JIRSS (2016)
Vol. 15, No. 2, pp 45-61
DOI:10.18869/acadpub.jirss.15.2.45

Learning Bayesian Network Structure Using Genetic Algorithm with Consideration of the Node Ordering via Principal Component Analysis

Vahid Rezaei Tabar[1,5], Maryam Mahdavi[2], Saghar Heidari[3], Sima Naghizadeh[4]

[1]Department of Statistics, Faculty of Mathematics and Computer Sciences, Allameh Tabataba'i University, Tehran, Iran
[2]Department of Information Technology Management, Faculty of Management, Kharazmi University, Tehran, Iran
[3]Department of Biostatistics, SBMU School of Para-Medical Sciences, Shahid Beheshti University, Tehran, Iran
[4]The National Organization for Educational Testing (NOET), Ministry of Science, Research and Technology, Tehran, Iran
[5]School of Biological Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

Abstract. The most challenging task in dealing with Bayesian networks is learning their structure. Two classical approaches are often used for learning Bayesian network structure: Constraint-Based method and Score-and-Search-Based one. However, neither the first nor the second one are completely satisfactory. Therefore, the heuristic search such as Genetic Algorithms with a fitness score function is considered for learning Bayesian network structure. To assure the closeness of the genetic operators, the ordering among variables (nodes) must be determined. In this paper, we determine the node ordering by considering the Principal Component Analysis (PCA). For this purpose, we first determine the appropriate correlation between variables and then use the absolute value of variable's coefficients in the first component. It means that a node X_i can only

Corresponding Author: Vahid Rezaei Tabar (vhrezaei@atu.ac.ir)
Maryam Mahdavi (mahdavimaryam_6897@yahoo.com)
Saghar Heidari (shsikiu.amar@gmail.com)
Sima Naghizadeh (s_naghizadeh@yahoo.com)

have the node X_j as a parent if the absolute value of coefficient X_j in the first component is higher than X_i. We then use the Genetic Algorithm with fitness score BIC regarding the node ordering to construct the Bayesian Network. Experimental results over well-known networks Asia, Alarm and Hailfinder show that our new technique has higher accuracy and better degree of data matching. In addition, we apply our technique to the real data set which is related to Bank's debtor that owe over 500 million Rials to Maskan Bank in Iran. Results also show that the proposed technique has greater modeling power than other node ordering techniques such as Hruschka et al. (2007), Chen et al. (2008) and K2 algorithm.

Keywords. Bayesian network, Node ordering, PCA, Genetic algorithm

MSC: 62C10; 62F15.

1 Introduction

Bayesian Network (BN) is a specific type of graphical model which is a directed acyclic graph (DAG) where the nodes are random variables and the arcs specify the independence assumptions between the random variables (Pearl (1988), Geiger (1990), Cooper (1992), Jensen (1996), Friedman (1997), Bouckaert (2001), Perrier et al. (2008)). The learning task in a BN can be separated into two subtasks: structure learning that is to identify the topology of the network, and parameter learning which estimates the conditional probabilities for a given network (Heckerman (1998), Ghahramani (1998), Grossman (2004)). In this paper, we concentrate on the structure learning. Two classical approaches are often used for learning a BN structure. The first one is based on the statistical tests and the second one uses a scoring metric (Spirtes et al. (1995), Chickering (1996)). However, neither the first nor the second are completely satisfactory in the task of learning a BN structure (Robinson (1977), Pearl (2011)). Therefore, heuristic search such as Genetic Algorithms (GA) which have emerged for solutions to combinatorial complex problems is considered (Larrañaga et al. (1996), Wong (2000), De Campos (2006)). GAs are a family of computational models inspired by Darwins theory of Evolution. GAs encode potential solutions to a problem in a chromosome-like data structure, exploring and exploiting the search space using dedicated operators. In a GA, the search space of a problem is represented as a collection of individuals which are often referred to as chromosomes. The purpose of the GA is to find the individual from the search space which has the best genetic material. The quality of an individual is measured with an objective function called fitness score. The part of the search space to be examined is called the population. In the process of the GA, a BN structure can be represented by a

connectivity matrix C where its elements satisfy in

$$C = \begin{cases} 1 & \text{if node j is a parent of node i} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, an individual of the population is represented by the string of c_{ij}s, i.e., $c_{11}, c_{12}, \ldots, c_{nn}$, where n is the number of nodes. For evaluating the structures constructed by a GA, we use the Bayesian information criterion (BIC) score. To assure the closeness of the genetic operators, the node ordering must be determined (Larrañaga et al. , 1996). The space of orders is much smaller than the space of network structures. Determining the node ordering is very important because a BN approach for learning structures upon variables can be expensive and lead to large dimension models. In other words, the number of BN structures is super-exponential in the number of random variables in the domain (Chickering , 1996). So, to overcome such difficulties in terms of computational complexity, the node ordering must be considered. It is easy to verify that, in case an ordering is assumed, the connectivity matrices of the network structures are triangulated and the genetic operators are closed operators. In this case, the strings length used to represent a BN structure with n nodes is $\binom{n}{2}$, instead of n^2 of the general case. Note that, if node X_i comes prior to the node X_j in the ordering, then the node X_j cannot be a parent of the node X_i.

In general, node ordering algorithms are categorized into two groups: evolutionary algorithms and heuristic algorithms. Initial research on evolutionary algorithms has provided extensive experimental results through various crossover and mutation methods (Romero et al. , 2004). In terms of heuristic methods, Hruschka et al. (2007) introduced the feature ranking-based node ordering algorithm, which is a type of feature selection method in the classification domain. It measures dependencies of variables over the class label using χ^2 statistical tests and information gain. It then sorts the variables by the dependence-based scores. The sorted variables are regarded as the node ordering. Chen et al. (2008) incorporated information theory and exhaustive search functions in their algorithm. The algorithm comprises three major phases. In the first two phases, it constructs an undirected structure through mutual information, independence tests, and d-separation. The last phase is related to determining the ordering between nodes.

In this paper, we use the Principle Component Analysis (PCA) to determine the node ordering according to the coefficients of variables in the first component (Abdi et al. , 2010). This is based on the fact that the first principal component has the greatest variance (Zwick et al. (1986), Jolliffe (2002)). Then, we use the GA for learning the BN structure. In addition, we use the Hruschka et al. (2007) and Chen et al. (2008) techniques for determining the node ordering as

an input of the GA. Since the K2 algorithm is one of the most famous score-based algorithms which receive the ordering of the variables, we also compare our results with the K2 algorithm (Cooper , 1992).

The paper is organized as follows. In Section 2, the concept of BN structures learning is introduced. Then our methodology for learning a BN structure using the GA as well as node ordering via PCA are introduced in Section 3. In Section 4, we also introduce the K2 algorithm which receives the ordering of the variables for learning the BN structure. Finally the efficiency of our proposed technique is compared with other node ordering techniques such as Hruschka et al. (2007) and Chen et al. (2008) as well as the K2 algorithm. For this purpose, we use three well-known datasets: Asia, Alarm and Hailfinder as well as real data set of Maskan Bank in Iran. This data is related to bank's debtor (corporations) that owe over 500 million Rials.

2 Learning Bayesian Network Structure

The global joint probability distribution of the BN constructed by variables is written as

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | pa(x_i)), \tag{2.1}$$

in which $pa(x_i)$ indicates the parents of the node x_i. It is not always possible for experts to determine the structure of a BN and in some cases the determination of the model can therefore be a problem to solve. The task in the BN structure learning is to find a structure of the BN that describes the observed data the most according to a score function, which is proved to be a NP-complete problem (Chickering , 1996). The methods used for learning the structure of BNs can be divided into two main groups (Heckerman , 1998):

- Constraint-Based Approach: Discovery of independence relationships using statistical tests, e.g. the PC and GS algorithm,
- Score-and-Search-Based Approach: Exploration and evaluation which use a score to evaluate the ability of the graph to recreate conditional independence within the model, e.g., AIC, BIC, and K2.

The Constraint-Based approaches are based on the conditional independence tests under the assumption that graphical separation and probabilistic independence imply each other. The Score-and-Search-Based approaches start from an initial structure (generated randomly or from domain knowledge) and move to

the neighbors with the best score in the structure space determinately or stochastically until a local maximum of the selected criteria is reached. Neither the first nor the second are really satisfactory in the task of learning a BN structure (Robinson , 1977). Therefore, evolutionary methods such as GA have already been used in various forms for learning BN structures (Larrañaga et al. (1996), De Jong (2006)). GAs are search algorithms based on the mechanism of natural selection and genetics. The search space in a GA for learning BN structures is all the possible structures of directed acyclic graphs (DAGs) given the number of variables in the domain. Therefore the number of DAGs is super-exponential in the number of nodes. So, considering the node ordering can reduce the space of network structures.

3 Learning Bayesian Network Structure using Genetic Algorithm and PCA

In this section, the BN structure leaning using a GA is introduced. GAs are search algorithms based on the mechanics of natural selection and natural genetics. The algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness function. 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 which are added to the population. With a probability near zero, all newly created individuals of the resulting population mutate , meaning that they change their hereditary distinctions. After that, some individuals are removed from the population according to the fitness score in order to reduce the population to its initial size. One iteration of the algorithm is referred to as a generation which consists of a population of character strings that are analogous to the chromosome. A chromosome is called a solution and is composed of several genes (categorical variables). The algorithm evolves the following three operators (Larrañaga et al. (1996), Yu et al. (2004)):

- Selection which equates to survival of the fitness,
- Crossover which represents mating between individuals,
- Mutation which introduces random modifications.

The steps of the genetic algorithm are as:

- Making initial population at random,
- Selecting parents from the population,
- Producing children from the selected parents (by crossing operator),
- Mutating the individuals (mutation operator),
- Extending the population by adding the children to it,
- Reducing the extended population.

In the process of a GA, parents are selected according to their fitness. For choosing the better chromosomes, the roulette wheel is chosen. A BN structure constructed by the GA can be represented by an $n \times n$ connectivity matrix C, where its elements c_{ij} is 1 if node j is a parent of node i (Larrañaga et al. , 1996). Therefore, an individual of the population is shown by the string

$$c_{11}, c_{12}, \ldots, c_{1n}, c_{21}, c_{22}, \ldots, c_{2n}, \ldots c_{n1}, c_{n2}, \ldots, c_{nn}.$$

If there is no assumption for node ordering, the genetic operators are not closed operators. Then the cardinality of the search is given by Pearl (1988) as

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} f(n-i), f(0) = 1, f(1) = 1.$$

To assure the closeness of the genetic operators, the node ordering must be determined. The space of orders is much smaller than the space of network structures. This is very important, because the GA starts with a random ordering.

In this paper, the node ordering is determined using the PCA. Since most of the times the variables are measured in different scales, the PCA must be performed with standardized data. For this purpose, we first determine the appropriate correlation between variables. We then perform the PCA on the correlation matrix. As we know, the number of components extracted by PCA is equal to the number of observed variables (Cattell (1966), Cliff (1988)). We only use the first component and consider the coefficients of the variables in this component for determining the ordering between them. The idea behind this fact is that the first principal component has the greatest variance. For instance, into the first component, a node X_i can only have node X_j as a parent if the absolute value of the coefficient of X_j will be higher than that of X_i.

In case an ordering is assumed, the connectivity matrix of the network structure is triangulated. Therefore, the genetic operators are closed operators. Hence, the network is represented by the strings of triangulated matrix and the fitness

score is calculated at each iteration. The initial population of a λ individuals is generated at random. The fitness function to be used to evaluate the quality of a structure is based on the BIC score. Each individual is selected to be a parent with a probability proportional to the rank of its fitness function. If we denote the j^{th} individual of the population at time t by I_t^j, and the rank of its fitness function by $rank(g(I_t^j))$, then the individual I_t^j is selected to be a parent with probability

$$P_{j,t} = \frac{rank(g(I_t^j))}{\lambda(\lambda+1)/2}.$$

Furthermore, the BIC score defined as (De Campos , 2006)

$$S_{BIC}(BN, D) = \log(L(BN, \theta|D)) - \frac{1}{2}Dim(BN).\log(N), \tag{3.1}$$

where D represents the data and $\theta = P(X_i|Pa(X_i))$ specifies the parameters. In Eq. 3.1, N is the total number of instances (cases) in data, and $Dim(BN)$ is the dimension function defined by

$$Dim(BN) = \sum_{i=1}^{n}(r_i - 1) \times \prod_{X_j \in Pa(X_j)} r_j, \tag{3.2}$$

where r_i is the number of possible values of X_i. The $\log(L(BN, \theta|D))$ in Eq. 3.1 is the log-likelihood (LL) score and defined by (De Campos , 2006)

$$\log(L(BN, \theta|D)) = \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk}\log\theta_{ijk}, \tag{3.3}$$

where q_i is the number of possible configurations of the parent set of X_i and equals to $\prod_{X_j \in Pa(X_j)} r_j$, and N_{ijk} is the number of instances in the data where the variable X_i takes its k^{th} value and the variables in $Pa(X_i)$ take their j^{th} configuration. In this paper, we divide the log-likelihood score into two terms as

$$\begin{aligned}\log(L(BN, \theta|D)) = \quad & \sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} N_{ijk} \times \log\theta_{ijk} \times \delta(pa(X_i) \neq \emptyset) \\ & + \sum_{i=1}^{n}\sum_{k=1}^{r_i} N_{i\_k} \times \log\theta_{i\_k} \times \delta(pa(X_i) = \emptyset)\end{aligned} \tag{3.4}$$

where θ_{i_k} indicates the probability that the variable X_i takes its k^{th} value without considering the j^{th} configuration, N_{i_k} is the number of cases with $X_i = k$ and

$\delta(pa(X_i) = \emptyset)$ tells us to include only variables which has no parents. In other words, the first term in Eq. 3.4 is used for those variables (nodes) which have at least one parent and second term is used for those variables that have no parents. The MLE estimate for θ_{ijk} and $\theta^{i}_{i\_k}$ are given as (Heckerman , 1998)

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}}, \tag{3.5}$$

$$\hat{\theta}_{i_k} = \frac{N_{i\_k}}{N_{i\_}}, \tag{3.6}$$

in which $N_{i\_} = \sum_k N_{i\_k}$.

4 K2 Algorithm

K2 algorithm is the most famous score-based algorithm in BNs. K2 algorithm is a greedy heuristic. It starts by assuming that a node lacks parents, after which in every step it adds incrementally that parent whose addition increases the probability of the resulting structure the most (Cooper , 1992). K2 stops adding parents to the nodes when the addition of a single parent cannot increase the probability. The K2 algorithm receives as input a total ordering of the variables which can have a big influence on its result. Thus, finding a good ordering of the variables is also crucial for the algorithm success (Ruiz, 2005). In other words, The K2 algorithm reduces this computational complexity by requiring a prior ordering of nodes as an input. The inputs of the e K2 algorithm are a set of n nodes, an ordering on the nodes, an upper bound on the number of parents as node may have, and a database containing all cases. In this paper, we set the upper bound value to $n - 1$.

As mentioned in Section 1, Hruschka et al. (2007) introduced the feature ranking-based node ordering algorithm and Chen et al. (2008) incorporated information theory and exhaustive search functions for determining the node ordering. However, to the best of our knowledge, the most effective heuristic algorithm for determining the node ordering is the one proposed by Chen et al. (2008) whose time complexity is $O(n^4)$. Therefore, we perform the K2 algorithm considering the node ordering obtained by Chen et al. (2008) technique.

5 Application

5.1 Asia, Alarm and Hailfinder Networks

In this section, we present the empirical results. For this purpose, we use three well-known BNs: Asia, Alarm and Hailfinder. These are BNs from which we

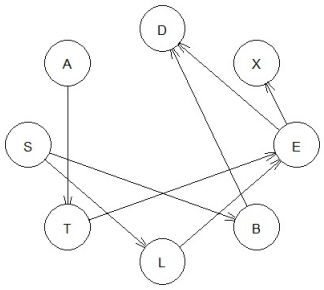

Figure 1: Asia Network

can sample any dataset we want in order to perform multiple tests and estimate more precise metrics. So, we conducted learning exercise using 10000 cases.

The Asia network has 8 variables and each one has two attributes (Figure 1). This BN is a small synthetic about lung diseases (tuberculosis, lung cancer or bronchitis) and visits to Asia (Lamma et al. , 2005). As the variables in the Asia network are nominal (TRUE or FALSE), we use the Nominal vs. Nominal correlation for performing PCA (as the data must be standardized before the analysis, we use the correlation matrix). For this purpose, we calculate Cramer's V. In statistics, Cramér's V (sometimes referred to as Cramér's ϕ) is a measure of association between two nominal variables.

The ALARM network has 37 variables and each one has two, three or four possible attributes (Beinlich et al. , 1989). The original BN structure of ALARM is shown in Figure 2. The 37 nodes in ALARM network can viewed as ordinal variables. So, the Spearman's rank correlation coefficient between variables for performing PCA is considered.

Hailfinder is a Bayesian network designed to forecast severe summer hail in northeastern Colorado (Abramson et al. , 1996). The number of nodes and arcs are 56 and 66, respectively (Figure 3). The 56 nodes in Hailfinder network can be viewed as ordinal variables. Therefore, the Spearman correlation coefficient between variables can be considered for doing PCA.

The existence of the original network structures allows us to define important terms which indicate the performance of the technique. We compare the edge scores by computing the number of edges that are correct, missing, reverse and additional by the following definitions:

- Correct edge: Edges detected with the same edge direction.
- Reverse edge: Edges detected with the opposite edge direction.
- Missing edge: Edges not detected compared to the true structure (original

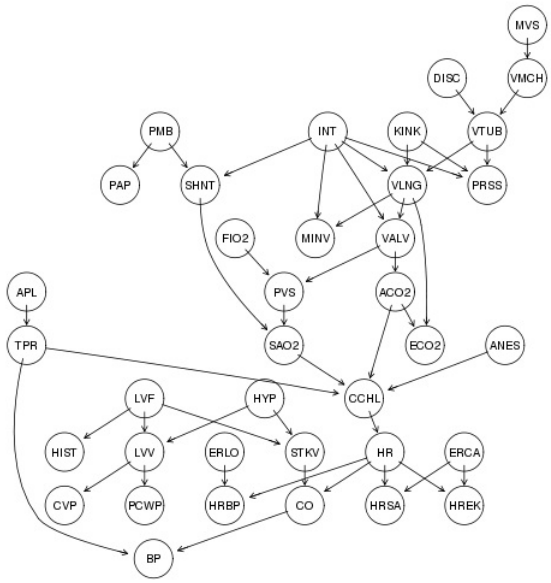

Figure 2: Alarm Network

structure).

- Additional edge: Detected edges that are not present in the true structure.

The edge scores make it possible to define the important terms which indicate the performance of the method. For this purpose, the True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) values are computed. In addition, known measure such as Positive Predictive Value (PPV), True Positive Rate (TPR) and F-score measure (F) are considered (Baesens et al. , 2002). The F-score is defined as

$$F - measure = \frac{2.PPV.TPR}{PPV + TPR}.$$

F-measure is a useful quantity used to compare learned and original (actual) networks. In this paper, F-measure is used as accuracy and degree of data matching. Comparing this measure between different techniques indicates that which technique is more efficient in the task of learning a BN. The algorithm with larger values for correct edges and F-measure is more efficient in the structure learning of a BN.

The input of a GA is determined via the PCA and the techniques in Hruschka et al. (2007) and Chen et al. (2008).

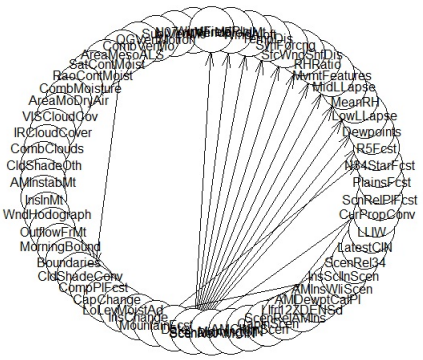

Figure 3: Hailfinder Network

Table 1 shows the correct edges, F-measure values and the graph error which is computed by summing the three erroneous edge type: missing, reverse, and additional for different techniques. It should be noted that we also perform the K2 algorithm with consideration of the node ordering via Chen et al. (2008) techniques. The experimental results show that the proposed technique outperforms

Table 1: Comparing Correct Edges (C), Graph Error (E) and F-measures (F)

Data	Proposed Technique			Hruschka et al.			Chen et al.			K2		
	C	E	F	C	E	F	C	E	F	C	E	F
Asia	5	3	**0.62**	4	7	0.42	5	6	0.52	5	7	0.50
Alarm	23	43	**0.41**	22	49	0.37	23	52	0.35	23	47	0.38
Hailfinder	30	55	**0.44**	22	62	0.29	30	69	0.36	27	61	0.35

other techniques. This methodology causes a significant decrease in the complexity of the GA algorithm. In addition, our proposed technique significantly avoids creating extra edges.

5.2 Real Data set

We also present the empirical results on a real data set. This data set is related to bank's debtors (corporations) that owe over 500 million Rials to the Maskan Banks in Iran from November 2010 to November 2014. We get the data from Maskan Banks in Tehran. The data set has 1358 rows as corporations and 6 columns as variables. All variables can be viewed as ordinal variables of the

Table 2: Coefficients of variables in first component of PCA

Variable's name	Coefficient
X_1	-0.345
X_2	-0.291
X_3	-0.464
X_4	0.381
X_5	0.513
X_6	-0.415

following form:

- Paid amount with four levels (X_1): less than 1,000,000,000 Rials, between 1,000,000,000 and 10,000,000,000 Rials, between 10,000,000,000 and 100,000,000,000 Rials, between 100,000,000,000 and 900,000,000,000 Rials.
- Corporation's Age with 6 levels (X_2): less than 10 years, between 10 and 20, between 20 and 30, between 30 and 40, between 40 and 50, more than 50 years.
- Number of years elapsed since the Date of Contract with 4 levels(X_3): less than 5 years, between 5 and 10 years, between 15 and 20 years, more than 20 years.
- Time of installments with 5 levels (X_4): one year, two years, between 3 and 4 years, between 5 and 7 years, more than 7 years.
- Deferred liabilities with 3 levels (X_5): between 2 and 6 months, between 6 and 18 months, more than 18 months.
- Level of branch with 5 levels (X_6): level 1, level 2, level 3, level 4, and level 5. This variable has been determined by the Maskan Bank based on the place of the branch.

The frequency of all variables are shown in Figure 4. As the variables are ordinal, the Spearman rank correlation is used for doing a PCA. The coefficients of variables in the first component are shown in Table 2. The absolute value of this coefficients is used for determination of the node ordering which is needed for learning of a BN structure using a GA.

Learning a BN structure regarding the node ordering is performed using the GA. The GA algorithm is stopped when 5000 structures have been evaluated with consideration of the following assumptions:

- Population size: 50

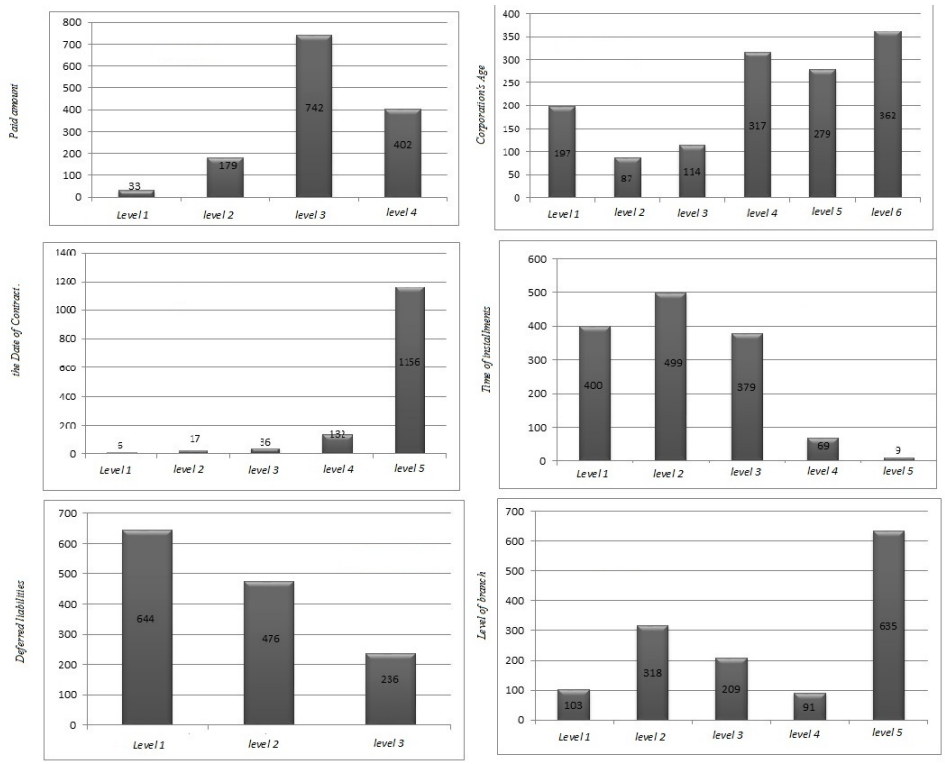

Figure 4: Frequency of the levels of all variables in the bank's debtor data set

- Crossover probability =0.5
- Mutation rate =0.01.

The learned BN structure on a real data set using a GA is shown in Figure 5. To provide a tool for an accurate performance evaluation of our technique, we calculate the BIC score for the learned BN structure. The BIC score of the learned BN structure using proposed technique is -9218.9202 which is higher than the BIC scores of other techniques used in this paper. Therefore we can conclude that the proposed technique is more efficient.

6 Conclusion

Yielding more effective node ordering is an important issue for running a GA in the task of the learning a BN structure. In this paper, using the PCA, we introduce a novel node ordering. This methodology causes a significant decrease

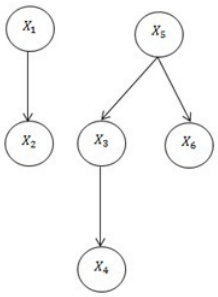

Figure 5: Bayesian Network Structure learned by GA on Real Data Set

Table 3: Time consumed (s) during node ordering

Data Set	Proposed Technique	Hruschka et al. (2007)	Chen et al. (2008)
Asia	8.9 s	30.3 s	11.1s
Alarm	508.8 s	705.9s	601.2s
Hailfinder	900.8s	1801.5 s	1084.1s

in the complexity of the algorithm. In addition, our proposed method significantly outperforms other techniques. The proposed technique has the following advantages:

- It reduces the computational complexity of a GA by considering a prior ordering of nodes as an input.
- It significantly avoids creating extra edges.

We also compare the time consumption by the node ordering techniques as input of GA. In this comparison, less time reflects a better performance. The results are presented in Table 3. It shows that our technique has the better performance.

Acknowledgements

The authors would like to thank the editor and reviewers for valuable comments and suggestions which considerably improved the quality of the presentation of the paper.

References

Abdi, H., and Williams, L. J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: *Computational Statistics*, **2**(4), 433-459.

Abramson, B., Brown, J., Edwards, W., Murphy, A., and Winkler, R. L. (1996). Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, **12**(1), 57-71.

Baesens, B., Egmont-Petersen, M., Castelo, R., and Vanthienen, J. (2002). Learning Bayesian network classifiers for credit scoring using Markov Chain Monte Carlo search. *Pattern Recognition* **3**, 49-52.

Beinlich, I. A., Suermondt, H. J., Chavez, R. M., and Cooper, G. F. (1989). *The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks* (pp. 247-256). Springer Berlin Heidelberg.

Bouckaert, R. R. (2001). Bayesian belief networks: from construction to inference.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate behavioral research*, **1**(2), 245-276.

Chen, X. W., Anantha, G., and Lin, X. (2008). Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm. *IEEE Transactions on Knowledge and Data Engineering*, **20**(5), 628-640.

Chickering, D. M. (1996). Learning Bayesian networks is NP-complete. *In Learning from data* (pp. 121-130). Springer New York.

Cliff, N. (1988). The eigenvalues-greater-than-one rule and the reliability of components.*Psychological bulletin*, **103**(2), 276.

Cooper, G. F., and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, **9**(4), 309-347.

De Campos, L. M. (2006). A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *The Journal of Machine Learning Research*, **7**, 2149-2187.

De Jong, K. A. (2006). *Evolutionary computation: a unified approach*. MIT press.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, **29**(2), 131-163.

Geiger, D., Verma, T., and Pearl, J. (1990). Identifying independence in Bayesian networks. *Networks*, **20**(5), 507-534.

Ghahramani, Z. (1998). Learning dynamic Bayesian networks. *In Adaptive processing of sequences and data structures* (pp. 168-197). Springer Berlin Heidelberg.

Grossman, D., and Domingos, P. (2004). Learning Bayesian network classifiers by maximizing conditional likelihood.*In Proceedings of the twenty-first international conference on Machine learning* (p. 46). ACM.

Heckerman, D. (1998).*A tutorial on learning with Bayesian networks* (pp. 301-354). Springer Netherlands.

Hruschka, E. R., and Ebecken, N. F. (2007). Towards efficient variables ordering for Bayesian networks classifier. *Data and Knowledge Engineering*, **63**(2), 258-269.

Jensen, F. V. (1996). *An introduction to Bayesian networks* (Vol. 210). London: UCL press.

Jolliffe, I. (2002). *Principal component analysis*. John Wiley and Sons, Ltd.

Kabli, R., Herrmann, F., and McCall, J. (2007). A chain-model genetic algorithm for Bayesian network structure learning. *In Proceedings of the 9th annual conference on Genetic and evolutionary computation* (pp. 1264-1271). ACM.

Larrañaga, P., Poza, M., Yurramendi, Y., Murga, R. H., and Kuijpers, C. M. (1996). Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,***18**(9), 912-926.

Lamma, E., Riguzzi, F., and Storari, S. (2005). Improving the K2 Algorithm Using Association Rule Parameters. *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU04)*, 1667-1674.

Margaritis, D. (2003). *Learning Bayesian network model structure from data* (Doctoral dissertation, US Army).

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pearl, J. (2011). *Bayesian networks*. Department of Statistics, UCLA.

Perrier, E., Imoto, S., and Miyano, S. (2008). Finding optimal Bayesian network given a super-structure. *Journal of Machine Learning Research*,**9**(2), 2251-2286.

Robinson, R. W. (1977). Counting unlabeled acyclic digraphs. *In Combinatorial mathematics*pp. 28-43. Springer Berlin Heidelberg.

Ruiz, C. (2005). *Illustration of the K2 algorithm for learning Bayes net structures*. Department of Computer Science.

Spirtes, P., and Meek, C. (1995). Learning Bayesian networks with discrete variables from data. *In KDD* **1**, 294-299.

Romero, T., Larrañaga, P., and Sierra, B. (2004). Learning Bayesian networks in the space of orderings with estimation of distribution algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, **18**(4), 607-625.

Wong, S. K. M., and Butz, C. J. (2000). A Bayesian approach to user profiling in information retrieval. *Technology Letters*, **4**(1), 50-56.

Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J., and Jarvis, E. D. (2004). Advances to Bayesian network inference for generating causal networks from observational biological data.*Bioinformatics*, **20**(18), 3594-3603.

Zwick, W. R., and Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain.*Psychological bulletin*, **99**(3), 432.