ELSEVIER

Pattern Recognition Letters 21 (2000) 779–786

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering

J.M. Peña *, J.A. Lozano, P. Larrañaga

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

Received 8 September 1999; received in revised form 27 March 2000

Abstract

The application of the Bayesian Structural EM algorithm to learn Bayesian networks (BNs) for clustering implies a search over the space of BN structures alternating between two steps: an optimization of the BN parameters (usually by means of the EM algorithm) and a structural search for model selection. In this paper, we propose to perform the optimization of the BN parameters using an alternative approach to the EM algorithm: the BC + EM method. We provide experimental results to show that our proposal results in a more effective and efficient version of the Bayesian Structural EM algorithm for learning BNs for clustering. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Clustering; Bayesian networks; EM algorithm; Bayesian Structural EM algorithm; Bound and collapse method

1. Introduction

One of the basic problems that arises in a great variety of fields, including pattern recognition, machine learning and statistics, is the so-called *data clustering problem* (Duda and Hart, 1973; Hartigan, 1975; Fisher, 1987; Kaufman and Rousseeuw, 1990; Banfield and Raftery, 1993). From the point of view adopted in this paper, the data clustering problem may be defined as the inference of a probability distribution for a database. We assume that, in addition to the observed variables, there is a *hidden* variable. This last unobserved variable would reflect the cluster membership for every case in the database. Thus, the clustering problem is also referred to as an example of learning from incomplete data due to the existence of such a hidden variable. In this paper, we focus on learning *Bayesian networks* (BNs) (Castillo et al., 1997; Jensen, 1996; Pearl, 1988) for clustering.

In the last few years, several methods for learning Bayesian networks from incomplete data have arisen (Cheeseman and Stutz, 1995; Friedman, 1997, 1998; Meilă and Heckerman, 1998; Peña et al., 1999; Thiesson et al., 1998). One of these methods is the *Bayesian Structural EM algorithm* developed by Friedman (1998). Due to its good performance this algorithm has received special attention in literature and it has motivated several variants of itself (Meilă and Jordan, 1997; Thiesson et al., 1998).

* Corresponding author. Tel.: +34-943-018-070; fax: +34-943-219-306; http: //www.sc.ehu.es/isg.
E-mail address: ccbpepaj@si.ehu.es (J.M. Peña).

PII: S 0 1 6 7 - 8 6 5 5 (0 0) 0 0 0 3 8 - 6

For learning Bayesian networks from incomplete data, the Bayesian Structural EM algorithm performs a search over the space of BN structures based on the well-known *EM algorithm* (Dempster et al., 1977; McLachlan and Krishnan, 1997). To be exact, the Bayesian Structural EM algorithm alternates between two steps: an optimization of the BN parameters, usually by means of the EM algorithm, and a structural search for model selection.

In this paper, we propose a change inside the general framework depicted by the Bayesian Structural EM algorithm: to perform the optimization of the BN parameters using the BC + EM *method* (Peña et al., 1999, 2000) instead of using the EM algorithm. We refer to this alternative approach as the Bayesian Structural BC + EM algorithm. Moreover, we provide experimental results showing the outperformance of our alternative approach over the Bayesian Structural EM algorithm in terms of effectiveness and efficiency when learning Bayesian networks for clustering.

The remainder of this paper is organized as follows. In Section 2, we describe BNs. Section 3 is dedicated to the Bayesian Structural EM algorithm. In Section 4, we introduce our alternative approach. Some experimental results comparing the performance of both algorithms for learning BNs for clustering from synthetic and real data are presented in Section 5. Finally, in Section 6 we draw conclusions.

2. Bayesian networks

First, let us introduce our notation. We follow the usual convention of denoting variables with upper-case letters and their states by the same letters in lower-case. We use a letter or letters in bold-italics upper-case to designate a set of variables and the same bold-italics lower-case letter or letters to denote an assignment of state to each variable in a given set. We use $p(\boldsymbol{x}|\boldsymbol{y})$ to denote the probability that $\boldsymbol{X} = \boldsymbol{x}$ given $\boldsymbol{Y} = \boldsymbol{y}$. We also use $p(\boldsymbol{x}|\boldsymbol{y})$ to denote the probability distribution (mass function as we restrict our discussion to the case where all the variables are discrete) for $\boldsymbol{X}$ given $\boldsymbol{Y} = \boldsymbol{y}$. Whether $p(\boldsymbol{x}|\boldsymbol{y})$ refers to a probability or a probability distribution should be clear from the context.

Given a n-dimensional variable $\boldsymbol{X} = (X_1, \ldots, X_n)$, a BN (Castillo et al., 1997; Jensen, 1996; Pearl, 1988) for $\boldsymbol{X}$ is a graphical factorization of the joint probability distribution of $\boldsymbol{X}$. A BN is defined by a directed acyclic graph $\boldsymbol{b}$ (model structure) determining the conditional independencies among the variables of $\boldsymbol{X}$ and a set of local probability distributions. When there is in $\boldsymbol{b}$ a directed arc from a variable X_j to another variable, X_i, X_j is referred to as a *parent* of X_i. We denote the set of all the parents that the variable X_i has in $\boldsymbol{b}$ as $\boldsymbol{Pa}(\boldsymbol{b})_i$. The model structure yields to a factorization of the joint probability distribution for $\boldsymbol{X}$

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|\boldsymbol{pa}(\boldsymbol{b})_i), \tag{1}$$

where $\boldsymbol{pa}(\boldsymbol{b})_i$ denotes the configuration of the parents of X_i, $\boldsymbol{Pa}(\boldsymbol{b})_i$, consistent with $\boldsymbol{x}$. The local probability distributions of the BN are those in Eq. (1). We assume that the local probability distributions depend on a finite set of parameters $\boldsymbol{\theta_b} \in \boldsymbol{\Theta_b}$. Therefore, Eq. (1) can be rewritten as

$$p(\boldsymbol{x}|\theta_{\boldsymbol{b}}) = \prod_{i=1}^{n} p(x_i|\boldsymbol{pa}(\boldsymbol{b})_i, \boldsymbol{\theta_b}). \tag{2}$$

If $\boldsymbol{b}^{\mathrm{h}}$ denotes the hypothesis that the conditional independence assertions implied by $\boldsymbol{b}$ hold in the true joint distribution of $\boldsymbol{X}$, then we obtain from Eq. (2) that

$$p(\boldsymbol{x}|\boldsymbol{\theta_b}, \boldsymbol{b}^{\mathrm{h}}) = \prod_{i=1}^{n} p(x_i|\boldsymbol{pa}(\boldsymbol{b})_i, \boldsymbol{\theta}_i, \boldsymbol{b}^{\mathrm{h}}). \tag{3}$$

In this paper, we limit our discussion to the case in which the BN is defined by multinomial distributions. That is, all the variables are finite discrete variables and the local distributions at each variable in the BN consist of a set of multinomial distributions, one for each configuration of the parents.

3. Bayesian Structural EM algorithm

Friedman (1997) introduces the *Structural EM algorithm* for searching over model structures in the presence of incomplete data. However, this algorithm is limited to use score functions that approximate the Bayesian score instead of dealing directly with Bayesian model selection. Hence, Friedman extends his previous work developing the Bayesian Structural EM BS–EM algorithm which attempts to directly optimize the Bayesian score rather than an approximation to it (Friedman, 1998).

The BS–EM algorithm is designed for learning a large class of models from incomplete data, including BNs and some variants thereof. However, throughout this paper we limit our discussion to the application of the BS–EM algorithm for learning BNs for clustering.

When dealing with a data clustering problem, we assume that we have a database of N cases, $\boldsymbol{d} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, where every case is represented by an assignment to $n-1$ of the n variables involved in the problem domain. So, we have $n \cdot N$ random variables that describe the database. Let us denote by $\boldsymbol{O}$ the set of observed variables or *predictive attributes*, that is, the $(n-1) \cdot N$ variables that have assigned one of their values. Similarly, let us denote by $\boldsymbol{H}$ the set of hidden or unobserved variables, that is, the N variables that reflect the unknown cluster membership of each case of $\boldsymbol{d}$.

As it can be seen in Fig. 1, the BS–EM algorithm follows the basic intuition of the EM algorithm: to complete the data using the best estimate of the distribution of the data so far, and then to perform a structural search using a procedure for complete data. At each iteration, the BS–EM algorithm attempts to maximize the expected Bayesian score instead of the true Bayesian score. For doing that, the BS–EM algorithm alternates between a step that finds the maximum a posteriori (MAP) parameters for the current BN structure and a step that searches over BN structures.

```
loop $l = 0, 1, \ldots$
  1. run the EM algorithm to compute the MAP $\widehat{\boldsymbol{\theta}}_{\mathbf{b}_l}$ parameters for $\mathbf{b}_l$
     given $\mathbf{o}$
  2. perform search over models, evaluating each model by
     $Score(\mathbf{b} : \mathbf{b}_l) = E[\log p(\mathbf{h}, \mathbf{o}, \mathbf{b}^h) | \mathbf{o}, \widehat{\boldsymbol{\theta}}_{\mathbf{b}_l}, \mathbf{b}_l^h]$
                        $= \sum_{\mathbf{h}} p(\mathbf{h} | \mathbf{o}, \widehat{\boldsymbol{\theta}}_{\mathbf{b}_l}, \mathbf{b}_l^h) \log p(\mathbf{h}, \mathbf{o}, \mathbf{b}^h)$
  3. let $\mathbf{b}_{l+1}$ be the model with the highest score among these encountered
     during the search
  4. if $Score(\mathbf{b}_l : \mathbf{b}_l) = Score(\mathbf{b}_{l+1} : \mathbf{b}_l)$
     then return $\mathbf{b}_l$
```

Fig. 1. A schematic of the BS–EM algorithm.

To completely specify the BS–EM algorithm, we have to decide on the structural search procedure (step 2 in Fig. 1). The usual approach is to perform a greedy hill-climbing search over BN structures considering at each point in the search all possible additions, removals and reversals of one directed arc. This structural search procedure is desirable as it exploits the decomposition properties of BNs and the factorization properties of the Bayesian score for complete data. However, any structural search procedure that exploits these referred properties can be used.

Friedman (1998) proves (i) the convergence of the BS–EM algorithm and (ii) that maximizing the expected Bayesian score at each iteration implies a maximization of the true Bayesian score.

4. Bayesian Structural BC + EM algorithm

As Friedman points in (Friedman, 1998), the computation of the MAP parameters $\widehat{\boldsymbol{\theta}}_{\boldsymbol{b}_l}$ for $\boldsymbol{b}_l$ given $\boldsymbol{o}$ (step 1 in Fig. 1) can be done efficiently using either the EM algorithm, gradient ascent or extensions of these methods. However, the EM algorithm is the only one used in (Friedman, 1997, 1998). We propose to use the BC + EM method to perform the search of the MAP parameters.

The BC + EM method is presented in (Peña et al., 1999, 2000) as an alternative approach to carry out the task of the EM algorithm when working with discrete variables (as is our present case). The key idea of the BC + EM method is to alternate between the *Bound and Collapse* (BC) *method* (Ramoni and Sebastiani, 1997, 1998) and the EM algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1997).

The BC method is a deterministic method to estimate conditional probabilities from incomplete databases. It bounds the set of possible estimates consistent with the available information by computing the minimum and the maximum estimate that would be obtained from all possible completions of the database. These bounds that determine a probability interval are then collapsed into a unique value via a convex combination of the extreme points with weights depending on the assumed pattern of missing data (see Ramoni and Sebastiani, 1998 for further details). This method presents all the advantages of a deterministic method and a dramatic gain in efficiency when compared with the EM algorithm.

The BC is described to be used in the presence of missing data, but it is not useful when there is a hidden variable as in the data clustering problem. The reason for this limitation is that the probability intervals returned by the BC method would be too huge and poorly informative as all the missing entries are concentrated in a single variable. The BC + EM method overcomes this problem by performing a partial completion of the database at each step (see Fig. 2 for a schematic of the BC + EM method).

For every case $\boldsymbol{x}_i$ in the database, $i = 1, \ldots, N$, the BC + EM method uses the current parameter values to evaluate the posterior distribution of the class variable c given $\boldsymbol{x}_i$. Then, it assigns the case $\boldsymbol{x}_i$ to the class with the highest posterior probability only if this posterior probability is greater than a threshold (which is called *fixing_probability_threshold*) that the user must determine. The case remains incomplete if there is no class with posterior probability greater than the threshold. As some of the entries of the hidden class variable have been completed during this process, we hope to have more informative probability intervals when running the BC. Then, the EM algorithm is executed to improve the parameter values that the BC have returned. The process is repeated until convergence.

```
1. for $i = 1, .., N$ do
   a. calculate the posterior probability distribution of the class variable $c$
      given $\mathbf{x}_i$, $p(c|\mathbf{x}_i, \boldsymbol{\theta}_{\mathbf{b}}, \mathbf{b}^h)$
   b. let $p_{max}$ be the maximum of $p(c|\mathbf{x}_i, \boldsymbol{\theta}_{\mathbf{b}}, \mathbf{b}^h)$ which is reached for
      $C = c_{max}$
   c. if $p_{max} > fixing\_probability\_threshold$
      then assign the case $\mathbf{x}_i$ to the class $c_{max}$
2. run the BC method
   a. bound
   b. collapse
3. set the parameter values for the current BN to be the BC's output
   parameter values
4. run the EM algorithm until convergence
5. if BC+EM convergence
   then stop
   else go to 1.
```

Fig. 2. A schematic of the BC + EM method.

Due to the fact that the BC + EM method exhibits a faster convergence rate and a more effective, efficient and robust behaviour than the EM algorithm (Peña et al., 1999, 2000), we apply it to search for the MAP parameters for the current BN structure inside the framework depicted by the BS–EM algorithm. We refer to the resulting algorithm as the Bayesian Structural BC + EM (BS–BC + EM) algorithm (Fig. 3).

To completely specify the BS–BC + EM algorithm, we have to decide on the structural search procedure. As the structural search step remains the same for the BS–EM algorithm and for the (BS–BC + EM) algorithm, what we discussed in the previous section concerning the structural search procedure for the former can also be applied to the latter.

Note that the BS–BC + EM algorithm keeps two desirable properties of the BS–EM algorithm: convergence and maximization of the true Bayesian score by means of the maximization of the expected score. The proofs of these two properties for the BS–BC + EM algorithm are the same as those presented in (Friedman, 1998) for the BS–EM algorithm. That is, those proofs are not affected by the procedure used to compute the MAP parameters for the current BN structure.

```
loop $l = 0, 1, \ldots$
   1. run the BC+EM method to compute the MAP $\widehat{\boldsymbol{\theta}}_{\mathbf{b}_l}$ parameters for $\mathbf{b}_l$
      given $\mathbf{o}$
   2. perform search over models, evaluating each model by
      $Score(\mathbf{b} : \mathbf{b}_l) = E[\log p(\mathbf{h}, \mathbf{o}, \mathbf{b}^h)|\mathbf{o}, \widehat{\boldsymbol{\theta}}_{\mathbf{b}_l}, \mathbf{b}_l^h]$
                          $= \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{o}, \widehat{\boldsymbol{\theta}}_{\mathbf{b}_l}, \mathbf{b}_l^h) \log p(\mathbf{h}, \mathbf{o}, \mathbf{b}^h)$
   3. let $\mathbf{b}_{l+1}$ be the model with the highest score among these encountered
      during the search
   4. if $Score(\mathbf{b}_l : \mathbf{b}_l) = Score(\mathbf{b}_{l+1} : \mathbf{b}_l)$
      then return $\mathbf{b}_l$
```

Fig. 3. A schematic of the BS–BC + EM algorithm.

5. Experimental results

As we are interested in solving data clustering problems of considerable size, the direct application of the BS–EM algorithm and/or the BS–BC + EM algorithm to medium or large size databases may be an unrealistic and inefficient solution. In our opinion, the two main reasons of this possible inefficiency are that (i) the computation of Score$(\boldsymbol{b} : \boldsymbol{b}_l)$ implies a huge computational expense as it takes account of every possible completion of the database and (ii) the space of BN structures is too huge to perform an efficient search.

Instead of considering every possible completion of the database in the computation of Score $(\boldsymbol{b} : \boldsymbol{b}_l)$, in this section, we evaluate and compare two relaxed versions of the presented BS–EM and BS–BC + EM algorithms that just consider the most probable completion of the database to compute Score $(\boldsymbol{b} : \boldsymbol{b}_l)$. In order to reduce the huge space of BN structures where the structural search is performed, we propose to learn a class of compromise BNs: *augmented Naive Bayes models* (Friedman and Goldszmidt, 1996; Keogh and Pazzani, 1999). These BNs are defined by the following two conditions:

- each predictive attribute has the class variable as a parent;
- predictive attributes may have one other predictive attribute as a parent.

Despite being widely accepted that augmented Naive Bayes models are a weaker representation of some domains than more general BNs, the expressive power of augmented Naive Bayes models is still recognized. Thus, these BNs are examples of an interesting balance between efficiency and effectiveness, that is, a balance between the cost of the learning process and the quality of the learnt BN (see Friedman and Goldszmidt, 1996; Keogh and Pazzani, 1999, for recent works on this topic).

To completely specify the BS–EM and the BS–BC + EM algorithms we have to decide on the structural search procedure. In our experimental comparison, both algorithms start from a Naive Bayes structure (denoted by $\boldsymbol{b}_0$) with randomly chosen parameters and perform a greedy hill-climbing search over augmented Naive Bayes model structures considering at each point in the search all possible additions, removals and reversals of one directed arc.

5.1. Performance criteria

Table 1 summarizes the criteria that we use to compare the BNs learnt by the BS–EM and the BS–BC + EM algorithms. The log marginal likelihood of the learnt BN, $\log p(\boldsymbol{d}|\boldsymbol{b}^{\mathrm{h}})$, is used in our comparison. In addition to this, we consider the runtime as valuable information. For the synthetic databases we pay attention to the cross-entropy between the true joint distribution for data and the joint distribution given by the learnt BN represented by $(\boldsymbol{b}, \boldsymbol{\theta}_{\boldsymbol{b}})$,

$$\sum_{\boldsymbol{x}} p^{\mathrm{true}}(\boldsymbol{x}) \log p(\boldsymbol{x}|\boldsymbol{\theta}_{\boldsymbol{b}}, \boldsymbol{b}^{h}). \tag{4}$$

In our experimental comparison on synthetic data, we estimate this criterion using a holdout database $\boldsymbol{d}_{\mathrm{test}}$ as

$$\frac{1}{|\boldsymbol{d}_{\mathrm{test}}|} \sum_{\boldsymbol{x} \in \boldsymbol{d}_{\mathrm{test}}} \log_2 p(\boldsymbol{x}|\boldsymbol{\theta}_{\boldsymbol{b}}, \boldsymbol{b}^{\mathrm{h}}). \tag{5}$$

For both the EM algorithm and the BC + EM method, the convergence criterion is satisfied when either the relative difference between successive values for the log marginal likelihood for the BN is less than 10^{-6} or 150 iterations are reached. For the BC + EM method, the fixing probability threshold is equal to 0.51.

Table 1
Performance criteria

Expression	Comment
$\boldsymbol{sc}_{\mathrm{initial}}$	Initial score, log marginal likelihood of the initial BN once the database has been completed with the initial model
$\boldsymbol{sc}_{\mathrm{final}} \pm S_n$	Mean ± standard deviation (over 5 runs) of the log marginal likelihood of the learnt BN
$\boldsymbol{L}_{\mathrm{test}} \pm S_n$	Mean ± standard deviation (over 5 runs) of the estimate of the cross entropy between the true joint distribution and the joint distribution given by the learnt BN
$\mathrm{time} \pm S_n$	mean ± standard deviation (over 5 runs) of the runtime (in seconds)

Notice should be taken that in all experiments we assume that the number of classes is known, thus, we do not perform a search to identify the number of classes in the database. As we have already said, we limit our present discussion to the case in which BNs are defined by multinomial distributions. That is, all the variables are finite discrete variables and the local distributions at each variable in every BN consist of a set of multinomial distributions, one for each configuration of the parents.

All the experiments are run on a Pentium 233 MHz computer.

5.2. Synthetic data

In this section, we describe our experimental results on synthetic data. The purpose of comparing the BS–EM and the BS–BC + EM algorithms on synthetic databases is to show the outperformance of the latter over the former in efficiency (measured in terms of $\boldsymbol{sc}_{\text{final}}$ and $\boldsymbol{L}_{\text{test}}$) and effectiveness (measured by *time*).

We constructed six synthetic databases as follows. There were 20 predictive binary attributes and one 4-valued hidden class variable involved. We randomly chose six different augmented Naive Bayes models. The number of directed arcs of each of these six BNs was 20, 22, 25, 30, 35 and 38, respectively. In order to get the six synthetic databases, from each of these six BNs, we sampled 8000 cases by means of the `h_domain_simulate` function provided by the software HUGIN API. [1] We denote each of these synthetic databases by $\boldsymbol{d}_i$, $i = 20, 22, 25, 30, 35, 38$ being the number of directed arcs of the BN that generated the database. From each of these six BNs, we also generated a holdout database containing 2000 cases to compute $\boldsymbol{L}_{\text{test}}$. Obviously, for the six synthetic and the six holdout databases we discarded all the entries corresponding to the class variable.

Table 2 compares the performance of the BS–EM and the BS–BC + EM algorithms applied to learn augmented Naive Bayes models as compromise BNs for clustering from the six synthetic databases. This table shows the clear dominance of the BS–BC + EM algorithm over the BS–EM algorithm for all performance criteria. The most striking difference between both algorithms is the runtime: the BS–BC + EM algorithm reaches better results than the BS–EM algorithm with a saving of runtime between 9% and 53%. Therefore, the BS–BC + EM algorithm exhibits a remarkable behaviour in these databases when compared with the BS–EM algorithm: a more effective and efficient behaviour.

5.3. Real data

Another source of data for our evaluation consists of two well-known real-world databases from the Machine Learning Repository (Merz et al., 1997): the tic-tac-toe database and the nursery database.

The tic-tac-toe database contains 958 cases, each of them representing a legal tic-tac-toe endgame board. Each case has nine 3-valued predictive attributes and there are two classes. The nursery database consists of 12,960 cases, each of them representing an application for admission in the public school system. Each case has eight predictive attributes, which have between two and five possible values. There are five classes. Obviously, for both databases we deleted all the class entries.

Due to the different number of classes and number of cases involved, the tic-tac-toe and the nursery databases appear to be good domains to compare the performance of the BS–EM and the BS–BC + EM algorithms once the class is hidden for both databases.

Table 3 compares the performance of the BS–EM and the BS–BC + EM algorithms applied to learn augmented Naive Bayes models as compromise BNs for clustering from the two real-world databases. This table reinforces what Table 2 revealed: the clear superiority of the BS–BC + EM algorithm over the BS–EM algorithm in terms of log marginal likelihood for the learnt BN and

[1] We used HUGIN API in the implementation of the algorithms. The HUGIN API (Application Program Interface) is a library that allows a program to create and manipulate BNs.

Table 2
Performance of the BS–EM and the BS–BC + EM algorithms on the six synthetic databases (averaged over five runs); desirable models are those with the highest $\boldsymbol{sc}_{\text{final}}$ and $\boldsymbol{L}_{\text{test}}$, and the lowest time

Database	Algorithm	$\boldsymbol{sc}_{\text{initial}}$	$\boldsymbol{sc}_{\text{final}} \pm S_n$	$\boldsymbol{L}_{\text{test}} \pm S_n$	time ± S_n
$\boldsymbol{d}_{20}$	BS–EM	−45 179	−40 691 ± 51	−16.736 ± 0.009	973 ± 341
	BS–BC + EM	−45 492	−40 662 ± 31	−16.731 ± 0.014	457 ± 127
$\boldsymbol{d}_{22}$	BS–EM	−46 951	−43 259 ± 107	−17.652 ± 0.029	794 ± 203
	BS–BC + EM	−46 663	−43 167 ± 1	−17.636 ± 0.000	440 ± 96
$\boldsymbol{d}_{25}$	BS–EM	−48 230	−44 587 ± 3	−18.148 ± 0.000	622 ± 79
	BS–BC + EM	−48 378	−44 586 ± 2	−18.151 ± 0.000	565 ± 93
$\boldsymbol{d}_{30}$	BS–EM	−46 731	−41 995 ± 182	−17.275 ± 0.065	1128 ± 424
	BS–BC + EM	−46 877	−41 713 ± 161	−17.174 ± 0.057	727 ± 310
$\boldsymbol{d}_{35}$	BS–EM	−48 583	−43 365 ± 118	−17.756 ± 0.049	1129 ± 286
	BS–BC + EM	−48 751	−43 173 ± 113	−17.690 ± 0.051	900 ± 109
$\boldsymbol{d}_{38}$	BS–EM	−47 712	−42 898 ± 142	−17.563 ± 0.039	1367 ± 159
	BS–BC + EM	−47 615	−42 455 ± 153	−17.473 ± 0.054	874 ± 230

Table 3
Performance of the BS–EM and the BS–BC + EM algorithms on the two real-world databases (averaged over five runs); desirable models are those with the highest $\mathbf{sc}_{\text{final}}$ and the lowest time

Database	Algorithm	$\boldsymbol{sc}_{\text{initial}}$	$\boldsymbol{sc}_{\text{final}} \pm S_n$	time ± S_n
Tic-tac-toe	BS–EM	−4145	−3949 ± 22	17 ± 5
	BS–BC + EM	−4150	−3931 ± 12	11 ± 1
Nursery	BS–EM	−56 995	−53 817 ± 206	1504 ± 505
	BS–BC + EM	−57 068	−53 397 ± 96	150 ± 55

runtime. As is reported in (Peña et al., 1999, 2000), it is for databases of considerable size, as the nursery database, when the BS–EM method shows all its advantages over the EM algorithm. This fact makes the BS–BC + EM algorithm able to reach better results than the BS–EM algorithm with up to 10 times less runtime for the nursery database.

6. Conclusions

We have proposed a change inside the general framework of the BS–EM algorithm resulting in an alternative approach for learning BNs for clustering: the BS–BC + EM algorithm. Our algorithm performs a search over the space of BN structures in the same way as the BS–EM algorithm does: alternating between an optimization of the BN parameters and a structural search for model selection. But, while the BS–EM algorithm performs the optimization of the BN parameters by means of the EM algorithm, the BS–BC + EM algorithm uses an alternative technique: the BC + EM method.

As we are interested in solving data clustering problems of considerable size, we have evaluated and compared the BS–EM and the BS–BC + EM algorithms in a realistic framework: some modifications have been proposed in order to gain in efficiency. Our experimental comparison between both algorithms have suggested the substantial gain in effectiveness and efficiency of the BS–BC + EM algorithm over the BS–EM algorithm.

Acknowledgements

The authors would like to thank the two anonymous referees for their useful comments related to this work. This work was supported by the Spanish Ministerio de Educación y Cultura under AP97 44673053 grant.

References

Banfield, J., Raftery, A., 1993. Model-based Gaussian and non-Gaussian clustering. Biometrics 49, 803–821.

Castillo, E., Gutiérrez, J.M., Hadi, A.S., 1997. Expert Systems and Probabilistic Network Models. Springer, New York.

Cheeseman, P., Stutz, J., 1995. Bayesian classification (AutoClass): Theory and results. In: Advances in Knowledge Discovery and Data Mining. AAAI Press, Menlo Park, CA, pp. 153–180.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. Ser. B 39, 1–38.

Duda, R., Hart, P., 1973. Pattern Classification and Scene Analysis. Wiley, New York.

Fisher, D., 1987. Knowledge acquisition via incremental conceptual clustering. Machine Learning 2, 139–172.

Friedman, N., Goldszmidt, M., 1996. Building classifiers using Bayesian networks. In: Proc. 13th National Conf. on Artificial Intelligence. AAAI Press, Menlo Park, CA, pp. 1277–1284.

Friedman, N., 1997. Learning belief networks in the presence of missing values and hidden variables. In: Proc. 14th Internat. Conf. on Machine Learning. Morgan Kaufmann, San Francisco, CA.

Friedman, N., 1998. The Bayesian Structural EM algorithm. In: Proc. 14th Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, pp. 129–138.

Hartigan, J.A., 1975. Clustering Algorithms. Wiley, Canada.

Jensen, F.V., 1996. An Introduction to Bayesian Networks. Springer, New York.

Kaufman, L., Rousseeuw, P., 1990. Finding Groups in Data. Wiley, New York.

Keogh, E.J., Pazzani, M.J., 1999. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In: Proc. Seventh Internat. Workshop on Artificial Intelligence and Statistics. Ft. Lauderdale, FL, pp. 225–230.

McLachlan, G.J., Krishnan, T., 1997. The EM Algorithm and Extensions. Wiley, New York.

Meilă, M., Heckerman, D., 1998. An experimental comparison of several clustering and initialization methods. In: Proc. 14th Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, pp. 386–395.

Meilă, M., Jordan, M.I., 1997. Estimating dependency structure as a hidden variable. Neural Inform. Process. Syst. 10, 584–590.

Merz, C., Murphy, P., Aha, D., 1997. UCI repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, http://www.ics.uci.edu/mlearn/MLRepository.html.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, Palo Alto, CA.

Peña, J.M., Lozano, J.A., Larrañaga, P., 1999. Learning Bayesian networks for clustering by means of constructive induction. Pattern Recognition Lett. 20 (11–13), 1219–1230.

Peña, J.M., Lozano, J.A., Larrañaga, P., 2000. Learning recursive Bayesian multinets for clustering by means of constructive induction. Machine Learning, accepted.

Ramoni, M., Sebastiani, P., 1997. Learning Bayesian networks from incomplete databases. In: Proc. 13th Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Mateo, CA.

Ramoni, M., Sebastiani, P., 1998. Parameter estimation in Bayesian networks from incomplete databases. Intelligent Data Analysis 2 (1).

Thiesson, B., Meek, C., Chickering, D.M., Heckerman, D., 1998. Learning mixtures of DAG models. In: Proc. 14th Conf. on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, pp. 504–513.