Intelligent Data Analysis 20 (2016) 933–951
DOI 10.3233/IDA-160839
IOS Press

A hierarchical clustering method: Applications to educational data

Byron Oviedo[a,*], Serafín Moral[b] and Amilkar Puris[a]
[a]*Universidad Técnica Estatal de Quevedo, Quevedo, Los Ríos, Ecuador, EC, Ecuador*
[b]*Universidad de Granada, Granada, España, España*

Abstract. The use of graphical probabilistic models in the field of education has been considered for this research. First, classical learning algorithms, as PC or K2 are reviewed. But the problem with these general learning procedures comes from the presence of a high number of variables that measure different aspects of the same concept, as it can be the case of socio-economic indicators in a population. In this case, we have that all the variables have some degree of dependence among them, without a true causal structure. So, a new procedure is presented which makes a hierarchical clustering of the data while learning a joint probability distribution. It generalizes AutoClass EM clustering allowing more complex models. Hierarchical clustering is compared in the experiments with classical learning algorithms showing a similar performance when considering the estimation of a joint probability distribution for all the variables, but with a clear advantage: the simplicity and easiness of the interpretation of the model. The method is applied to the analysis of two datasets of the educational data: socio-economic, academic achievement and drop outs at the Engineering Faculty of Quevedo State Technical University, and student evaluation of teachers from Gazi University in Ankara (Turkey).

Keywords: Bayesian networks, K2, PC, EM, hierarchical clustering, academic performance, student evaluation

1. Introduction

Bayesian networks are models to represent uncertainty involving a set of variables based on the existing conditional (in)dependence relationships [1]. They are also known in specialized literature as causal networks, causal probabilistic networks, or belief networks. A Bayesian network encodes the uncertainty associated to a set of variables trough the use of a joint probability distribution [2]. An efficient representation is obtained by exploiting the independence relationships among the variables to obtain a factorization of this distribution as a product of conditional probability distributions.

A Bayesian network has two parts: a directed acyclic graph in which each node represents a variable and independences are represented through the d-separation criterion [2], and a list of conditional probability distributions, one for each variable conditioned to its parents in the graph. In principle a Bayesian network only represents conditional independence relations, but it can also have a causal interpretation. In that case, the variable pointed to by the arc is dependent (cause-effect) on the one at its origin. Independences simplify knowledge representation (less parameters are needed) and reasoning (probability propagation).

*Corresponding author: Byron Oviedo, Universidad Técnica Estatal de Quevedo, Quevedo, Los Ríos, Ecuador, EC 120508, Ecuador. E-mail: boviedo@uteq.edu.ec.

Obtaining a Bayesian Network from data is a learning process divided in two stages: graph learning and parametric learning [3]. The first of them consists in obtaining a Bayesian network structure itself, i.e. dependence or independence relationships between the variables involved. The second stage has the purpose of estimating the necessary conditional probability distributions given the graph.

This paper investigates procedures for learning probabilistic graphical models in situations in which there are many variables which are dependent of some hidden common variables. This situation produces too complex graphs representing the relationships between observed variables. This is, for example, the case of data containing socio-economic data of students. All these data are related to education and living standards levels that are not directly measured. These type of problems are usually addressed using cluster procedures trying to estimate a Naive Bayes model with a hidden variable such as AutoClass [4], where parameters are estimated using the EM (Expectation-Maximization) algorithm [5]. In this paper, we follow the same line, but proposing a model which is able of learning a set of hidden variables with a tree structure, allowing more complex relationships between observed variables, but keeping simple models, both for learning and computation.

As an additional example, consider the case of UCI database *Turkiye Student Evaluation Data Set* [6] with the results of student evaluations of teachers from Gazi University in Ankara (Turkey). The variables are given in Table 1. It is clear that all the variables are related and that we may consider that they measure different aspects of the general degree of satisfaction with the course and the instructor. If we try to learn a Bayesian network with classical learning procedures [3] we will obtain a dense and complex graph (see Fig. 5). AutoClass tries to avoid this by considering that there there is a hidden variable that encodes this degree of satisfaction. However, AutoClass builds a simple Naive Bayes model in which there is an arc from the hidden variable to each one of the observed variables. This puts all the initial variables at the same level. But, seeing these variables, one can see that there are groups of variables that have stronger relationships among them than with other variables in other group. For example, variables Q1 and Q2 measure aspects related with the information given at the start, and variables variables Q13 and Q14 measure the preparation of the classes by the instructor. Then, in the model, there should be a stronger relation inside pairs Q1, Q2 and Q13, Q14, than between variables of different pairs. This is not possible in AutoClass and that is the reason of introducing our new hierarchical clustering procedure. It consists in making first a clustering of variables trying to find groups of variables that have strong relationships with variables of the same group. In this case, for example, variables Q1, Q2, Q3, and Q4 are in the same group. After, an AutoClass procedure is applied to each group of variables obtaining a hidden or auxiliary variable summarizing the values of variables. The process is recursively repeated with the different hidden variables introduced for each group till we have only one single variable. The final result is a tree (or set of trees) with a hierarchy of auxiliary variables that is more in accordance with the real structure of the variables (see Fig. 6 for the final tree obtained in this case in the experimental part).

A similar procedure has been proposed by Choi et al. [7] and Zhang [8]. The last one, proposes a model optimizing a BIC score similar to our procedure. Choi et al. [7] propose a complex theoretical procedure to find optimal models, but which is based on exact knowledge of pairwise joint distributions for the problem variables, which is not true in practice, and the fact that there is a true latent hierarchical structure. Procedures are given for the adaptation to the case in which we only have a sample of observations. Our procedure is similar to the one by Zhang [8] but in our case, instead of using general search algorithms, we use a fast heuristic procedure to find the graphical structure and propose heuristics to find good initial points for applying the EM optimization algorithm.

Table 1
Variables and their descriptions (Turkiye student evaluation)

Variable	Description
Instr	Instructor's identifier; values taken from 1,2,3.
Class	Course code (descriptor); values taken from 1–13.
Repeat	Number of times the student is taking this course; values taken from 0, 1, 2, 3, ...
Attendance	Code of the level of attendance; values from 0, 1, 2, 3, 4.
Difficulty	Level of difficulty of the course as perceived by the student; values taken from 1, 2, 3, 4, 5.
Q1	The semester course content, teaching method and evaluation system were provided at the start.
Q2	The course aims and objectives were clearly stated at the beginning of the period.
Q3	The course was worth the amount of credit assigned to it.
Q4	The course was taught according to the syllabus announced on the first day of class.
Q5	The class discussions, homework assignments, applications and studies were satisfactory.
Q6	The textbook and other courses resources were sufficient and up to date.
Q7	The course allowed field work, applications, laboratory, discussion and other studies.
Q8	The quizzes, assignments, projects and exams contributed to helping the learning.
Q9	I greatly enjoyed the class and was eager to actively participate during the lectures.
Q10	My initial expectations about the course were met at the end of the period or year.
Q11	The course was relevant and beneficial to my professional development.
Q12	The course helped me look at life and the world with a new perspective.
Q13	The Instructor's knowledge was relevant and up to date.
Q14	The Instructor came prepared for classes.
Q15	The Instructor taught in accordance with the announced lesson plan.
Q16	The Instructor was committed to the course and was understandable.
Q17	The Instructor arrived on time for classes.
Q18	The Instructor has a smooth and easy to follow delivery/speech.
Q19	The Instructor made effective use of class hours.
Q20	The Instructor explained the course and was eager to be helpful to students.
Q21	The Instructor demonstrated a positive approach to students.
Q22	The Instructor was open and respectful of the views of students about the course.
Q23	The Instructor encouraged participation in the course.
Q24	The Instructor gave relevant homework assignments/projects, and helped/guided students.
Q25	The Instructor responded to questions about the course inside and outside of the course.
Q26	The Instructor's evaluation system effectively measured the course objectives.
Q27	The Instructor provided solutions to exams and discussed them with students.
Q28	The Instructor treated all students in a right and objective manner.

The experimental part is based on general supervised classification problems from UCI repository [9], but in this case we have tested the models for their capability of approximating the joint probability distribution of class and attributes, measuring the log probability of the cases in test datasets. This measures the quality of the fitted probability distribution of the data and it is an indirect evaluation of the quality of the groups of variables and the values of the discovered hidden variables, which can be interpreted as the different clusters of the problem.

We have also used a dataset containing socio-economic data of 773 students enrolled in 2012–2013 academic year in the Faculty of Engineering Sciences of the Quevedo State Technical University together with desertion results for them. In this way, we look for models to find factors affecting desertion for students. The use of probabilistic graphic models in education field for accomplishing students diagnosis and for determining university student desertion causes have already been subject of other researches. Magaña et al. [10] analyze it by grouping individuals or objects into clusters according to their similarities, maximizing the homogeneity of the objects inside the conglomerates at the same time maximizing the heterogeneity between aggregates. Another case study to predict the likelihood that a student leaves the educational institution has been conducted using data mining techniques; among them we can cite [11] which is based on discovery rules and on TDIDT (top-down Induction of Decision

Trees) using the consortium SIU of Argentina academic management database (which brings together 33 universities in Argentina). This allows an interesting analysis to find the rules predicting desertion.

Additionally, we have also carried out some experiments in another educational problem: instructor evaluation by students. For that, we have considered the above mentioned UCI database *Turkiye Student Evaluation Data Set* [6]. We have compared our hierarchical clustering procedure with other learning methods in two aspects: the capacity of fitting a joint probability distribution for the variables and the simplicity and easiness to understand of the learned models.

The work is structured as follows: In Section 2, the key aspects of Bayesian networks and learning are described. In Section 3, the new clustering method is presented. An experimental analysis of the results obtained through the use of the new algorithm in comparison with other procedures is carried out in Section 4. Finally, Section 5 presents the conclusions and recommendations.

2. Bayesian networks and learning

Assume that we have a set of variables denoted $\mathbf{X} = \{X_1, X_2, .., X_n\}$ where each variable X_i takes values within a finite set Ω_i; then, we use x_i to express one of the values of $X_i, x_i \in \Omega_{x_i}$. Now if we have a set of indexes denoted as I, we will denote as $\mathbf{X}_I$ the subset of variables X_i where $i \in I$. Ω_I will represent the Cartesian product $\prod_{i \in I} \Omega_i$. Its elements are called configurations of $\mathbf{X}_I$ and will be represented as $\mathbf{x}_I$ or simply as $\mathbf{x}$ when there is no ambiguity with the index set.

`Definition of Bayesian network`: Given a set of variables $\mathbf{X}$, a Bayesian network (see Fig. 1) is a directed acyclic graph with a node for each variable X_i representing conditional independences in accordance with the d-separation criterion [2] and a conditional probability $p(X_i|\Pi_i)$ for each variable given its set of parents Π_i in the graph.

Given the independence relationships represented by a Bayesian network, a single joint probability distribution can be determined by means of the following expression:

$$P(\mathbf{X} = x) = \prod_{i \in N} p(x_i|\Pi_i), \ \forall x_i \in \Omega_i \tag{1}$$

Learning consists in estimating a Bayesian network from a set of observations for the variables in the problem $\mathcal{D}$. Automatic learning is divided into supervised learning and unsupervised learning. In supervised learning the attention focuses on the prediction of the values of a special variable called class and denoted as C, while in unsupervised learning the purpose is to find compact description of data, so that it approaches the distribution of the set of all variables and it generalizes well to new cases in which some of these variables are observed.

`Supervised learning`: Given a set of data $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1, \ldots, N\}$; where $(\mathbf{x}^n, c^n)$ is a set of observations for a set of attributes and the class, the relationship between the input $\mathbf{x}^n$ and the output c^n must be learned, so that when a new input $\mathbf{x}^*$ appears, it is possible to predict the value of $C = c^*$. Pair $(\mathbf{x}^*, c^*)$ is not in $\mathcal{D}$, but it is assumed that it is generated by the same unknown process that generated to $\mathcal{D}$.

Any Bayesian network can be used to perform a classification by simply distinguishing the variable of interest in the problem as the class variable. Then some propagation algorithm should be applied to this variable with new evidence and a decision rule for choosing a class value depending on the conditional posterior distribution: in general the value c^{pred} maximizing the posterior probability is the chosen value.

But in most of the cases, restricted types of Bayesian networks are used for classification. The most used and simple graphic probabilistic model for supervised classification is Naive Bayes (NB) (see

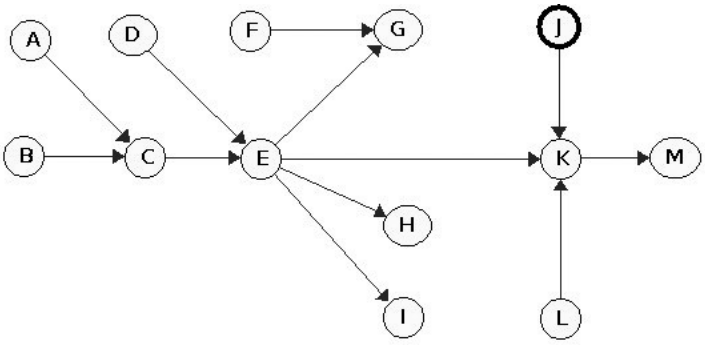

Fig. 1. Example of a Bayesian network.

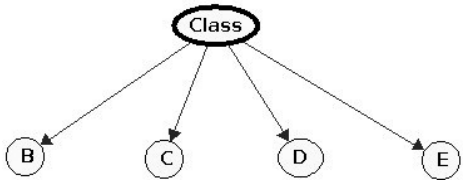

Fig. 2. Naive Bayes classifier.

Fig. 2) which is based on two assumptions: first, each attribute is conditionally independent given the attributes of the class and second, that all the attributes have influence on the class. NB has been shown to be comparable in terms of classification accuracy in many domains to many more complex algorithms, such as neural networks and decision trees [12].

`Unsupervised learning`: Given a set of data $\mathcal{D} = \{\mathbf{x}^n, n = 1, \ldots, N\}$ the goal is to find a compact description of the data. In this type of learning there is not a special class variable. The interest lies on estimating the distribution with which the set of variables $\mathbf{X}$ takes its data $P(\mathbf{X} = \mathbf{x})$. Given a new set of observations, the likelihood of the learned model given the data (the probability of the data given the model) is a popular expression of description accuracy [13].

Some techniques for learning are unsupervised clustering (objects are grouped in regions where mutual similarity is high) and dimensionality reduction (input data are grouped into subspaces of a dimension lower than the initial). The estimation of a Bayesian network can be considered as a case of unsupervised learning.

PC Algorithm [14] is one of the algorithms used for Bayesian learning. This algorithm is based on testing independences among pairs of variables X_i and X_j conditioned to a subset of variables $\mathbf{A}$. It is based on the known fact, that if the set of independences in a set of variables can be faithfully represented by a directed acyclic graph, then there is not a link connecting X_i and X_j if an only if there is a set of variables $\mathbf{A}$ such that X_i and X_j are independent given $\mathbf{A}$. The algorithm PC starts with a complete non directed graph and then it reduces it step by step. First, the edges connecting two nodes verifying a zero order conditional independence are deleted. Then, conditional independences of order one are deleted, and so on. The set of nodes likely to constitute a separation set (the set to which it is conditioned) is the set of nodes adjacent to any of the nodes intended to be separated. The PC implementation of Elvira Program[1] [15], with a 0.05 level of significance has been used in this work.

Score+search function based algorithms try to have a graph that better models the input in accordance with a specific criterion which constitutes a measure of fitting (metric or score) of a directed acyclic graph with a dataset of observations $\mathcal{D}$. In addition there is a process of exploration to find the network optimizing the metric. The most common scores are BIC [16], BDEu [17], and K2 [18]. In order to achieve efficiency the used metric should be decomposable, so that under local modifications of a graph, only the part corresponding to that part should be recomputed.

[1] Available at http://www.ia.uned.es/investig/proyectos/elvira/.

The metric K2 for a network G and a dataset $\mathcal{D}$ is as follows:

$$f(G:\mathcal{D}) = \sum_{i=1}^{n}\sum_{k=1}^{s_i}\left[\log\left(\frac{\Gamma(r_i)}{\Gamma(N_{ij}+r_i)}\right)\sum_{j=1}^{r_i}\log(\Gamma(N_{ijk}+1))\right] \tag{2}$$

where N_{ijk} is the frequency of the cases found in the database $\mathcal{D}$ in which X_i takes its k value conditioned to the j configuration of the parents; n is the number of variables; s_i is the number of possible configurations of the set of parents of X_i; r_i is the number of values that variable X_i can take;

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \tag{3}$$

and Γ is the Gamma function.

BDEu metric depends on a single parameter (the equivalent sample size) S and is defined as follows:

$$\textit{gBDeu}(G:\mathcal{D}) = \sum_{i=1}^{n}\left[\log\left(\frac{\Gamma\left(\frac{S}{q_i}\right)}{\Gamma\left(N_{ij}+\frac{S}{q_i}\right)}\right)+\sum_{k=1}^{r_i}\log\left(\frac{\Gamma\left(N_{ijk}+\frac{S}{q_i r_i}\right)}{\Gamma\left(\frac{S}{q_i r_i}\right)}\right)\right] \tag{4}$$

If we call

$$\textit{Score}(X_i,\Pi_i|\mathcal{D}) = \log\left(\frac{\Gamma\left(\frac{S}{q_i}\right)}{\Gamma\left(N_{ij}+\frac{S}{q_i}\right)}\right)+\sum_{k=1}^{r_i}\log\left(\frac{\Gamma\left(N_{ijk}+\frac{S}{q_i r_i}\right)}{\Gamma\left(\frac{S}{q_i r_i}\right)}\right),$$

then BDEu metric is simply $\textit{gBDeu}(G:\mathcal{D}) = \sum_{i=1}^{n}\textit{Score}(X_i,\Pi_i|\mathcal{D})$.

BIC metric is defined as follows:

$$B(G:\mathcal{D}) = \sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\frac{N_{ijk}}{N_{ik}} - \frac{1}{2}C(G)\log N \tag{5}$$

where: N is the number of records in the database; $C(G)$ is a measure of the complexity of network G, defined as:

$$C(G) = \sum_{i=1}^{n}(r_i-1)s_i \tag{6}$$

Finally, the Akaike information criterion is a modification of BIC metrics given by:

$$f(G:\mathcal{D}) = \sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\frac{N_{ijk}}{N_{ik}} - C(G) \tag{7}$$

Once a metric is considered, then a search algorithm tries to find the graph optimizing the score in the space of directed acyclic graphs. In the following we describe one of the simplest search strategies:

`K2 Algorithm`: This algorithm makes a greedy and very effective search for finding a good quality graph in an acceptable time [18]. It is a hill climbing algorithm. This algorithm assumes that there is a prior order between the variables of the problem in such a way that the parents of X_i can be chosen from the set of preceding variables $X_1, \ldots, X_{i-1}$. The basic steps can be found in Algorithm 1.

Algorithm 1 Computing the groups of variables

```
Let G be the empty graph
for all variable X_i do
    Let Π_i the empty set
    while there are changes in set Π_i do
        Let max = Score(X_i, Π_i|D)
        Let m = 0
        for all variable X_j in {X_1, ..., X_{i-1}} \ Π_i do
            Compute x = Score(X_i, Π_i ∪ {X_j}|D)
            if x > max then
                max = x
                m = j
            end if
        end for
        if max > Score(X_i, Π_i|D) then
            Π_i = Π_i ∪ {X_m}
        end if
    end while
    For each variable X_j ∈ Π_i add an arc from X_j to X_i in G
end for
return G
```

When in a problem there are key variables that are missing (not observed in data $\mathcal{D}$) this algorithm can produce that the relations between observed variables are too complex. This is due to the fact that many conditional independence relationships hold when the key variables are observed. Methods learning graphical models with hidden variables are also considered as clustering methods: each value of a hidden variable defines a cluster or group of cases.

AutoClass is the simplest procedure to learn a Bayesian network with one hidden variable. AutoClass assumes that the graph has the same structure than a Naive Bayes classifier in which the hidden variable is the class. The model is a finite mixture of independent probability distributions, each with its set of parameters. It implies that the data are conditionally independent given the (hidden) class. When new observations are obtained for a new case, a probability of belonging to a class (or weight) can be computed in the learned model.

To estimate the parameters the EM algorithm (Expectation Maximization) [5], which is a method to find the maximum likelihood estimator of the parameters of a probability distribution, is considered. This algorithm is useful when part of the observations are missing. The algorithm considers an initial set of parameters. Then, two steps should be followed to find the optimal parameters: first (Expectation) calculate the expectation of the likelihood with respect to known information and proposed parameters, then (Maximization) maximize this expectation with respect to parameters. These two steps are repeated until convergence [19].

Assume a set of variables $\mathbf{Z} = (\mathbf{X}; \mathbf{Y})$, where data $\mathbf{X}$ are visible, but data $\mathbf{Y}$ are hidden. In this case $\mathbf{Y}$ is the class C. We assume that Θ is the vector of parameters of the associated model (in our case the conditional probability distributions of the Naive Bayes model in which C is the root and there is a link from this variable to each one of the variables in $\mathbf{X}$). In this situation, we have that

$$P(\mathbf{z}|\Theta) = P(\mathbf{x}, \mathbf{y}|\Theta) = P(\mathbf{x}|\mathbf{y}, \Theta)p(\mathbf{y}|\Theta) \tag{8}$$

Assume that the set of observations about $\mathbf{X}$ is give by $\mathcal{D} = \{\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^N}\}$ the logarithm of the likelihood function of the parameters given the sample is:

$$LL(\Theta|\mathcal{D}) = \log \prod_{i=1}^{N} P(\mathbf{x}^i|\Theta) = \sum_{i=1}^{N} \log \left(\sum_{\mathbf{y}} P(\mathbf{x^i}, \mathbf{y}|\Theta) \right) \tag{9}$$

This expression is not simple to evaluate in closed form due to the logarithm of the sum. EM algorithm tries to find a local optimum of this function with an iterative algorithm. It starts with an initial assessment of the parameters Θ^t, and then it repeats the two following steps until convergence:

– Step E (Expectation): calculate the expected logarithm of the likelihood with respect to known information and parameters $\Theta^{(t)}$:

$$Q(\Theta, \Theta^t) = \sum_{i=1}^{N} E[\log P(\mathbf{x^i}, \mathbf{y}|\Theta^t)] \tag{10}$$

where the expectation is taken with respect to variable $\mathbf{Y}$ considering Θ^t as parameters.

– Step M (Maximization): maximize Q with respect to the parameters:

$$\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta, \Theta^t) \tag{11}$$

– Iteratively repeat E and M with Θ^{t+1} in the place of Θ^t, till the difference between the parameters is below a given threshold.

An important issue is to determine the number of cases of hidden class C. This can be done considering different number of cases and selecting the one optimizing the BIC score (5).

3. Hierarchical clustering

In this section we propose a new method for unsupervised classification based on building a hierarchy of artificial variables on top of the observed variables $\mathbf{X} = \{X_1, \ldots, X_n\}$. The basic idea is similar to AutoClass but instead of using all the variables, first it makes a partition of the variables looking for groups with a high degree of dependence of the variables belonging to it.

The final result will be a Bayesian network $\mathcal{B}$, which initially contains variables $\mathbf{X}$ and no links. The basic stages are described in the following subsections.

3.1. Clustering variables

First compute an $n \times n$ matrix A, where a_{ij} represents the degree of dependence of variables X_i and X_j. This degree of dependence is computed with BDEu score. If $\mathcal{D}$ is the set of observed data, then

$$a_{ij} = \mathit{Dep}(X_i, X_j|\mathcal{D}) = \mathit{Score}(X_i, \{X_j\}|\mathcal{D}) - \mathit{Score}(X_i, \emptyset|\mathcal{D}) \tag{12}$$

where $Score$ is computed with BDEu with a given parameter S.

a_{ij} is a measure of the degree of dependence of variable X_i and X_j. It considers two graphs with two variables, X_i and X_j: one in which X_j is a parent of X_i (dependence case) and other in which the graph is empty (the independence case) and it computes the difference between the scores of these two graphs. This value can be positive indicating that there is dependence or negative (independence).

Other measures could be used as the p-value in an test of independence of the two variables, but in comparative experimental studies this difference of scores shows a very good performance [20].

Given the properties of BDEu (equivalent graphs have the same score), we have that $a_{ij} = a_{ji}$, i.e. matrix A is symmetric. When $i = j$, we can consider that $\mathit{Score}(X_i, \{X_i\}|\mathcal{D}) = 0$, and $\mathit{Dep}(X_i, X_i|\mathcal{D}) = -\mathit{Score}(X_i, \emptyset|\mathcal{D})$. However, this value is not important for us and it will not be computed, setting $a_{ii} = 0$.

Then, we define the following relation in $\mathbf{X}$:

$X_i \to X_j$ if and only if $X_j = \arg\max_k a_{ik}$, $i \neq j$, and $a_{ij} > 0$.

i.e. each variable X_i points to the variables X_j with a greatest degree of dependence with X_i given that this degree of dependence is greater than 0.

Now, this relation is extended to be a symmetrical relation:

$X_i \leftrightarrow X_j$ if and only if $X_i \to X_j$ or $X_j \to X_i$.

Finally, this relation is extended to be transitive and reflexive:

$X_i \equiv X_j$ if and only if there is a finite sequence $X_{k_1}, \ldots X_{k_m}$, such that $X_i = X_{k_1}, X_j = X_{k_l}$ and $X_{k_{l-1}} \leftrightarrow X_{k_l}$, $\forall\, 2 \leqslant l \leqslant m$.

The groups of variables $\mathcal{P} = \{\mathbf{C_1}, \ldots, \mathbf{C_k}\}$ are the equivalence classes of relation ($\equiv$). Then, a group of variables $\mathbf{C_l}$ is the smallest non-empty set of variable such that for any variable $X_i \in \mathbf{C_l}$, the variable X_j with greatest degree of dependence with X_i is also in the same group ($X_j \in \mathbf{C_l}$) when $a_{ij} > 0$.

To compute this partition of the set of variables $\mathbf{X}$ we can use a simple procedure that starts with a partition $\mathcal{P} = \{\{X_1\}, \ldots, \{X_n\}\}$, i.e. in the initial partition the groups of variables only contains a single variable.

Then, two groups $\mathbf{C_1}$ and $\mathbf{C_2}$ are joined if there is a variable $X_i \in \mathbf{C_1}$ such that the variable with greatest dependence with it X_j is in the other group $\mathbf{C_2}$ and $a_{ij} > 0$.

Details are in Algorithm 2.

Algorithm 2 Computing the groups of variables

```
Set P ← {{X_1}, ..., {X_n}}
for all pair of variables X_i, X_j (i ≠ j) do
  if a_ij = max_k a_ik and a_ij > 0 then
    Let C_1 ∈ P, such that X_i ∈ C_1
    Let C_2 ∈ P, such that X_j ∈ C_2
    if C_1 ≠ C_2 then
      C ← C_1 ∪ C_2
      P ← (P ∪ {C}) \ {C_1, C_2}
    end if
  end if
end for
return P
```

3.2. Adding artificial hidden variables

Once the partition $\mathcal{P}$ has been computed, we consider a variable Y_i for each group $\mathbf{C_i} \in \mathcal{P}$ if the number of elements in $\mathbf{C_i}$ is greater than 1: $|\mathbf{C_i}| > 1$. These variables Y_i associated to groups $\mathbf{C_i}$ are added to Bayesian network $\mathcal{B}$. We also add a link from Y_i to each variable X_j such that $X_j \in \mathbf{C_i}$. This variable is a hidden variable with no real observations. The idea is that Y_i contains a value for each cluster or group of individuals taken into account only variables in $\mathbf{C_i}$. Instead of making only one classification as in AutoClass, we make a different classification for each groups of variables $\mathbf{C_i} \in \mathcal{P}$.

To assign an initial number of cases to variables to variable Y_i, we first select a representative variable in $\mathbf{C_i}$. To do it, we compute for each variable $X_j \in \mathbf{C_i}$ the value $c_j = \sum_{X_k \in C_i} a_{jk}$ and then we select the variable $X_{j_i} \in \mathbf{C_i}$ with greatest c_j among all the variables in $\mathbf{C_i}$. The idea is that a representative variable is a variable with a highest degree of dependence with the rest of the variables in the group. Once, X_{j_i} is computed we assign to Y_i a number of cases equal to the number of cases of X_{j_i}.

3.3. Recursive computation

If the number of artificial variables Y_i is greater than one, everything is repeated again (grouping of variables and adding artificial variables), but using the set of hidden variables $\mathbf{Y} = \{Y_i : \mathbf{C_i} \in \mathcal{P}, |\mathbf{C_i}| > 1\}$ instead of initial set of variables $\mathbf{X}$. To apply above procedure, we need to determine a new matrix A' with dependences between variables in $\mathbf{Y}$. We make the assumption that as Y_i summarizes the common information of variables in group $\mathbf{C_i}$, then the dependence between Y_i and Y_j is estimated as the average of dependences between the variables in $\mathbf{C_i}$ and variables in $\mathbf{C_j}$. Namely,

$$a'_{ij} = \frac{1}{|\mathbf{C_i}| \cdot |\mathbf{C_j}|} \sum_{X_l \in \mathbf{C_i}, X_k \in \mathbf{C_j}} a_{lk} \tag{13}$$

With variables $\mathbf{Y}$ and matrix A' we repeat the steps of clustering variables and adding artificial variables till the number of artificial variables is less or equal than 1. Each time an artificial variable is added a link is added from the artificial variable to each one of the variables in the associated group.

3.4. Parameter estimation and optimization

Assume that we have computed the structure of the Bayesian network with original and artificial variables which, in general is a tree or forest of trees (if some cluster only contain one variable, then no artificial variable will be added and this part will be disconnected from the rest). Now we have to estimate the parameters and to optimize the number of cases of each artificial variable. We will assume that $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$ is the set of observed variables $\mathbf{X}$ and artificial variables $\mathbf{Y}$.

For each conditional probability $P(Z_i|\pi_{ik})$, where π_{ik} is the k-th configuration of the parents of Z_i, Π_i, we consider the vector of parameters $\Theta_{ik} = (\theta_{i1k}, \ldots, \theta_{ir_ik})$ where $\theta_{ijk} = P(Z_i = z_i^j|\pi_{ik})$ and we assume that this vector of parameters of each conditional probability $P(Z_i|\pi_{ik})$ follows a Dirichlet distribution $D(1, \ldots, 1)$, which implies that parameters are estimated with the Laplace correction. Then, we apply EM algorithm to maximize the posterior probability. For that, we need an initial estimation of the parameters. Taking into account that EM is guaranteed to converge to a local maximum, the quality of the initial parameters can have an influence in the quality of the final values. We propose a procedure based on assigning an observed variable to each artificial variable. This is done in a recursive way. The first time that the clusters are computed and artificial variables are added, a variable Y_i is added for each

group of observed variables C_i, then Y_i is associated with the representative variable X_{j_i} of this group as computed in Subsection 3.2. In the following steps, the groups C_i will be composed of artificial variables (each one of them with an associated observed variable). Then, we proceed as in the initial step with the only difference that the variable Y_i added for group C_i will have as associated observed variable the same observed variable then the most representative variable of group C_i, that now will be an artificial variable.

Now, to give an initial estimation of the parameters of $P(Z_i|\pi_k)$ we compute for each value z_i^j of Z_i the number of occurrences N_{ijk} in data $\mathcal{D}$, where each artificial variable is replaced by its associated observed variable (in that way, the frequencies can be computed). Given these frequencies, the initial estimation is

$$\theta_{ijk}^0 = \frac{N_{ijk} + \log(N+1) + (R_{ijk})}{\sum_j (N_{ijk} + \log(N+1) + (R_{ijk}))}.$$

where R_{ijk} is a random number between 0 and 1. In this way, we assume that the artificial variable associated to a cluster of variables has a similar behavior to the most meaningful variable in the cluster, but we add a random factor (to avoid early convergences due to identical initial parameters for different values of the variables) and a smoothing factor ($\log(N+1)$) similar to Laplace correction but increasing with the sample size, to avoid start the algorithm with too extreme (very close to 0 or 1) initial parameters.

Once we have a first estimation of the parameters, we proceed with EM algorithm for the optimization of posterior probability with Laplace correction. The two steps to improve a set of parameters Θ^t are as follows:

- *Expectation*: Compute $N_{ijk}^t = E[N_{ijk}|\Theta^t] = \sum_{\mathbf{x}\in\mathcal{D}} P(Z_i = z_i^j, \Pi_i = \pi_{ik}|\mathbf{X} = \mathbf{x})$.
  These conditional probabilities are computed in the learned Bayesian network with current parameter set Θ^t.
- *Maximization*: Update the parameter vector to a new value maximizing posterior probability (using Laplace correction):

$$\theta_{ijk}^{t+1} = \frac{N_{ijk}^t + 1}{\sum_{j=1}^{r_i}(N_{ijk}^t + 1)}$$

The two steps are repeated till the difference $|\theta_{ijk}^{t+1} - \theta_{ijk}^t|$ is less or equal than a given threshold ϵ for all the parameters.

Finally, once the parameters are optimized, we proceed with an estimation of the number of cases for each artificial variable Y_l. To do it, we score the different options with BIC or Akaike scores. In this case, the scores are computed as in Eqs (5) and (7), where

$$\sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\frac{N_{ijk}}{N_{ik}}$$

is replaced by

$$\sum_{i=1}^{n}\sum_{j=1}^{r_i}\sum_{k=1}^{s_i} N_{ijk}\log\theta_{ijk}$$

and θ_{ijk} are the parameters estimated with EM algorithm.

To do this, we try to enlarge the number of cases of variable Y_l doing so while the BIC or Akaike metric applied to the resulting networks improves. If no improvement has been obtained by increasing the number of cases of Y_i, then we try to optimize BIC or Akaike metrics by reducing the number of cases.

It is important to remark that each time that the number of cases of an artificial variable changes we have to recompute the optimal parameters by using EM algorithm in order to compute BIC and Akaike scores. Again, the initial parameters can be important for a fast convergence. If we change only the number of cases of variable Y_l, then in order to select the initial parameters for each conditional probability $\theta_{ijk} = P(Z_i = z_i^k | \pi_{ij})$, we follow the following rules:

- If Y_l is not involved in $P(Z_i|\Pi_i)$, i.e. is not Z_i and it is not included in Π_i, then the initial parameters are the ones obtained in previous application of EM algorithm before changing the number of states of Y_l.
- If Y_l is Z_i, then the number of cases of Z_i changes but not the number of configurations of Π_i, then
  - If the number of cases of Z_i increases, i.e. changes from $\{z_i^1, \ldots, z_i^{r_i}\}$ to $\{z_i^1, \ldots, z_i^{r_{i+1}}\}$, then initial parameters are computed as:

$$\theta_{ijk}^0 = \begin{cases} \theta_{ijk} \dfrac{r_i}{r_i + 2R_{ik}} & \text{if } j \leqslant r_i \\ \dfrac{2R_{ij}}{r_i + 2R_{ik}} & \text{if } j = r_i + 1 \end{cases}$$

    where R_{ij} is a random number between 0 and 1 and θ_{ijk} are the previous parameters before adding a new value. The idea is that the new value $z_i^{r_i+1}$ has a random behavior, and the other values of Z_i have similar probabilities to the old r_i values of Z_i. With this assessment we have that if the conditional probabilities of Z_i are conditioned to the fact that Z_i takes one of its old values, then these probabilities are the same than before increasing the number of values of Z_i.
  - If the number of cases of Z_i decreases, i.e. changes from $\{z_i^1, \ldots, z_i^{r_i}\}$ to $\{z_i^1, \ldots, z_i^{r_i-1}\}$, then initial parameters are computed as:

$$\theta_{ijk}^0 = \begin{cases} \theta_{ijk} & \text{if } j < r_i - 1 \\ \theta_{ijk} + \theta_{i(j+1)k} & \text{if } j = r_i - 1 \end{cases}$$

    where θ_{ijk} are the previous parameters before deleting new value. The idea is that the new value, $z_i^{r_i-1}$ plays the role of the union of former values, $z_i^{r_i-1}$ and $z_i^{r_i}$, so the probability of the new value is the addition of the previous probabilities of the two elements.
- If Y_l is different from Z_i and $Y_l = \Pi_i$, i.e. Y_l is the only parent of Z_i (as the network is a tree, if Y_l appears in the set of parents, then it will be the only parent),
  - If the number of cases of Y_l increases, i.e. changes from $\{y_l^1, \ldots, y_l^{r_l}\}$ to $\{y_l^1, \ldots, y_r^{r_{l+1}}\}$, then initial parameters are computed as:

$$\theta_{ijk}^0 = \begin{cases} \theta_{ijk} & \text{if } k \leqslant r_l \\ 0.5 \dfrac{\sum_{k=0}^{r_l} \theta_{ijk}}{r_l} + R_j & \\ \dfrac{}{0.5 + \sum_j R_j} & \text{if } k = r_l + 1 \end{cases}$$

where R_j is a random number between 0 and 1 and θ_{ijk} are the previous parameters before adding a new value. The idea is that when conditioning to the old values of Y_l, the conditional probabilities do not change, and when conditioning to the new value $y_l^{r_l+1}$, then the conditional probabilities are proportional to the average of the conditional probabilities plus a random factor (the denominator $(0.5 + \sum_j R_j)$ is a normalization factor so that the probabilities sum 1 when adding in j).

* If the number of cases of Y_l decreases, i.e. changes from $\{y_l^1, \ldots, y_l^{r_l}\}$ to $\{y_l^1, \ldots, y_l^{r_l-1}\}$, then initial parameters are computed as:

$$\theta^0_{ijk} = \begin{cases} \theta_{ijk} & \text{if } k < r_l - 1 \\ 0.5(\theta_{ijk} + \theta_{ij(k+1)}) & \text{if } k = r_l - 1 \end{cases}$$

where θ_{ijk} are the previous parameters before deleting new value. We follow the same idea than when increasing the number of values: $y_l^{r_l-1}$ plays the role of the union of former values $y_i^{r_l-1}$ and $y_l^{r_l}$, so when conditioning to the new value we compute the average probability of conditioning to the two former values.

The optimization of the values of the artificial variables is done variable by variable, so it is possible that if Y_i is optimized after Y_j, then after optimizing Y_i, the number of optimal values of Y_j would change, so the process of optimization of the number of values of a variable should be repeated while there are changes in the number of cases of the rest of variables. However, we have noticed that this rarely happens in practice.

When decreasing the number of cases of a variable it could be the case that the optimal number of cases is 1. A variable with only one case has not influence in the rest of variables and this is equivalent to removing it and the links with other variables.

4. Comparative experimental analysis

In this section we carry out an experimental study to test the performance of the proposed program using program Elvira [15]. We have carried out three experiments. In all of them we compare our clustering procedure with other learning algorithms as K2, PC, Naive Bayes (the last variable is the class) and AutoClass. In the first one, we try to assess the capabilities of our procedure to represent a joint probability distribution. For that we compare the performance of the different algorithms in standard UCI databases [9]. In the second, we analyze the data of student desertion and socio-economic indicators from the University of Quevedo (Ecuador). In the third experiment, we have carried out a similar study to the student desertion data, but now with the UCI database *Turkiye Student Evaluation Data Set* [6]. The comparison is done with a 10-fold cross validation. As, the objective is to represent the joint probability of the variables in the problem, for each case in the test data, we have measured the logarithm of the probability of the new case assigned by the learned Bayesian network model. We report the sum of these values for all the test cases.

4.1. Comparison with UCI databases

In this section we report the results in UCI databases of the log-likelihood computed by 10-fold cross validation (Table 2). AClass i is the AutoClass procedure with i unobserved values. HNBayesEM A is

Table 2
Log-likelihood in 10-fold cross validation

Datasets	PC	K2	NaiveBayes	AClass 2	AClass 5	HNBayesEM A	HBayesEM B
Mammographic	−8440.69	−7899.52	−7858.38	−7649.42	−7615.95	−7885.965	−7898.26
Lung-cancer	−1440.19	−1948.59	−1489.05	−1418.50	−1410.84	−1366.07	−1366.38
Hepatitis	−4262.48	−4567.96	−4358.94	−3606.53	−3558.61	−3603.19	−3608.51
E colic	−5300.78	−4898.12	−5008.56	−3470.97	−3240.55	−3450.27	−3452.84
Breast-cancer-w	−18404.49	−16680.98	−16317.35	−12871.11	−12694.05	−12922.98	−13046.08
Contact-lenses	−107.90	−95.83	−101.19	−104.58	−104.13	−102.51	−102.50
Hayes-roth-m	−1974.11	−1984.55	−1918.73	−1442.43	−1393.92	−1431.70	−1433.62
Monk 1	−1947.65	−1422.21	−1920.35	−1466.90	−1454.540	−1466.74	−1466.93
Vote	−4918.90	−3328.22	−3768.07	−3497.17	−3371.00	−3305.59	−3338.69
Balance-scale	−4616.48	−4649.34	−4419.63	−4426.84	−4434.59	−4424.79	−4424.75
Tic-tac-toe	−9827.24	−9324.91	−9738.50	−9761.78	−9543.61	−9415.01	−9539.52
Iris	−1622.46	−1369.90	−1255.81	−1365.65	−1270.99	−1291.65	−1371.01
Labor	−650.00	−661.86	−626.98	−630.17	−602.03	−600.86	−612.31
Soybean	−922.11	−600.63	−528.67	−810.13	−584.90	−606.99	−659.44

our hierarchical procedure with Akaike metric to optimize the number of cases of variables and HNBayesEM B corresponds to the use of BIC metric.

We have carried out a non-parametric Friedman test in which there are significant differences between procedures. The post hoc analysis shows that these differences are significant between PC and AClass 2, AClass 5, HNBayesEM A and HBayesEM B (high significance) and between PC and NaiveBayes (moderate significance). PC is the worst performing learning algorithm.

4.2. Desertion data

This dataset contains socio-economic and academic variables measured in 773 students in the Polytechnics School of Quevedo University (Ecuador) during academic year 2012/13. The measured variables can be seen in Table 3. The program variable (A) has as values the different degrees that can be taken in this school: Systems Engineering, Graphic Design Engineering, Mechanical Engineering, Industrial Engineering, Telematics Engineering, Electrical Engineering, Agroindustry Engineering, and Industrial Security Engineering. Variable R (Passes) determines whether the student passed in previous academic year and variable S (drops out) determines whether the student deserts the degree.

Continuous variables have been discretized in meaningful intervals. For example, the variable cost of education represented by the letter E has been discretized in values below 200 dollars, from 200 - 800, and more than 800.

The log-likelihood results under 10-fold cross validation are given in Table 4. The best results are obtained for K2 algorithm. Our procedure obtains similar results, specially with Akaike metric. The performance of the other procedures is poorer.

Additionally, our procedure has an additional advantage over K2 and other procedures learning a generic Bayesian network. In our case, the graph is always a tree (or forest tree) making the interpretation and computations easier. To see that in Fig. 3 we can see the network learned with K2 algorithm using the entire database and in Fig. 4 the network learned with our hierarchical clustering procedure with Akaike metric. We can see that both methods determine the same set of variables that are dependent of S (drops out) and these are R (passes), A (program), B (academic year), and E (education cost). But in K2 algorithm all the dependences vanish when conditioning in R. This is the only variable directly affecting S. However, our model, even being a simple one, is able of representing that given R, the cost of the education is affecting S through variable *AuxNode*5. Higher costs imply an increasing in the probability

Table 3
Variables and their descriptions

Variable	Description
A	Program
B	Academic year
D	Disabled
E	Education cost
F	Lives apart from family
G	Type of family house
H	House-owner
I	Cable Tv services
J	Credit card services
K	Internet access services
L	Basics services
M	Private transport services
N	Cellphone plan services
O	Own car services
P	Comes in own car services
Q	Currently working
R	Passes
S	Drops out

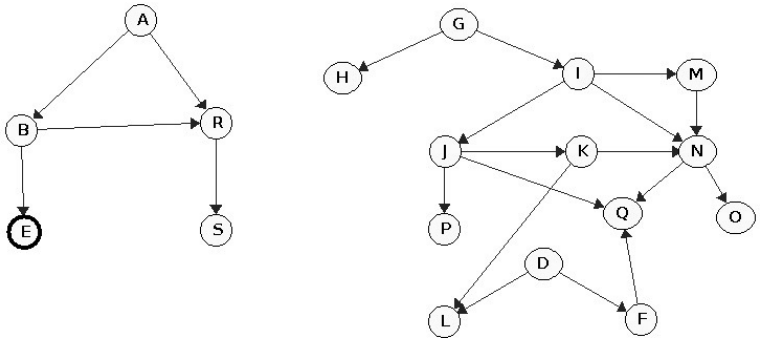

Fig. 3. Bayesian network K2.

of deserting, even if R is known. For example, in our model, knowing that a student does not pass, the probability of desertion changes from 0.19 to 0.34, depending of the cost of education. The program in which the student is enrolled (variable A) has an influence in the probability of desertion. However, this influence is weaker (as it goes through hidden variables *AuxNode*0, *AuxNode*7, *AuxNode*5). Even if we know whether the student pass, this variable has an influence on desertion, though a weak one from a quantitative point of view.

On the other hand, the relationships between the other variables are summarized through the use of several hidden auxiliary variables. For example, *AuxNode*4 considers the variables related to having own car. *AuxNode*3 summarizes all the technology related services. Finally, all these socio-economic variables are summarized in one variable *AuxNode*6. The relations of these variables in K2 algorithm are more cumbersome. Though there is a tendency to have direct links for the variables in the same group, the way variables relate among them is more complex.

We would like also highlight that AutoClass only introduces one hidden variable which is a parent of all the observed variables. For problems like this one in which there are different groups of variables, this is not appropriate as it forces all the variables to have strong relations among them: for example, if P and O have a high degree of dependence, as this dependence is only obtained through the only hidden variable, both of them will have a high degree of dependence with the hidden variable, and then they will have dependence with all the other variables that are dependent of the hidden class. This might explain the poor results of AutoClass when considering log-likelihood performance (see Table 4). A hierarchy of hidden variables seem to be more appropriate in this case.

4.3. Student evaluation

In this case we have used the UCI dataset *Turkiye Student Evaluation Data Set* [6] with variables in Table 3. All the variables Qi take five values $1, \ldots, 5$. The log-likelihood results under 10-fold cross validation are given in Table 5.

In this case the best results are obtained by using our procedure with Akaike metric. Our procedure with BIC score provides also similar good results. The performance of AutoClass depends on the number

Table 4
Log-likelihood in 10-fold cross validation. Desertion data

PC	K2	NaiveBayes	AClass 2	AClass 5	HNBayesEM A	HBayesEM B
−8832.76	−8403.18	−8780.48	−8660.89	−8607.18	−8452.37	−8494.48

Table 5
Log-likelihood in 10-fold cross validation. Desertion data

PC	K2	NaiveBayes	AClass 2	AClass 6	AClass 12	HNBayesEM A	HNBayesEM B
−263425.48	−130190.69	−171900.24	−228961.76	−140751.66	−128164.73	−113482.01	−114998.16

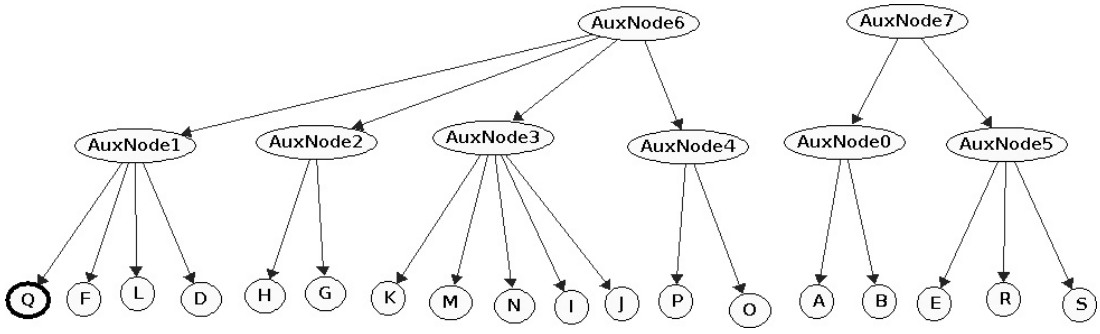

Fig. 4. Bayesian network with hierarchical clustering for socio-economic data.

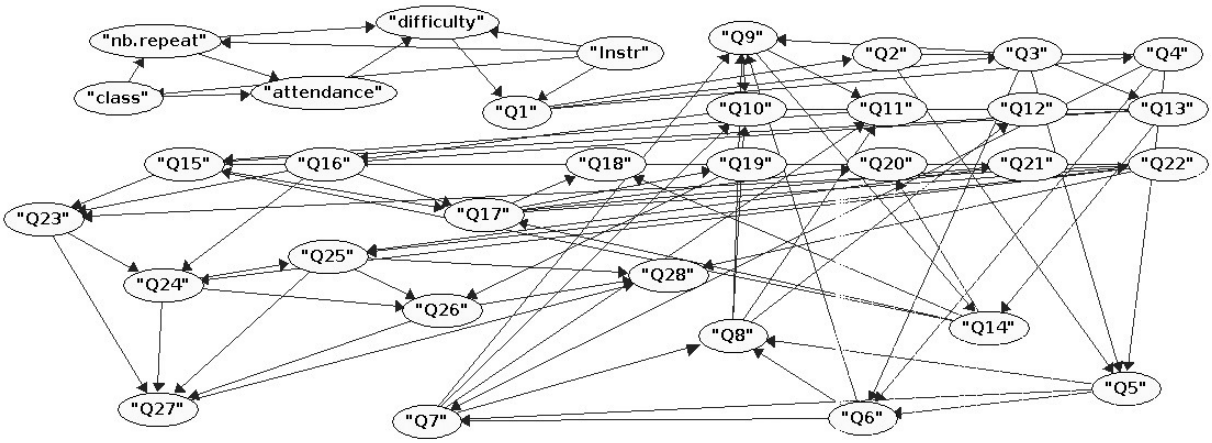

Fig. 5. K2 network for student evaluation.

of hidden classes, being very poor for 2 cases, but improving with the number of cases. With 12 hidden values is even better than K2. This case seems to be specially appropriate for our method as there are a high number of variables which are pairwise related, but that they measure different aspects of the general degree of satisfaction of students with a course. So, only one hidden variable does not seem to be enough. This is more evident in the learned tree with the observed and hidden variables which is depicted in Fig. 6 in which Akaike metric is used. We can see the following facts:

– Variable *AuxNode*2 summarizes variables $Q1, Q2, Q3, Q4$, which are variables related to the information provided to the student. Variable *AuxNode*3 summarizes variables $Q5, Q6, Q7, Q9, Q10$

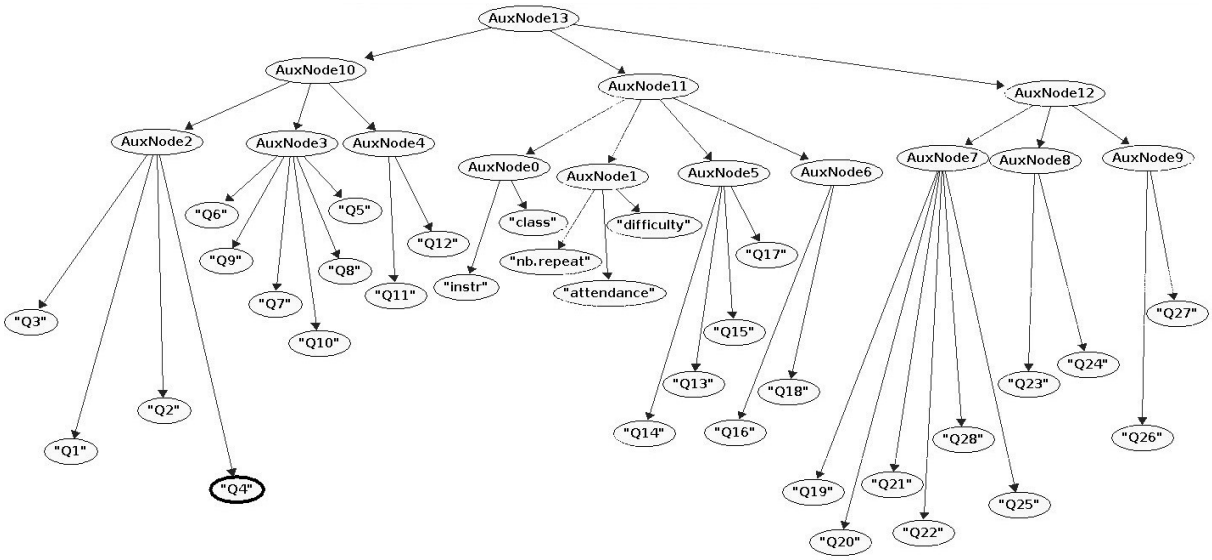

Fig. 6. Bayesian network with hierarchical clustering for student evaluation.

which are related to the materials use by the instructor. Variable *AuxNode*4 is associated with variables $Q11$ and $Q12$ related to the expected profit to the student from the course. At the same time, these three variables *AuxNode*2, *AuxNode*3, *AuxNode*4 constitute a new group with artificial variable *AuxNode*10.

– *AuxNode*0 is the artificial variable for the group of variables related to the course and the instructor, while *AuxNode*1 corresponds to the group of variables related to the difficulty of the course. *AuxNode*5 is the variable for the group $\{Q13, Q14, Q15, Q17\}$, variables related to the teacher preparation. *AuxNode*6 is related to the pedagogical abilities for the exposition by the instructor. All these hidden variables are grouped on variable $Aux11$.
– *AuxNode*7 covers variables $Q19, Q20, Q21, Q22, Q25, Q28$, that are related to the attitude of the instructor with the students. *AuxNode*8 is the variable associated to variables $Q23, Q24$ which are related with student participation, while *AuxNode*9 depends on $Q26, Q27$, variables relative to the student evaluation. All these 3 hidden variables are grouped with *AuxNode*12 as artificial variable.

In summary, our procedure produces a very good estimation of the joint probability distribution, but at the same time produces a natural and easy to interpret graphical structure grouping observable variables in a hierarchy. In contrast, the model learned with K2 algorithm (see Fig. 5) is extremely complex and difficult to understand.

5. Conclusions and future research

In this paper we have proposed a new procedure to induce probabilistic graphical models with hidden variables from data: a hierarchical clustering. Instead of considering only one hidden variable as in AutoClass, we consider a tree of hidden variables. Experiments have shown that this model have good performance for estimating a joint probability distribution from a dataset of observations, but it has some additional advantages against general learning procedures of Bayesian networks: it produces trees (or forest trees) that are simple to interpret and fast for computing.

The performance of this model depends on the particular case to which it is applied. In general, we can say that it is appropriate in problems in which there are variables that are related between them and that this is due to the fact that they depend of hidden non observed factors. This has been the case of the dataset with data from students from the University of Quevedo (Ecuador) in which we have some variables that are manifestations (depend of) the economic level of the students families, other variables that are related to the academic characteristics of the students, etc.

In the future, we plan to extend the study of desertion including more academic variables similar the ones having more influence in the variable of interest. We also plan to consider the desertion problem as a supervised classification problem, adapting our hierarchical procedure to classification and comparing its performance with other methods.

We also want to single out other practical problems in which our model can have a role, as it can be summarizing the answers obtained in questionnaires, similar to the case of students evaluation considered in this paper.

Acknowledgments

This research was supported by the FOCICYT project of the University State Technique of Quevedo and by the Spanish Ministry of Education and Science under project TIN2013-46638-C3-2-P and the European Regional Development Fund (FEDER).

References

[1] W. Edwards, Hailfinder: Tools for and experiences with Bayesian normative modeling, *American Psychologist* **53**(4) (1998), 416.
[2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.
[3] R. Neapolitan, *Learning Bayesian Networks*, Prentice Hall Upper Saddle River, 2004.
[4] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor and D. Freeman, Autoclass: A Bayesian classification system, in: *Readings in knowledge acquisition and learning*, Morgan Kaufmann Publishers Inc. (1993), 431–441.
[5] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of the Royal Statistical Society, Ser B* (1977), 1–38.
[6] N. Gunduz and E. Fokoue, UCI machine learning repository, 2013.
[7] M.J. Choi, V.Y.F. Tan, A. Anandkumar and A.S. Willsky, Learning latent tree graphical models, *Journal of Machine Learning Research* **12** (2011), 1771–1812.
[8] N.L. Zhang, Hierarchical latent class models for cluster analysis, *The Journal of Machine Learning Research* **5** (2004), 697–723.
[9] M. Lichman, UCI machine learning repository, 2013.
[10] M.A. Magaña, O.A. Montesinos-López, Suárez and C.M. Hernández, Análisis de la evolución de los resultados obtenidos por los profesores en las evaluaciones esdeped y las realizadas por los estudiantes, *Revista de la Educación* Superior **35**(140) (2006), 29–48.
[11] H. Kuna, R. García-Martínez and F. Villatoro, Identificación de causales de abandono de estudios universitarios. uso de procesos de explotación de información, *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología* **5** (2009), 39–44.
[12] G.I. Webb and M.J. Pazzani, Adjusted probability naive Bayesian induction, in: *Proceedings of the Eleventh Australian Joint Conference on Artificial Intelligence*, Springer-Verlag (1998), 285–295.
[13] J.R. Anderson, R.S. Michalski, J.G. Carbonell and T. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, Morgan Kaufmann, 1986.
[14] P. Spirtes, C. Glymour and R. Scheines, *Causation, Prediction and Search*, Springer Verlag, Berlin, 1993.
[15] E. Consortium, Elvira: An environment for probabilistic graphical models, in: *Proceedings of the 1st European Workshop on Probabilistic Graphical Models*, J. Gámez and A. Salmerón, eds, 2002, pp. 222–230.
[16] G. Schwarz, Estimating the dimension of a model, *The Annals of Statistics* **6**(2) (1978), 461–464.

[17] D. Heckerman, C. Kadie and J. Listgarten, Leveraging information across HLA alleles/supertypes improves epitope prediction, *Journal of Computational Biology* **14**(6) (2007), 736–746.
[18] G.F. Cooper and E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, *Machine Learning* **9**(4) (1992), 309–347.
[19] B. Sierra, *Aprendizaje Automático: Conceptos Básicos y Avanzados*, Prentice-Hall, 2006.
[20] J. Abellán and S. Moral, A new imprecise score measure for independence, *Intelligent Data Analysis* **16** (2012), 847–863.