Optimization Letters manuscript No.
(will be inserted by the editor)

Structure Learning of Bayesian Networks using Global Optimization with Applications in Data Classification

Sona Taheri · **Musa Mammadov**

Received: date / Accepted: date

Abstract Bayesian Networks are increasingly popular methods of modeling uncertainty in artificial intelligence and machine learning. A Bayesian Network consists of a directed acyclic graph in which each node represents a variable and each arc represents probabilistic dependency between two variables. Constructing a Bayesian Network from data is a learning process that consists of two steps: learning structure and learning parameter. Learning a network structure from data is the most difficult task in this process. This paper presents a new algorithm for constructing an optimal structure for Bayesian Networks based on optimization. The algorithm has two major parts. First, we define an optimization model to find the better network graphs. Then, we apply an optimization approach for removing possible cycles from the directed graphs obtained in the first part which is the first of its kind in the literature. The main advantage of the proposed method is that the maximal number of parents for variables is not fixed a priory and it is defined during the optimization procedure. It also considers all networks including cyclic ones and then choose a best structure by applying a global optimization method. To show the efficiency of the algorithm, several closely related algorithms including unrestricted dependency Bayesian Network algorithm, as well as, benchmarks algorithms SVM and C4.5 are employed for comparison. We apply these algorithms on data classification; data sets are taken from the UCI machine learning repository and the LIBSVM. keywordsData ClassificationBayesian NetworksGlobal Optimization

S. Taheri
School of Science, Information Technology and Engineering, University of Ballarat, VIC 3353, Australia
E-mail: sonataheri@students.ballarat.edu.au
sona.taheri@unisa.edu.au

M. Mammadov
School of Science, Information Technology and Engineering, University of Ballarat, VIC 3353, Australia
National ICT Australia, VRL, VIC 3010, Australia
E-mail: m.mammadov@ballarat.edu.au
musa.mammadov@nicta.com.au

1 Introduction

Classification is a basic task in data mining that requires the construction of a classifier, that is, a function which assigns a class label to observations described by a set of feature variables. Learning accurate classifiers from preclassified data is a very active research topic in machine learning and artificial intelligence. One of the most effective classifiers is Bayesian Networks.

Bayesian Networks (BNs) are widely used representation frameworks for reasoning with probabilistic information [4,12,15,30,37]. These models use graphs to capture dependence and independence relationship between feature variables, allowing a concise representation of the knowledge as well as efficient graph based query processing algorithms. This representation is defined by two components: structure learning and parameter learning. The structure of this model represents a directed acyclic graph. The nodes in the graph correspond to the feature variables in the domain, and the arcs (edges) show the causal relationships between feature variables. A directed edge relates the variables so that the variable corresponding to the terminal node (child) will be conditioned on the variable corresponding to the initial node (parent). More incoming edges into a node result in a conditional probability of the corresponding variable with conjunctional condition containing all its parents. The parameter learning represents probabilities and conditional probabilities based on prior information or past experience. Once the network structure is constructed, the probabilistic inferences are readily calculated, and can be performed to predict the outcome of some variables based on observations of others. However, the problem of structure learning is a much more complex problem since the number of candidate structures grows exponentially when the number of features increases [32].

In recent years, the search for the structure of a BN able to reflect all existing relations of dependence in a data base has constituted a research topic of fundamental importance. Given a set of features and a data set composed of all features, the problem is to build a structure to present the connections among the features. This structure learning process needs to select the arcs between them, and therefore construct a network from data. Developing a structure is very useful for a variety of applications in general, for example, where there are masses of data available and we want to understand what underlies the knowledge or which features are correlated. In addition to providing a network that will allow us to predict behavior under conditions that we have not seen, the structure can also incorporate domain expert knowledge to provide more reliable suggestions. Nevertheless, there still remains the problem of building such a network structure. It is an important task, therefore, to develop some methods capable of learning a network structure directly from data.

Nowadays, the problem of learning structure of a BN based on optimization is receiving increasing attention within the community of researchers into uncertainty in artificial intelligence and machine learning. Various optimization problems for finding a structure of a BN have been defined. The papers [17,23,21,29] have presented new approaches based on the Genetic algorithm to find an optimal BNs' structure among alternative structures. The Simulated Annealing for structure learning in BNs have been studied, for example, in [14,35]. Application of the Particle Swarm optimization to discover better structures of BNs has been studied in [34,45]. In [27], the

Branch and Bound method has been applied for constructing a structure in a BN. The papers [3,7,16] have proposed BNs' structure learning algorithms based on the Ant Colony optimization.

More recently, we introduced an iterative unrestricted dependency algorithm for learning structure of Bayesian Networks for binary classification using a combinatorial optimization model [41]. Although this algorithm performs well and the results are promising, it does not involve all possible networks. The algorithm considers only acyclic networks and choose a network structure with the maximum training accuracy (equation (9) in [41]). However, there might be a cyclic network that results an optimal solution. In the present paper, we address this challenge by developing a new algorithm based on optimization approaches. The aim of optimization is to remove some edges in the cyclic networks to obtain acyclic ones. The final structure is a network with the highest accuracy among all proposed networks.

The algorithm is a general method for structure learning of BN which is used for multi-classification in the present work. It consists of two main parts. In the first part, we deal with an optimization model similar to that introduced in [41] to find better networks. Then, we apply optimization techniques to remove possible cycles to obtain an acyclic network structure. We consider two different cases for the second part. The first one is a simple case when we have a small number of cycles. In this case, we choose an optimal network from all possible combinations of removing some arcs in the existing cycles. In the second case, when we have a large number of cycles, we apply the global optimization algorithm AGOP introduced in [24,25] in conjunction with the recently developed local optimization algorithm CGN [39,40]. AGOP is an efficient algorithm in solving many difficult practical problems where objective functions were discontinuous [19,26] and even piecewise constant [42].

The paper is structured as follows: we briefly describe BNs in Section 2. In Section 3, we develop a new algorithm based on optimization for structure learning in BNs. In Section 4, we present some experimental results to compare the proposed algorithm with some well-known classification methods. Finally, Section 5 contains the concluding remarks.

2 Bayesian Networks

A BN is a directed acyclic graph containing nodes and edges and a set of conditional probability distributions. Suppose a set of variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where X_i denotes both the feature variable and its corresponding node. Let $Pa(X_i)$ stand for the set of parents of the node X_i as well as the feature variables corresponding to those parents. When there is an edge from X_i to X_j, then X_j is called the child variable for the parent variable X_i. A conditional dependency connects a child variable with a set of parent variables. The lack of possible edges encode conditional independencies.

Throughout this paper, we will refer to the collection of edges (arcs), the conditional dependence and independence relations among the variables, as the structure of BNs. In particular, given a structure, the joint probability distribution for $\mathbf{X}$ is given by

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i|Pa(X_i)). \quad (1)$$

However, the accurate estimation of $Pa(X_i)$ in the equation (1) is non trivial. Finding such estimation requires searching the space of all possible network structures for one that best describes the data. Structure learning algorithms determine for every possible edge in the network whether to include the edge in the final network and which direction to orient the edge. The number of possible graph structures grows super exponentially as every possible subset of edges could represent the final model. Due to this growth in graph structures, even a restricted form of structure learning has been proven to be an NP-hard problem [6, 13].

One of the restricted models in BNs is k-dependence BNs introduced by Sahami [33]. In this algorithm, each feature variable could have a maximum of k feature variables as parents, and these parents are obtained by using mutual information. The value of k is initially chosen before applying the k-dependence BNs, $k = 0, 1, 2, ...$ Naive Bayes (NB) [20] is a very simple form of this algorithm when $k = 0$. In the NB, feature variables are conditionally independent given the class. Although the NB is a very efficient method on a variety of data mining problems, the strong assumption that all features are conditionally independent given the class is often violated on many real world applications. Friedman et al. [10] introduced Tree Augment Naive Bayes (TAN). The TAN is a special form of the k-dependence BNs when $k = 1$. In the TAN, each feature variable has the class and at most one other feature variable as parents.

Although the mentioned methods were shown to be efficient, the features in these methods depend on the class and a priori given number of features; $k = 0$ dependence for the NB, $k = 1$ dependence for the TAN, and an priory given k for the k-dependence algorithm. In fact, by setting k, i.e., the maximum number of parent nodes that any feature may have, the final structure of BNs have been constructed. Since k is the same for all nodes, it is not possible to model cases where some nodes have a large number of dependencies, whereas others just have a few. We tried to solve this problem by introducing the unrestricted dependency BNs algorithm (UDBN) in [41], where the number k is defined by the algorithm internally. In this paper, we develop this idea further by proposing an optimization problem to eliminate possible cycles and therefore to learn an optimal structure of a BN.

3 A New Algorithm for Structure Learning in Bayesian Networks

In this section, we propose a new algorithm to learn an optimal structure of a Bayesian Network based on the approach developed in [41]. It uses a heuristic procedure where Bayesian Networks are build step-by-step by adding new possible links until the network obtained is acyclic. In this paper we extend this approach further by continuing that procedure and considering all possible cases including cyclic networks. The main question then is to make the network acyclic before building Bayesian Networks; that is, to eliminate some links. This paper suggests to use a criteria of deleting the smallest number of links that makes the network under consideration acyclic. Therefore, in

each step a combinatorial optimization problem is solved. As a result, a sequence of acyclic networks are generated by keeping as many as possible links in each. Therefore a network with maximum training accuracy is chosen as a final structure.

Below we describe the procedure used in [41]. Consider the following optimization model

$$\begin{aligned} &\text{Maximize } \sum_{\substack{i=1 \\ j \neq i}}^{n} \sum_{j=1}^{n} (K_{ij} - K) w_{ij}, \\ &\text{s.t. } w_{ij} \in \{0,1\}, \;\; w_{ij} + w_{ji} \leq 1, \;\; 1 \leq i, j \leq n, \;\; i < j. \end{aligned} \tag{2}$$

Given $1 \leq i, j \leq n, \;\; i \neq j$, the value w_{ij} is the association weight (to be found), defined by

$$w_{ij} = \begin{cases} 1 \text{ if feature } X_i \text{ is the parent of feature } X_j, \\ \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

and

$$K_{ij} = \sum_{\nu_2=1}^{|X_j|} \sum_{\nu_1=1}^{|X_i|} \max\{P(X_{\nu_2 j}|C_1, X_{\nu_1 i}), P(X_{\nu_2 j}|C_2, X_{\nu_1 i}), ..., P(X_{\nu_2 j}|C_q, X_{\nu_1 i})\}. \tag{4}$$

Here, $|X_j|$ and $|X_i|$ are the number of values of features X_j and X_i, respectively, and $X_{\nu l}$ shows the ν-th value of feature X_l, $1 \leq l \leq n$. The notations $C_1, C_2, ..., C_q$ stand for the class labels and K is a threshold such that $K \geq 0$.

From formula (2), $w_{ij} = 1$ if $K_{ij} > K_{ji}$ and $K_{ij} > K$, and therefore, $w_{ji} = 0$ due to the constraint $w_{ij} + w_{ji} \leq 1$. It is clear that $w_{ii} = 0, \;\; 1 \leq i \leq n$. Thus problem (2) can be solved easily.

Let us denote the solution of the problem (2) by $W(K) = [w_{ij}(K)]_{n \times n}$, where

$$w_{ij}(K) = \begin{cases} 1 \text{ if } K_{ij} > K_{ji} \text{ and } K_{ij} > K, \\ \\ 0 \text{ otherwise,} \end{cases} \tag{5}$$

and the set of arcs is represented by

$$A(W) = \{(i, j) : \; if \; w_{ij} = 1, \; 1 \leq i, j \leq n, \; i \neq j\}. \tag{6}$$

Here, (i, j) shows the arc from X_i to X_j. It is clear that $A(W) \subset I$, where

$$I = \{(i, j) : \; 1 \leq i, j \leq n\}$$

is the set of all possible couples (i, j).

We find the best value for K based on the maximum training accuracy for $W(K)$, defined by (5), that is a solution to (2). K is in the interval $[0, K^{max}]$ where

$$K^{max} = max\{K_{ij} : \; 1 \leq i, j \leq n, \; i \neq j\}. \tag{7}$$

We will consider different values $K = K_r \doteq K^{max} - \varepsilon r, \;\; r = 0, 1, ...$ until $K_r < 0$. Given K_r, let $W(K_r) = [w_{ij}(K_r)]_{n \times n}$ be the matrix defined by (5).

With the matrix $W(K_r)$, the set of arcs $A(W(K_r))$ and, therefore, a network will be learnt.

Since the network structure is constrained to be acyclic, we should not have any cycle in the network obtained by $A(W(K_r))$. Because of this limitation, in [41] the above procedure of considering different values $K_r = K^{max} - \varepsilon r,\ r = 0,1,...$ is terminated when the network obtained is cyclic. In this paper we continue this procedure until $K_r < 0$ and construct acyclic networks by eliminating some links in each step. This allows us to consider a significantly larger set of networks when searching the best one.

Suppose we have m cycles in the network found by $A(W(K_r))$, and consider

$$\mathscr{C}(W(K_r)) = \cup_{l=1}^{m} \mathscr{C}_l(W(K_r))$$

where $\mathscr{C}_l(W(K_r))$ denotes the set of arcs which makes l-th cycle, $l = 1,2,...,m$. Clearly $\mathscr{C}_l(W(K_r)) \subset A(W(K_r))$ for all $l = 1,...,m$, and $w_{ij} = 1$ if $(i,j) \in \mathscr{C}_l(W(K_r))$. We also define $\overline{\mathscr{C}}(W(K_r)) = I \setminus \cup_{l=1}^{m} \mathscr{C}_l(W(K_r))$.

Consider the set of all arcs that are at least in one cycle in the network obtained by $A(W(K_r))$. Let $\overline{m}$ be the number of arcs in this set. Accordingly, we define

$$V(K_r) = \{v_{r_1}, v_{r_2}, ..., v_{r_{\overline{m}}}\}, \tag{8}$$

where v represents an arc, $r \in \{r_1, r_2, \cdots, r_{\overline{m}}\}$ is related to an arc (i_r, j_r) that belongs to some cycle in the network obtained by $A(W(K_r))$ and $v_r = w_{i_r j_r} = 1$.

The aim is to delete a minimal number of arcs to have an acyclic structure. Deleting existing arcs in (8) means setting 0 to some $v_{r_i},\ l = 1,...,\overline{m}$. We apply an optimization procedure to existing arcs in cycles (8). We utilize two different methods.

1. The first one is a simple case that can be used if the number $\overline{m}$ is small. In this case, we can consider all the possible combinations of deleting arcs in the existing cycles. Let us denote by

$$\mathbb{V} = \{V_s(K_r),\ s = 1,2,...,\rho\}, \tag{9}$$

the set of all possible combinations of $\overline{m}$ dimensional vectors $V_s(K_r)$ with values 0 and 1. Clearly $\rho = 2^{\overline{m}}$. Then we chose a vector $V^* = (v^*_{r_1}, v^*_{r_2}, ..., v^*_{r_{\overline{m}}})$ that has a maximal norm $\|V^*\|$ provided that the corresponding network is acyclic.

2. We consider continuous variables $(\mathfrak{v}_{\mathfrak{r}_1}, \mathfrak{v}_{\mathfrak{r}_2}, ..., \mathfrak{v}_{\mathfrak{r}_{\overline{\mathfrak{m}}}})$ with $\mathfrak{v}_{\mathfrak{r}_i} \in [0,1],\ i = 1,...,\overline{m}$, to formulate an optimization problem.

Denote by B a binary transformation given by $B(\mathfrak{v}_{\mathfrak{r}_1}, \mathfrak{v}_{\mathfrak{r}_2}, ..., \mathfrak{v}_{\mathfrak{r}_{\overline{\mathfrak{m}}}}) = (\widetilde{\mathfrak{v}}_{\mathfrak{r}_1}, \widetilde{\mathfrak{v}}_{\mathfrak{r}_2}, ..., \widetilde{\mathfrak{v}}_{\mathfrak{r}_{\overline{\mathfrak{m}}}})$, where for $i = 1,...,\overline{m}$

$$\widetilde{\mathfrak{v}}_{\mathfrak{r}_i} = \begin{cases} 0 \text{ if } \mathfrak{v}_{\mathfrak{r}_i} \leq \frac{1}{2}, \\ \\ 1 \text{ if } \mathfrak{v}_{\mathfrak{r}_i} > \frac{1}{2}. \end{cases} \tag{10}$$

We denote by $\gamma(B(\mathfrak{v}_{\mathfrak{r}_1}, \mathfrak{v}_{\mathfrak{r}_2}, ..., \mathfrak{v}_{\mathfrak{r}_{\overline{\mathfrak{m}}}}))$ the number of cycles in the corresponding structure. Clearly, for large $\overline{m}$, the number ρ will grow exponentially and therefore

searching an acyclic network by considering all the possible combinations with the maximal norm $\|V\|$ will be impossible. In this case, we generate an optimization problem involving variables $\mathfrak{v}_{\mathfrak{r}_i}$, $i = 1,...,\overline{m}$, as follows:

$$\text{Maximize} \quad \sum_{i=1}^{\overline{m}} \left((\mathfrak{v}_{\mathfrak{r}_i} - \frac{1}{2})^2 + \zeta \mathfrak{v}_{\mathfrak{r}_i}\right) - \mu\gamma(B(\mathfrak{v}_{\mathfrak{r}_1}, \mathfrak{v}_{\mathfrak{r}_2}, ..., \mathfrak{v}_{\mathfrak{r}_{\overline{m}}})), \quad \text{s.t. } \mathfrak{v}_{\mathfrak{r}_i} \in [0,1], \ \forall\, i. \tag{11}$$

Here $\zeta \in (0, \frac{1}{2})$ and μ is a penalty parameter assigned to the number of cycles.

Problem (11) attempts to find an acyclic network with the largest number of arcs. We apply algorithm AGOP and CGN to solve problem (11). Let $(\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{m}}})$ be a global optimal solution to (11). The proposition below shows that it is a binary vector. Therefore, we can set $V^* = (v^*_{r_1}, v^*_{r_2}, ..., v^*_{r_{\overline{m}}}) = (\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{m}}})$.

Proposition: Let $(\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{m}}})$ is a global optimal solution to problem (11). Then it is a binary vector; that is, $\mathfrak{v}^*_{\mathfrak{r}_i} \in \{0,1\}, \forall\, i$; and the corresponding structure is acyclic: $\gamma(B(\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{m}}})) = 0$.

Proof: The fact $\gamma(B(\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{m}}})) = 0$ is a direct result of applying a large penalty parameter μ; thus, the corresponding structure is acyclic. Now we show that the vector $(\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{m}}})$ is binary.

Take any $1 \le i \le \overline{m}$, and denote $x = \mathfrak{v}^*_{\mathfrak{r}_i}$. For the sake of simplicity, let $i = 1$. After fixing all other elements $\mathfrak{v}^*_{\mathfrak{r}_j}$, $j \ne i$, we obtain

$$\psi(x) = \varphi(x) + \lambda, \tag{12}$$

where

$$\varphi(x) = (x - \frac{1}{2})^2 + \zeta x, \tag{13}$$

and

$$\lambda = \sum_{j \ne i} \left((\mathfrak{v}_{\mathfrak{r}_j} - \frac{1}{2})^2 + \zeta \mathfrak{v}_{\mathfrak{r}_j}\right) - \mu\gamma(B(x, \mathfrak{v}_{\mathfrak{r}_2}, ..., \mathfrak{v}_{\mathfrak{r}_{\overline{m}}})). \tag{14}$$

By assumption $x = \mathfrak{v}^*_{\mathfrak{r}_1}$ is a global maximum of $\psi(x)$. On the contrary, assume that $\mathfrak{v}^*_{\mathfrak{r}_1}$ is not binary.

It is clear that the function $\varphi(x)$ has one minimum at $\overline{x} = \frac{1-\zeta}{2}$. Moreover, $\varphi(0) = \frac{1}{4}$ and since $0 < \zeta < \frac{1}{2}$, we have

$$\varphi(\frac{1}{2}) = \frac{1}{2}\zeta < \frac{1}{4}.$$

Therefore, $\varphi(x)$ has a unique maximizer $x^* = 0$ on the interval $[0, \frac{1}{2}]$; and has a unique maximizer $x^{**} = 1$ on the interval $[0,1]$.

Now, if $\mathfrak{v}^*_{\mathfrak{r}_1} \in (\frac{1}{2}, 1)$, then taking $x = 1$ from (10) we have

$$B(1, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{\mathfrak{m}}}}) = B(\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{\mathfrak{m}}}})$$

and, therefore, $\gamma(B(1, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{\mathfrak{m}}}})) = 0$. Then $\psi(1) > \psi(\mathfrak{v}^*_{\mathfrak{r}_1})$. This contradicts to the assumption that $\mathfrak{v}^*_{\mathfrak{r}_1}$ is a global maximizer of $\psi(x)$.

On the other hand, if $\mathfrak{v}^*_{\mathfrak{r}_1} \in (0, \frac{1}{2}]$, then taking $x = 0$ from (10) we have $B(0, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{\mathfrak{m}}}}) = B(\mathfrak{v}^*_{\mathfrak{r}_1}, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{\mathfrak{m}}}})$ and, therefore, $\gamma(B(0, \mathfrak{v}^*_{\mathfrak{r}_2}, ..., \mathfrak{v}^*_{\mathfrak{r}_{\overline{\mathfrak{m}}}})) = 0$. Then $\psi(0) > \psi(\mathfrak{v}^*_{\mathfrak{r}_1})$ that is again a contradiction. □

Once the acyclic network structure for a specific r is found, the corresponding training accuracy is estimated. Based on the highest training accuracy, the value for r and, therefore, the best value for K_r is chosen.

According to explanations above, we present the following algorithm for learning an optimal structure of a BN, and we call it Algorithm OpBN.

4 Numerical Experiments

The experimental work is carried out using 22 benchmark data sets taken from the UCI machine learning repository [1] and the tools page of the LIBSVM [5]. These data sets have been considered quite frequently in the literature. A brief description of the data sets is given in Table 1.

We use three different methods to discretize the continuous features. In the first one, we apply a mean value of each feature to discretize values to binary, $\{0, 1\}$. In the second one, we apply the Fayyad and Irani's discretization method [9]. The third one is Algorithm SOAC [44]; the parameter θ in this algorithm is chosen as 0.2. This parameter has not been fitted by preliminary experimentation, and is similar to the one used for other problems in [44].

We conduct experiments to compare the proposed algorithm (OpBN) with the most relevant methods in which these methods has the number of parents priory given; the Naive Bayes (NB), the Tree Augmented Naive Bayes (TAN), the k-Dependency Bayesian Networks (k-DBN), $k = 2$. We also compare the present algorithm with our previous work [41], the unrestricted dependency BNs algorithm (UDBN) to show the efficiency of applying optimization techniques to the BNs. Some benchmark classifiers like the SVM and the C4.5 have been chosen to show the proposed algorithm is comparable with these well known methods. In all cases we use 10-fold cross validation. Runs with the various classifiers were carried out on the same training sets and evaluated on the same test sets. In particular, the cross validation folds are the same for all experiments on each data set. We used Weka machine learning software for C4.5 and SVM with radial basis kernel.

In calculations, we take $\eta = 10^{-3}$, $\vartheta = 1.1$, $\delta = 10^{-3}$, $\omega = 10^{-10}$, $\varpi = 10^{10}$ for the CGN based on [40] and we set $\mu = 10^3$, $\varepsilon = 0.1$, $\rho_0 = 2^{10}$ for the proposed algorithm.

Algorithm3. Algorithm OpBN

Step 1. Compute $\{K_{ij},\ 1 \le i,j \le n,\ i \ne j\}$ using (4).

Step 2. Determine K^{max} using (7), and set $r = 0$.

Step 3. While $K_r = K^{max} - \varepsilon r \ge 0$:
3.1. Compute $\{w_{ij}(K_r),\ 1 \le i,j \le n,\ i \ne j\}$, using (5). Set $w_{ij}(K_r) = 0$ for $i = j$, and let $W(K_r) = [w_{ij}(K_r)]_{n\times n}$.

3.2. Find the set of arcs $A(W(K_r))$ using (6).

3.3. Apply the topological traversal algorithm [2,11] to detect possible cycles in the network obtained by $A(W(K_r))$. If there is no cycle, then calculate the training accuracy, $accuracy(A(W(K_r)))$; set $r = r+1$ and go back to Step 3.

3.4. Apply the DFS algorithm [36,43] to determine a vector $V(K_r)$ in (8).

3.5. Find $\mathbb{V}$, using (9), and determine ρ. If $\rho > \rho_0$ go to 3.10.

3.6. For $s = 1,2,...,\rho$, check the network obtained by $A(\overline{W}_s(K_r))$ for any possible cycle, using the topological traversal algorithm, where $\overline{W}_s(K_r) = [\overline{w}_{ij}(K_r)]_{n\times n}$, and

$$\overline{w}_{ij}(K_r) = \begin{cases} w_{ij}(K_r) & \text{if } (i,j) \in \overline{\mathscr{C}}(W(K_r)), \\ v_{r_\tau} & \text{if } (i_{r_\tau}, j_{r_\tau}) \in \mathscr{C}(W(K_r)). \end{cases}$$

3.7. Set $\widetilde{\mathbb{V}} = \{\widetilde{V}_{\widetilde{s}}(K_r),\ \widetilde{s} = 1,2,...,\widetilde{\rho}\}$ including those vectors from the set $\mathbb{V}$ that are acyclic, and $\widetilde{\rho} \le \rho$.

3.8. Let $\widehat{\mathbb{V}} = \{\widehat{V}_{\widehat{s}}(K_r),\ \widehat{s} = 1,2,...,\widehat{\rho}\}$ combines all vectors in the set $\widetilde{\mathbb{V}}$ having maximum norm. Clearly $\widehat{\rho} \le \widetilde{\rho}$ and often there are several vectors with the same maximum norm; that is $\widehat{\rho}$ might be greater than 1.

3.9. Find the maximum training accuracy, $accuracy(A(\widehat{W^*}(K_r)))$ between the network structures obtained by $\widehat{W}_{\widehat{s}}(K_r),\ \widehat{s} = 1,2,...,\widehat{\rho}$ corresponding to $\widehat{V}_{\widehat{s}}(K_r)$ and set $W(K_r) = \widehat{W^*}(K_r)$; set $r = r+1$ and go back to Step 3.

3.10. Solve the optimization problem (11) by applying algorithm AGOP; denote the solution found by $V'(K_r)$. Then apply algorithm CGN starting from this solution to obtain a vector $V^*(K_r)$. After this optimization procedure we create corresponding matrix $W^*(K_r)$. Set $W(K_r) = W^*(K_r)$, and find the new acyclic network structure by a set of arcs $A(W(K_r))$ using (6).

3.11. Compute the training accuracy, $accuracy(A(W(K_r)))$; set $r = r+1$ and go back to Step 3.

Step 4. Find the best K_r^* where $accuracy(A(W(K_r^*)))$ has the maximum value among the training accuracies, $accuracy(A(W(K_r))),\ r = 1,2,....$

Step 5. Return an optimal acyclic structure using a set of arcs $A(W(K_r^*))$.

4.1 Results

In this section, we present accuracies obtained with the proposed algorithm OpBN. We compare the OpBN by means of the predictive accuracies obtained with some well-known classifiers such as the NB, the TAN, the k-DBN, the UDBN, the SVM, and the C4.5. The predictive accuracy of each method is the percentage of test sets for which it predicts the class correctly. The predictive accuracies, for each classifier in each data set, are summarized in Tables 2 to 4, where continuous features are discretized by using mean values, the Fayyad and Irani's method and discretization algorithm SOAC, respectively. Since the UDBN is an algorithm proposed for binary classification, we do not have the accuracy results for multi class data sets.

Figure 1 shows the scatter plots comparing the proposed algorithm with the NB, the TAN, the k-DBN, the UDBN, the SVM, and the C4.5 on different data sets using the Fayyad and Irani's method discretization method. In these plots, each point represents a data set, where the x coordinate of a point is the percentage of miss classifications according to the proposed algorithm, and the y coordinate is the percentage of miss classification according to the chosen classifier for comparison. Therefore, points above the diagonal line correspond to data sets where the proposed algorithm performs better, and points below the diagonal line correspond to data sets where the chosen classifier performs better.

Tables 2 to 4 demonstrate the efficiency of the proposed algorithm when we compare it to some other well known methods. This algorithm has not only the advantage of finding the number of parents for each node internally during the optimization process but also outperforms other listed methods in these tables in terms of the accuracy. The test set accuracies of the proposed algorithm (OpBN) are significantly higher than the NB, the TAN and the k-DBN in all data sets using different discretization methods for continuous features. Compared to the UDBN, it has better accuracies in the majority of data sets; In 21 cases out of 22 , using mean values for discretization and the algorithm SOAC, the OpBN shows greater accuracies than the UDBN and also higher in 20 cases out of 22 data sets where continuous features are discretized by applying the Fayyad and Irani's method (FaI). It is also notable that the proposed algorithm has greater accuracies than the benchmark classifiers in the most of data sets and almost ties in few ones.

The comparison of the time complexity of our method with the others is not considered in this paper as different platforms were used to conduct the results. We have used Matlab to create code for our method and the UDBN, Fortran for the NB, the TAN and the k-DBN and Weka machine learning was used for the SVM and the C4.5. But we include running time for the proposed algorithm when we apply it to the Diabetes data set in the next subsection.

4.2 Dynamic structures generated by OpBN

As mentioned above the main advantage of the proposed method is that it does not set the number of parents a priory. This number comes from the optimization procedure; it might be different for different folds on the same data. To demonstrate this we consider two examples; the first one is the Diabetes data set with 8 features (see Table 1). In the Table 5, we demonstrate structures obtained by the proposed algorithm for different folds. Four different structures obtained when applying 10-folds cross validation. For instance, parents of the feature X_7 are X_3, X_4, X_5 for folds 1 and 5-10 and X_4, X_5 for folds 2 and 3. This feature (X_7) does not have any parent in the structure obtained for the fold 4. The average computational time of the proposed algorithm using 10-folds cross validation to find an optimal structure of this data set is 22 seconds.

The second one is the Ionosphere data set with 34 features (see Table 1) when we have quite large number of features. In the Table 6, we provide structures of the Ionosphere data set in the folds 1 and 10. We only include features with parents in

this table and skip those without parents. For example, in the fold 10, feature X_{33} has the parents X_6, X_{10}, X_{20} and X_{34} doesn't have any and therefore it is not in the table. The average running time employed by the proposed algorithm over 10-folds cross validation to find an optimal structure of the Ionosphere data set is 569 seconds.

In the Table 7, we include the number of cycles and the length of each one in the networks obtained for these data sets for different values of r in one fold before the application of optimization techniques. This table shows that the Diabetes data set has no cycles for $r = 0, 1, 2, 3$. They start from $r = 4$ and for $r > 9$, the number of cycles and their length are the same. For the data set Ionosphere, there is no cycle when $r < 6$ starting for $r = 6$ and repeating the same ones after $r > 15$. Note that, given any r, the steps $3.5 - 3.10$ (global optimization phase) of the algorithm OpNB, eliminate all these cycles and produce an acyclic network by removing a minimal number of arcs.

5 Conclusion

In this paper, a new algorithm has been proposed to learn an optimal structure of a Bayesian Network. The algorithm consists of two major parts in which in the first part a combinatorial optimization model has been constructed to determine the better structures and, then, an optimal structure is obtained using an optimization approach to remove possible cycles from the initial graphs obtained in the first part. The number of parents for each node is determined along the process of the algorithm when we apply optimization approaches. This is the main advantage of the proposed algorithm to some well known BN models with the given number prior to the algorithm, but in reality it may vary for each data sets. This algorithm also choose an optimal network from all possible networks including cyclic ones by applying optimization techniques which is the main improvement of our previous work, unrestricted dependency Bayesian Networks.

Some benchmark data sets, from the UCI machine learning repository and the LIBSVM, are used to evaluate the effectiveness of the proposed algorithm on data classification and to compare its performance with some commonly used classifiers. The obtained results indicate that the new algorithm outperforms all the other mentioned algorithms for accuracy. An interesting aspect of the proposed algorithm and its learning method is that it discovers unrestricted edges between nodes (dependencies between features) which is common in real life data sets.

References

1. Asuncion, A. and Newman, D. UCI machine learning repository. School of Information and Computer Science, University of California http://www.ics.uci.edu/mlearn/MLRepository.html. (2007)
2. Bender, M. A. , Fineman, J. T. Gilbert, S, A New Approach to Incremental Topological Ordering, Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms Society for Industrial and Applied Mathematics Philadelphia, PA, USA, (2009)
3. Campos, L., Fernandez-Luna, M. , Gamez, A. , Puerta, M., Ant colony optimization for learning Bayesian networks. International Journal of Approximate Reasoning, 291-311 (2002)

Table 1 A brief description of data sets.

Data sets	# Observations	# Features	# Classes
Breast Cancer	699	10	2
Congres Vote	435	16	2
Credit Approval	690	15	2
Diabetes	768	8	2
German.numer	1000	24	2
Glass Identification	214	10	7
Haberman Survival	306	3	2
Heart Disease	303	14	2
Hepatitis	155	19	2
Image Segmentation	2310	19	7
Ionosphere	351	34	2
Iris	150	4	3
Letter Recognition	20000	16	26
Liver Disorders	345	6	2
Lymphography	148	18	4
Sonar	208	60	2
Soybean-Large	307	35	19
Spambase	4601	57	2
Svmguide1	7089	4	2
Svmguide3	1284	21	2
Vehicle Silhouettes	946	18	4
Waveform-21	5000	21	3

Table 2 Average predictive accuracy over 10 fold cross validation for 22 data sets using mean value for discretization.

Data sets	Bayesian Network Classifiers					Benchmark Classifiers	
	NB	TAN	k-DBN	UDBN	OpBN	SVM	C4.5
Breast Cancer	96.10	95.71	97.31	97.66	**97.89**	95.15	91.06
Congress Vote	90.31	91.42	94.62	95.48	**96.93**	96.02	95.51
Credit	84.95	82.88	86.87	87.46	**88.14**	85.31	87.47
Diabetes	75.90	76.48	75.03	75.98	**77.81**	76.72	75.98
German	75.41	74.13	76.35	76.27	**78.30**	76.11	72.43
Glass	69.40	68.95	69.64	—	**73.74**	71.14	69.35
Haberman	75.01	73.85	76.43	77.86	**78.92**	73.34	71.60
Heart Disease	81.12	84.12	84.27	84.71	**84.85**	80.14	81.53
Hepatitis	83.61	83.14	84.12	85.25	**86.08**	83.61	82.97
Image Seg	90.65	85.01	91.08	—	93.10	89.35	**93.13**
Ionosphere	82.90	84.02	88.35	89.98	**90.21**	85.94	86.20
Iris	95.66	95.66	95.66	—	**96.11**	95.66	95.66
Letter	64.65	73.01	73.91	—	86.98	82.10	**87.41**
Liver	61.86	61.89	62.22	64.17	**65.73**	60.16	60.79
Lymphography	79.71	66.89	71.43	—	86.24	**86.31**	77.12
Sonar	75.18	75.44	75.61	76.89	**78.92**	76.98	76.65
Soybean	91.08	92.02	92.27	—	**94.28**	93.52	91.85
Spambase	90.03	89.69	89.27	92.37	**94.12**	90.17	91.89
Svmguide1	92.57	91.99	92.98	94.17	**97.09**	93.24	93.62
Svmguide3	81.51	83.04	83.64	85.41	**85.41**	80.16	81.24
Vehicle	59.15	68.70	68.91	—	**76.18**	73.98	69.99
Waveform-21	76.98	75.12	75.86	—	85.51	**85.59**	74.51

Table 3 Average predictive accuracy over 10 fold cross validation for 22 data sets using discretization method FaI.

Data sets	Bayesian Network Classifiers					Benchmark Classifiers	
	NB	TAN	k-DBN	UDBN	OpBN	SVM	C4.5
Breast Cancer	97.18	96.52	96.92	97.72	**97.98**	96.52	94.11
Congress Vote	90.11	93.21	94.73	95.12	**96.71**	95.04	95.32
Credit	86.10	84.78	86.44	87.21	**88.93**	85.03	84.87
Diabetes	74.56	75.14	75.12	75.85	**77.84**	75.51	73.83
German	74.50	73.13	76.32	76.27	**79.82**	76.41	71.92
Glass	69.63	69.15	69.84	—	**74.30**	71.50	69.58
Haberman	75.09	74.41	76.89	77.91	**79.18**	73.20	71.24
Heart Disease	82.93	81.23	83.45	85.14	**85.31**	81.67	82.85
Hepatitis	84.56	83.91	83.90	85.17	**86.87**	85.16	83.87
Image	91.15	85.31	91.18	—	93.58	89.52	**93.62**
Ionosphere	88.62	89.77	89.83	91.10	**92.62**	89.67	89.98
Iris	95.87	95.87	95.87	—	**96.11**	95.87	95.87
Letter	64.93	73.41	73.86	—	**87.68**	82.22	**87.68**
Liver	63.26	63.18	64.17	65.91	**66.86**	62.03	62.15
Lymphography	79.70	66.85	76.34	—	**87.94**	86.48	77.01
Sonar	76.32	76.47	76.49	76.74	**79.35**	77.96	77.31
Soybean	91.19	92.10	92.52	—	**94.12**	93.85	91.97
Spambase	90.41	89.78	89.39	93.18	**93.18**	90.43	92.97
Svmguide1	92.39	91.61	92.76	94.45	**97.22**	94.31	95.99
Svmguide3	81.23	82.47	83.23	84.42	**84.42**	80.37	81.38
Vehicle	58.27	67.85	67.88	—	**77.32**	74.34	72.45
Waveform-21	77.87	75.35	76.71	—	86.50	**86.68**	74.68

Table 4 Average predictive accuracy over 10 fold cross validation for 22 data sets using discretization Algorithm SOAC.

Data sets	Bayesian Network Classifiers					Benchmark Classifiers	
	NB	TAN	k-DBN	UDBN	OpBN	SVM	C4.5
Breast Cancer	96.12	95.60	96.76	97.65	**97.94**	95.31	91.16
Congress Vote	90.11	91.42	92.61	94.16	**96.97**	96.75	95.12
Credit	85.85	84.98	86.53	87.17	**89.11**	86.11	87.54
Diabetes	75.78	75.90	75.82	76.22	**78.02**	76.68	75.63
German	74.61	74.01	75.31	76.15	**79.14**	76.35	72.21
Glass	69.52	69.02	69.76	—	**73.84**	71.62	69.46
Haberman	74.66	76.08	75.64	77.31	**79.24**	73.36	72.15
Heart Disease	78.62	77.37	79.54	81.69	**83.46**	77.96	79.17
Hepatitis	82.93	81.54	84.21	85.93	**86.12**	84.24	82.34
Image	91.37	85.51	91.24	—	93.41	89.47	**93.72**
Ionosphere	85.92	86.18	85.94	88.62	**90.23**	86.15	86.71
Iris	93.43	93.42	94.11	—	**95.36**	94.18	94.18
Letter	64.80	73.71	73.98	—	87.34	82.41	**87.71**
Liver	65.82	65.73	65.95	65.97	**66.81**	63.69	64.98
Lymphography	79.76	66.95	71.81	—	86.34	**86.73**	77.11
Sonar	75.09	75.76	75.85	76.91	**79.31**	77.74	76.41
Soybean	91.21	92.15	92.31	—	**94.79**	93.81	91.99
Spambase	89.30	89.04	90.69	92.54	93.26	91.56	**93.73**
Svmguide1	95.81	94.91	96.32	97.54	**97.91**	95.94	96.91
Svmguide3	77.25	79.99	80.75	82.92	**82.92**	78.32	78.49
Vehicle	62.23	69.97	69.78	—	**75.24**	73.81	72.88
Waveform-21	76.98	74.58	75.64	—	**88.78**	86.12	74.06

Table 5 Parents of feature X_i in the data set 'Diabetes' obtained by Algorithm 'OpBN'.

X_i	Parents(X_i)			
	Folds: 1, 9, 10	Folds: 2, 3	Fold: 4	Folds: 5, 6, 7, 8
X_1	X_3-X_5,X_7	X_3-X_5,X_7	X_3-X_5,X_7	X_3-X_5,X_7
X_2	X_1,X_3-X_8	X_1,X_3-X_8	X_1,X_3-X_8	X_1,X_3-X_8
X_3	X_4,X_5	X_4,X_5,X_7	X_4,X_5,X_7	X_4,X_5
X_4	*no parent*	*no parent*	X_7	*no parent*
X_5	X_4	X_4	X_4,X_7	X_4
X_6	X_1,X_3-X_5,X_7,X_8	X_1,X_3-X_5,X_7,X_8	X_1,X_3-X_5,X_7	X_1,X_3-X_5,X_7
X_7	X_3-X_5	X_4,X_5	*no parent*	X_3-X_5
X_8	X_1,X_3-X_5,X_7	X_1,X_3-X_5,X_7	X_1,X_3-X_7	X_1,X_3-X_7

Table 6 Parents of feature X_i in the data set 'Ionosphere' obtained by Algorithm 'OpBN'; no parents for missing X_i's

	Fold 1		Fold 10
X_i	Parents(X_i)	X_i	Parents(X_i)
X_1	X_{21}	X_1	X_6
X_5	X_{32}	X_4	X_{13},X_{15}
X_7	X_{14}	X_5	X_{16}
X_8	$X_{13},X_{15},X_{17},X_{19},X_{25}$	X_7	X_{16}
X_{10}	X_9,X_{19},X_{25}	X_8	$X_{15},X_{19},X_{21},X_{29}$
X_{12}	X_{13}	X_9	$X_6,X_{16},X_{18},X_{27},X_{30},X_{32},X_{34}$
X_{14}	X_5,X_{13},X_{31},X_{33}	X_{10}	X_{15},X_{21},X_{29}
X_{15}	X_{32}	X_{11}	X_{14}
X_{16}	$X_{15},X_{19},X_{21},X_{23},X_{25},X_{31}$	X_{12}	X_{15}
X_{18}	X_3,X_{29}	X_{13}	X_{12},X_{16},X_{26}
X_{22}	X_4	X_{14}	X_6
X_{27}	X_{12},X_{21}	X_{15}	$X_3,X_6,X_{16},X_{18},X_{22},X_{24},X_{27},X_{28},X_{30},X_{32},X_{34}$
		X_{19}	X_3,X_6
		X_{21}	$X_3,X_6,X_{16},X_{27},X_{34}$
		X_{29}	$X_3,X_6,X_{16},X_{18},X_{22},X_{27},X_{34}$
		X_{33}	X_6,X_{10},X_{20}

Table 7 Number of cycles and length of each cycle for Diabetes and Ionosphere for different r.

r	Diabetes		Ionosphere	
	No.Cycles	Length of Cycles	No.Cycles	Length of Cycles
0-3	0	-	0	-
4	1	3	0	-
5	1	3	0	-
6	2	3, 5	3	3, 5, 3
7	3	3, 5, 6	5	3, 5, 3, 4, 3
8	5	3, 5, 6, 7, 4	6	3, 5, 3, 4, 3, 7
9	11	3, 5, 6, 7, 4 , 6, 3, 4, 5, 6, 4	7	3, 5, 3, 4, 3, 7, 10
10			7	3, 5, 3, 4, 3, 7, 10
11			7	3, 5, 3, 4, 3, 7, 10
12			8	3, 5, 3, 4, 3, 7, 10, 8
13			9	3, 5, 3, 4, 3, 7, 10, 8, 4
14			10	3, 5, 3, 4, 3, 7, 10, 8, 4, 11
15			15	3, 5, 3, 4, 3, 7, 10, 8, 4, 11, 7, 6, 8, 10, 9

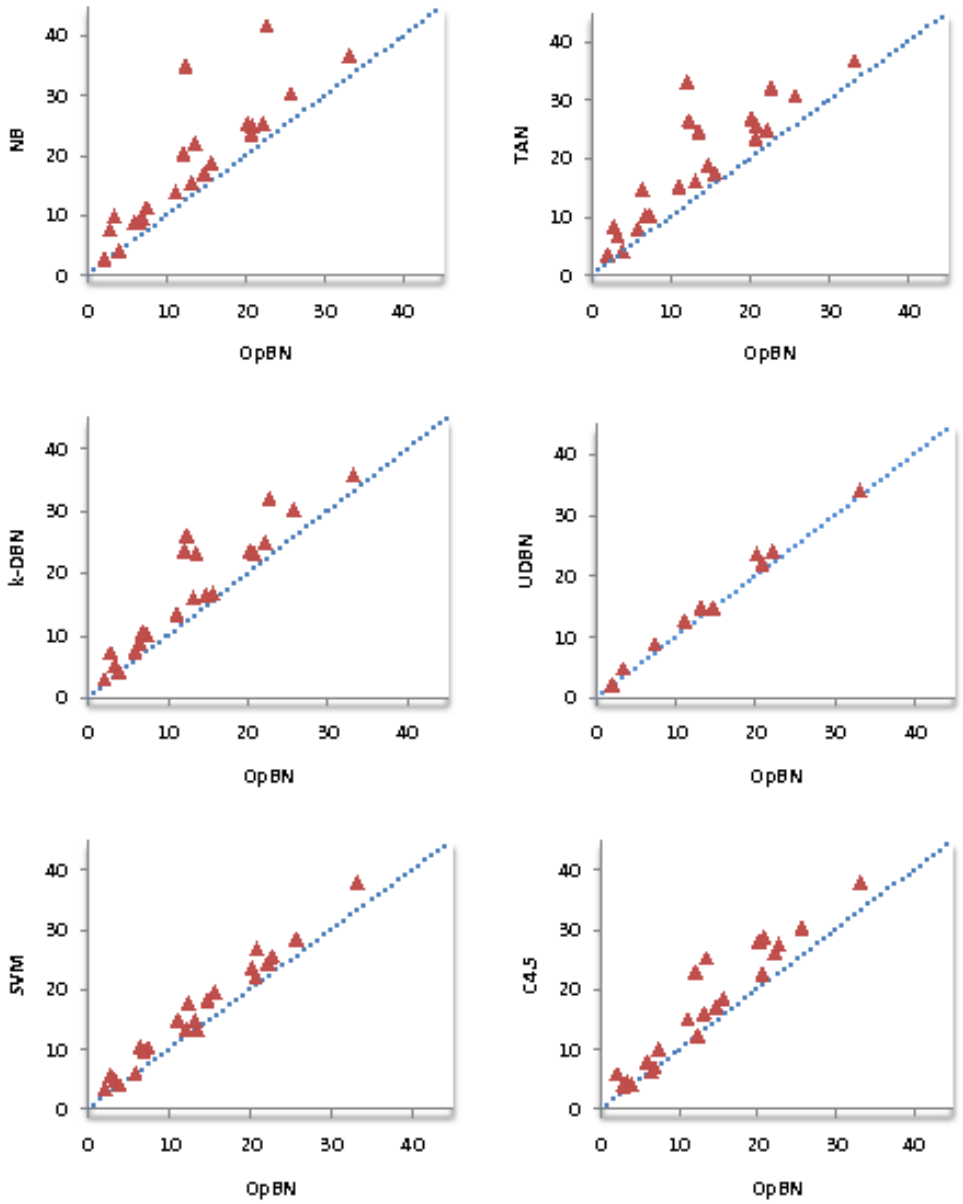

Fig. 1 Scatter plots comparing miss classifications of the proposed algorithm (x coordinate) with competing methods (y coordinate); using the discretization method FaI

4. Castillo, E., Gutierrez, J.M and Hadi, A.S. Expert Systems and Probabilistic Network Models. Springer Verlag, New York. (1997)
5. Chang, C, and Lin, C. LIBSVM: A library for support vector machines, 2001a. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm. (2001)
6. Chickering, D. M., Learning Bayesian Networks is NP-complete, Articial Intelligence and statistics V, Springer, 121-130 (1996)
7. Daly, R., , Shen, Q., Learning Bayesian Network Equivalence Classes with Ant Colony Optimization, Journal of Articial Intelligence Research, Elsevier, 391-447 (2009)
8. Domingos, P. , Pazzani, M., On the optimality of the simple Bayesian classier under zero-one loss, Machine Learning, 103-130 (1997)
9. Fayyad, U. M , Irani, K. B., On the Handling of Continuous-Valued Attributes in Decision Tree Generation, Machine Learning, 87-102 (1993)
10. Friedman, N. , Geiger, D. , Goldszmidt, M., Bayesian network classiers, Machine Learning, 131-163 (1997)
11. Haeupler, B. , Kavitha, T. , Mathew, R. , Sen, S. , Tarjan, R. E., Incremental Cycle Detection, Topological Ordering, and Strong Component Maintenance, 35th International Colloquium on Automata, Languages, and Programming (ICALP), Reykjavik, Iceland (2008)
12. Heckerman, D. , Geiger, D. , Chickering, D.M., Learning Bayesian Networks: the Combination of Knowledge and Statistical Data, Machine Learning, 197-243 (1995)
13. Heckerman, D. , Chickering, D. , Meek, C., Large-Sample Learning of Bayesian Networks is NP-Hard, Machine Learning, 1287-1330 (2004)
14. Janzura, M., Nielsen, J., A Simulated Annealing-Based Method for Learning Bayesian Networks from Statistical Data, International Journal of Intelligent Systems, 335-348 (2006)
15. Jensen, F., An Introduction to Bayesian Networks, Springer, New York (1996)
16. Ji, Z. , Zhong, H. , Hu. R. , Liu, C., Bayesian Network Learning Algorithm Based on Independence Test and Ant Colony Optimization, Acta Automatica Sinica (2009)
17. Kabli, R. , Herrmann, F. , McCall, J., A Chain-Model Genetic Algorithm for Bayesian Network Structure Learning, Proceedings of the 9th annual conference on Genetic and evolutionary computation, ACM New York, NY, USA (2007)
18. Kolda, G. , Lewis, M. , Torczon, V., Optimization by direct search: New perspectives on some classical and modern methods, SIAM Review, 385-482 (2003)
19. Kouhbor, S. , Ugon, J. , Rubinov, A. , Kruger, A. , Mammadov, M., Coverage in WLAN with minimum number of access points, Vehicular Technology Conference, VTC Spring, 1166-1170 (2006)
20. Langley, P. , Iba, W. , Thompson, K., An Analysis of Bayesian Classiers, In 10th International Conference Articial Intelligence, AAAI Press and MIT Presspp, 223-228 (1992)
21. Larranaga, P. , Murga, R. , Poza, M. , Kuijpers, C., Structure Learning of Bayesian Networks by Hybrid Genetic Algorithms, Spriger-Verlag, 165-174 (1996)
22. Larranaga, P. , Poza, M. , Yurramendi, Y. , Murga, H. , Kuijpers, C., Structure Learning of Bayesian Networks by Genetic Algorithms: A Performance Analysis of Control Parameters, IEEE Transactions on Pattern Analysis and Machine Intelligence archive (1996)
23. Larranaga, P. , Sierra, B. , Gallego, J. , Michelena, J. , Picaza. M., Learning Bayesian Networks by Genetic Algorithms: A case study in the prediction of survival in malignant skin melanoma, Articial Intelligence in Medicine, 261-272 (1997)
24. Mammadov, M.A. , Rubinov A.M , Sniedovich, M., A New Global Optimization Algorithm Based on Dynamical Systems Approach, 6th International Conference on Optimization: Techniques and Applications, Ballarat, Australia (2004)
25. Mammadov, M.A. , Rubinov A.M , Yearwood, J., Dynamical Systems Described by Relational Elasticities with Applications to Global Optimization, 6th In Continuous Optimisation: Current Trends and Modern Applications, V.Jeyakumar and A. Rubinov, Eds. Springer, 365-385 (2005)
26. Mammadov, M.A. , Orsi, R., H-innity via a nonsmooth, nonconvex optimization approach, Pacic Journal of Optimization, 405-420 (2005)
27. Marinescu, R. , Dechter, R., AND/OR Branch-and-Bound search for combinatorial optimization in graphical models, Articial Intelligence, Elsevier, 1457-1491 (2009)
28. Maroosi, A. , Amiri, B., A new clustering algorithm based on hybrid global optimization based on a dynamical systems approach algorithm, Expert Systems with Applications, 5645-5652 (2010)
29. Park, H. , Cho, S., An Efcient Attribute Ordering Optimization in Bayesian Networks for Prognostic Modeling of the Metabolic Syndrome, Springer-Verlag Berlin Heidelberg, 381-391 (2006)
30. Pearl, j., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann (1988)

31. Richter, F. , Fettweis, G., Base Station Placement Based on Force Fields, IEEE VTC-Spring, Yokohama, Japan (2012)
32. Robinson, R. W., Counting unlabeled acyclic diagraphs, Springer-Verlag, New York, 28-43 (1997)
33. Sahami, M., Learning Limited Dependence Bayesian Classiers, In the 2nd International Conference. Knowledge Discovery and Data mining (KKD), 335-338 (1996)
34. Sahin, F. , Yavuz, M. , Arnavut, Z. , Uluyol, O., Fault diagnosis for airplane engines using Bayesian networks and distributed particle swarm optimization, Parallel Computing, Elsevier, 124-143 (2007)
35. Schleip, C. , Rais, A. , Menzel, A., Bayesian analysis of temperature sensitivity of plant phenology in Germany, Agricultural and Forest Meteorology, Elsevier, 1699-1708 (2009)
36. Sedgewick, R., Graph algorithms, Addison-Wesley, ISBN 0-201-06672-6 (1983)
37. Shafer, G. , Pearl, j., Readings in Uncertain Reasoning, Morgan Kaufmann, San Mateo, CA (1990)
38. Sun, W. , Yuan, Y.X., Optimization theory and methods, nonlinear programming, Springer (2006)
39. Taheri, S. , Mammadov, M., Solving Systems of Nonlinear Equations using a Globally Convergent Optimization Algorithm, Global Journal of Technology and Optimization, 132-138 (2012)
40. Taheri, S., Mammadov, M. Seifollahi, S., Globally Convergent Optimization Methods for Unconstrained Problems, Optimization, 124-143 (2012)
41. Taheri, S. , Mammadov, M., Structure Learning of Bayesian Networks using a New Unrestricted Dependency Algorithm, Second International Conference on Social Eco-Informatics, Venice, Italy (2012)
42. Tilakaratne, C.D. , Mammadov, M. , Morris, S.A., Modied neural network algorithms for predicting trading signals of stock market indices, J. Appl. Mathe. Decis. Sci (2009)
43. Tucker, A., Covering Circuits and Graph Coloring, Applied Combinatorics (5th ed). Hoboken: John Wiley and sons (2006)
44. Yatsko, A. , Bagirov, A. , Stranieri, A., On the Discretization of Continuous Features for Classication, In the proceedings of Ninth Australasian Data Mining Conference (AusDM 2011), Ballarat, Australia (2011)
45. Zhao, J. , Sun, J. , Xu, W. , Zhou, D., Structure Learning of Bayesian Networks Based on Discrete Binary Quantum-behaved Particle Swarm Optimization Algorithm, Proceedings of the Fifth International Conference on Natural Computation, IEEE Computer Society Washington, DC, USA (2009)