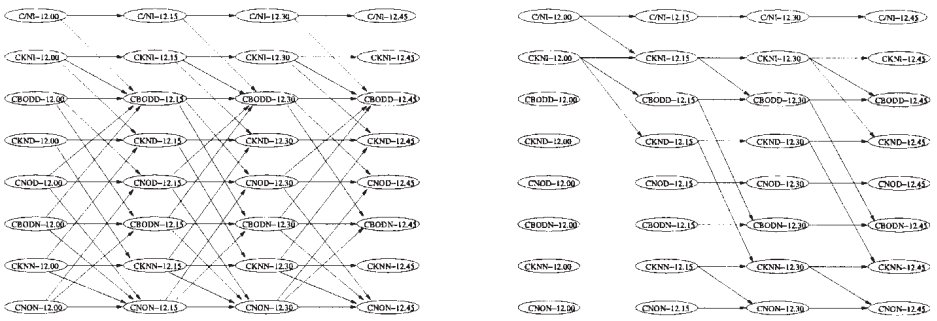


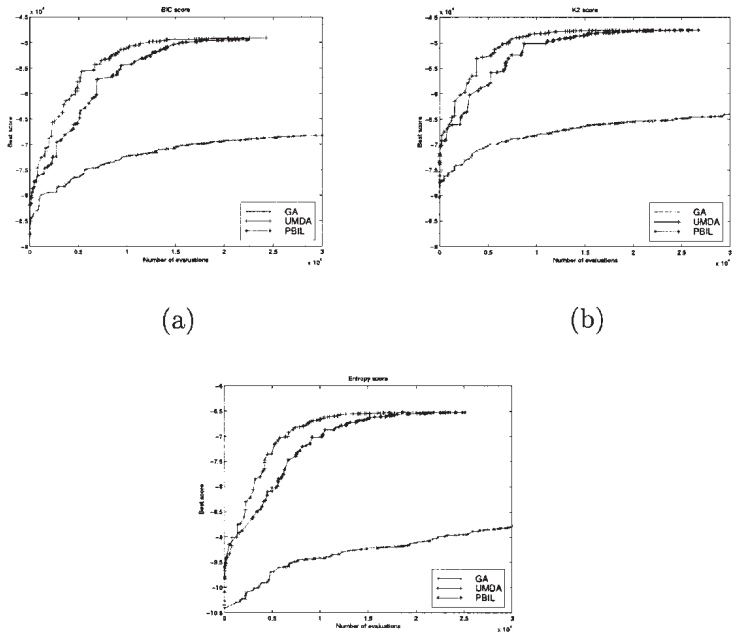
Figure 5. (a) Real Alarm network with 37 nodes and 46 arcs. (b) Learned Alarm network with 45 arcs. It can been seen that the arc from node 12 to 32 is missing.



(a) (b)

Figure 6. (a) Real Water network with 32 nodes and 66 arcs. (b) Learned Water network with 36 missing arcs.

cally. GA seems to increase logarithmically as well, but the growth is small and slow with respect to UMDA and PBIL. The number of generations required by GA is higher than those needed by UMDA and PBIL.



(a) (b)

(c)

Figure 7. Evolution of the best value found in the search process for the Alarm network when the ordering is available with (a) BIC score, (b) K2 score, and (c) entropy score.

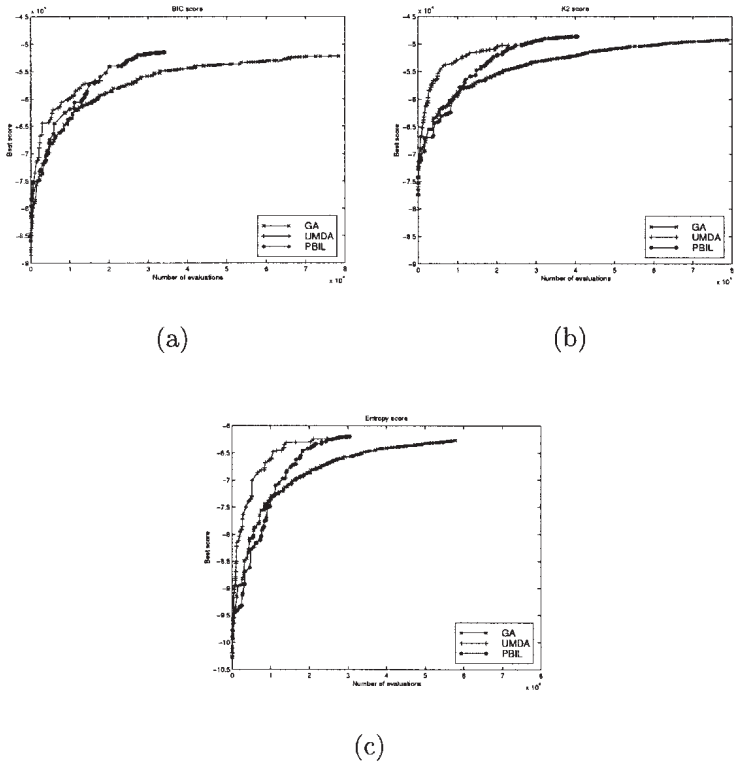


Figure 8. Evolution of the best value found in the search process for the Alarm network when the ordering is ignored with (a) BIC score, (b) K2 score, and (c) entropy score.

In Figure 8, it can be seen that the growth of the best values is very similar for the three search algorithms. It must be noted that UMDA and PBIL need less evaluations than GA. It seems that UMDA and PBIL find better values before GA in the search process, maintaining this difference in the rest of the search.

6. CONCLUSIONS

Two novel population-based, stochastic approaches UMDA and PBIL are used in the well-known problem of learning a Bayesian structure from a database of cases in a score + search framework.

In an extensive comparison with three frequently used score metrics, competitive results are achieved by these approaches with respect to a genetic approach with two different suppositions: when the ordering is known and when it is ignored.

These competitive results are better and only in two cases similar to those obtained by the K2 algorithm. The results obtained by UMDA and PBIL always improve the real values of the three proposed networks for the three scores.

The comparison of the learned structures show that if the ordering is taken into account, the obtained structures are similar to the original network and the score of the network is improved. If the ordering is not taken into account the learned structures are different in a large degree with respect to the original.

It must be noted that the experiments are performed only over three networks: Asia with a small number of nodes, and Alarm and Water, with a similar number of nodes. Thus, the obtained results must be generalized with caution.

Acknowledgments

We thank Basilio Sierra and Elena Lazkano for their useful suggestions. This work is partially supported by the University of the Basque Country; by the Department of Education, University and Research of the Basque Government; and by the Ministry of Science and Technology under Grants 9/UPV/EHU 00140.226-12084/2000, PI 1999-40, and TIC2001-2973-C05-03, respectively.

References

1. Lauritzen SL, Spiegelhalter DJ. Local computations with probabilities on graphical structures and their application on expert systems. J R Stat Soc B 1988;50(2):157–224.
2. Dawid AP. Conditional independence in statistical theory. J R Stat Soc B 1979;41:1–31.
3. Castillo E, Gutiérrez JM, Hadi AS. Expert systems and probabilistic network models. New York: Springer-Verlag; 1997.
4. Cowell RG, Dawid AP, Lauritzen SL, Spiegelhalter DJ. Probabilistic networks and expert systems. New York: Springer-Verlag; 1999.
5. Jensen FV. Bayesian networks and decision graphs. Berlin: Springer Verlag; 2001.
6. Pearl J. Probabilistic reasoning in intelligent systems. San Mateo, CA: Morgan Kaufmann; 1988.
7. Spirtes P, Glymour C, Scheines R. Causation, prediction, and search. In: Lecture Notes in Statistics 81. Berlin: Springer-Verlag; 1993.
8. Buntine W. A guide to the literature on learning probabilistic networks from data. IEEE Trans Knowl Data Eng 1996;8(2):195–210.
9. de Campos LM. Automatic learning of graphical models. I: Basic methods. In: Gámez JA, Puerta JM, editors. Probabilistic expert system. Ediciones de la Universidad de Castilla-La Mancha; 1998. pp 113–140. (In Spanish)
10. Heckerman D, Geiger D. Likelihoods and parameters priors for Bayesian networks. Technical Report MST-TR-95-54, Microsoft Advanced Technology Division, Microsoft Corporation, Seattle, WA, 1995.
11. Schwarz G. Estimating the dimension of a model. Ann Stat 1978;7(2):461–464.
12. Cooper GF, Herskovits EA. A Bayesian method for the induction of probabilistic networks from data. Machine Learning 1992;9:309–347.
13. Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 1995;20:197–243.
14. Shannon CE. A mathematical theory of communication. Bell Syst Tech J 1948;27:379–423.
15. Herskovits EA, Cooper GF. Kutató: An entropy-driven system for construction of probabilistic expert systems from database. In: Proc 6th Conf on Uncertainty in Artificial Intelligence. New York, NY: Elsevier Science; 1990. pp 54–62.

16. Chickering DM, Geiger D, Heckerman D. Learning Bayesian networks is NP-hard. Technical Report MSR-TR-94-17, Microsoft Research, Advanced Technology Division, Microsoft Corporation, Redmond, WA, 1994.
17. Buntine W. Theory refinement in Bayesian networks. In: Proc 7th Conf on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann; 1991. pp 52–60.
18. Chickering DM, Geiger D, Heckerman D. Learning Bayesian networks: Search methods and experimental results. In: Preliminary Papers of the 5th Intl Workshop on Artificial Intelligence and Statistics. Fort Lauderdale, FL: IEEE Press; 1995. pp 112–128.
19. Bouckaert RR. Bayesian belief networks: From construction to inference. PhD thesis, University of Utrecht, 1995.
20. de Campos LM, Puerta JM. Stochastic local and distributed search algorithms for learning Bayesian networks. In: Proc 3rd Symp on Adaptive Systems, Havana, Spain. 2001. pp 109–115.
21. Tian J. A branch and bound algorithm for MDL learning Bayesian networks. In: Proc 16th Conf on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann; 2000. pp 580–588.
22. Larrañaga P, Kuijpers CMH, Murga RH, Yurramendi Y. Searching for the best ordering in the structure learning of Bayesian networks. IEEE Trans Syst Man Cybern 1996;41(4): 487–493.
23. Larrañaga P, Poza M, Yurramendi Y, Murga RH, Kuijpers CMH. Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. IEEE Trans Pattern Anal Mach Intell 1996;18(9):912–926.
24. Myers JW, Laskey KB, Levitt T. Learning Bayesian networks from incomplete data with stochastic search algorithms. In: Proc 15th Conf on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann; 1999. pp 476–485.
25. Wong ML, Lam W, Leung KS. Using evolutionary computation and minimum description length principle for data mining of probabilistic knowledge. IEEE Trans Pattern Anal Mach Intell 1999;21(2):174–178.
26. Friedman J, Koller D. Being Bayesian about network structure. In: Proc 16th Conf on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann; 2000. pp 201–210.
27. Kočka T, Castello R. Improved learning of Bayesian networks. In: Proc 17th Conf on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann; 2001. pp 269–276.
28. de Campos LM, Puerta JM. Learning Bayesian network structures using ant colony optimization. 2001. In press.
29. Robinson RW. Counting unlabelled acyclic digraphs. In: Lecture Notes in Mathematics: Combinatorial Mathematics V. Berlin: Springer-Verlag; 1977. pp 28–43.
30. Chickering M. Learning equivalence classes of Bayesian networks structures. In: Proc 12th Conf on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann; 1996. pp 150–157.
31. Gillispie S, Perlman M. Enumerating Markov equivalence classes of acyclic digraph models. In: Proc 17th Conf Uncertainty in Artificial Intelligence. 2001. pp 171–177.
32. Steck H. On the use of skeletons when learning in Bayesian networks. In: Proc 16th Conf on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann; 2000. pp 558–565.
33. de Campos LM, Huete JF. Approximating causal orderings for Bayesian networks using genetic algorithms and simulated annealing. In: 8th Intl Conf on Information Processing

and Management of Uncertainty in Knowledge Based Systems IPMU2000, Madrid, Spain. 2000. pp 333–340.
34. de Campos LM, Puerta JM. Stochastic local algorithms for learning belief networks: searching in the space of the orderings, symbolic and quantitative approaches to reasoning with uncertainty. Lect Notes Artif Intell 2001;2143:228–239.
35. Inza I, Larrañaga P, Sierra B. Feature subset selection by Bayesian networks: A comparison with genetic and sequential algorithms. Intl J Approx Reason 2001;27(2):143–164.
36. Larrañaga P, Etxeberria R, Lozano JA, Peña JM. Combinatorial optimization by learning and simulation of Bayesian networks. In: Proc 16th Conf on Uncertainty in Artificial Intelligence. 2000. pp 343–352.
37. Larrañaga P, Lozano JA. Estimation of distribution algorithms. A new tool for evolutionary computation. Boston, MA: Kluwer Academic Press; 2001.
38. Müehlenbein H, Paaß G. From recombination of genes to the estimation of distributions. In: Lecture Notes in Computer Science: Parallel Solving from Nature IV, Vol 1411. Berlin: Springer Verlag; 1996. pp 178–187.
39. Müehlenbein H. The equation for response to selection and its use for prediction. Evol Comput 1998;5:303–346.
40. Baluja S. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.
41. González C, Lozano JA, Larrañaga P. Analyzing the PBIL algorithms by means of discrete dynamical systems. Complex Syst 2000;12(4):465–479.
42. Beinlinch IA, Suermondt HJ, Chavez RM, Coooper GF. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In: Proc 2nd European Conf on Artificial Intelligence in Medicine. Berlin, Germany: Springer-Verlag; 1989. pp 247–257.
43. Jensen FV, Kjaerulff U, Olesen KG, Pedersen J. An expert system for control of waste water treatment—a pilot project. Technical Report, Judex Datasystemer A/S, Aalborg, Denmark, 1989. (In Danish)
44. Henrion M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Lemmer JF, Kanal LN, editors. Uncertainty in artificial intelligence, Vol 2. Amsterdam: North-Holland; 1988. pp 149–163.