Structure Learning of Bayesian Networks Using a New Unrestricted Dependency Algorithm

Sona Taheri
Centre for Informatics and Applied Optimization
School of Science, Information Technology and Engineering
University of Ballarat, VIC 3353, Australia
Email: sonataheri@students.ballarat.edu.au

Musa Mammadov
University of Ballarat, VIC 3353, Australia
Email: m.mammadov@ballarat.edu.au
National ICT Australia, VRL, VIC 3010, Australia
Email: musa.mammadov@nicta.com.au

***Abstract*—Bayesian Networks have deserved extensive attentions in data mining due to their efficiencies, and reasonable predictive accuracy. A Bayesian Network is a directed acyclic graph in which each node represents a variable and each arc a probabilistic dependency between two variables. Constructing a Bayesian Network from data is the learning process that is divided in two steps: learning structure and learning parameter. In many domains, the structure is not known a priori and must be inferred from data. This paper presents an iterative unrestricted dependency algorithm for learning structure of Bayesian Networks for binary classification problems. Numerical experiments are conducted on several real world data sets, where continuous features are discretized by applying two different methods. The performance of the proposed algorithm is compared with the Naive Bayes, the Tree Augmented Naive Bayes, and the $k-$Dependency Bayesian Networks. The results obtained demonstrate that the proposed algorithm performs efficiently and reliably in practice.**

Keywords-Data Mining; Bayesian Networks; Naive Bayes; Tree Augmented Naive Bayes; $k-$Dependency Bayesian Networks; Topological Traversal Algorithm.

I. INTRODUCTION

Data Mining is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [6]. The whole process of data mining consists of several steps. Firstly, the problem domain is analyzed to determine the objectives. Secondly, data is collected and an initial exploration is conducted to understand and verify the quality of the data. Thirdly, data preparation is made to extract relevant data sets from the database. A suitable data mining algorithm is then employed on the prepared data to discover knowledge represented in different representations such as decision trees, neural networks, support vector machine and Bayesian Networks. Finally, the result of data mining is interpreted and evaluated. If the discovered knowledge is not satisfactory, these steps will be iterated. The discovered knowledge is then applied in decision making. Recently, there is an increasing interest in discovering knowledge represented in Bayesian Networks [13], [14], [17], [15], [19] and [28]. Bayesian networks (BNs), introduced by Pearl [21], can encode dependencies among all variables; therefore, they readily handle situations where some data entries are missing. BNs are also used to learn causal relationships, and hence can be used to gain understanding about a problem domain and to predict the consequences of intervention. Moreover, since BNs in conjunction with Bayesian statistical techniques have both causal and probabilistic semantics, they are an ideal representation for combining prior knowledge and data [10]. In addition, BNs in conjunction with Bayesian statistical methods offer an efficient and principal approach for avoiding the over fitting of data [20]. BNs have been applied widely for data mining, causal modeling and reliability analysis [29].

This paper presents a novel unrestricted dependency algorithm to learn knowledge represented in BNs from data. A BN is a graphical representation of probability distributions over a set of variables that are used for building a structure of the problem domain. The BN defines a network structure and a set of parameters, class probabilities and conditional probabilities. 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 the observations of others.

The main task of learning BNs from data is finding directed arcs between variables, or, in other words, the structure discovery, which is the more challenging, and thus, more interesting phase. Two rather distinct approaches have been used widely to structure discovery in BNs: the constraint-based approach [22], [27] and the score-based approach [1], [5], [26]. In the the constraint-based approach, structure learning cares about whether one arc in the graph should be existed or not. This approach relies on the conditional independence test to determine the importance of arcs [4]. In the score-based approach, several candidate graph structures are known, and we need choosing the best one out. In order to avoid over fitting, investigators often use model selection methods, such as Bayesian scoring function [5] and entropy-based method [12]. Several exact algorithms based on dynamic programming have recently been developed to learn an optimal BN [16], [24], [25] and [31]. The main idea in these algorithms is to solve small subproblems first and

ISBN: 978-1-61208-227-1

use the results to find solutions to larger problems until a global learning problem is solved. However, they might be inefficient due to their need to fully evaluate an exponential solution space.

It has been proved that learning an optimal BN is NP-hard [11]. In order to avoid the intractable complexity for learning BNs, the Naive Bayes [18] has been used. The Naive Bayes (NB) is the simplest among BNs. In the NB, features are conditionally independent given the class. It has shown to be very efficient on a variety of data mining problems. However, the strong assumption that all features are conditionally independent given the class is often violated on many real world applications. In order to relax this assumption of the NB while at the same time retaining its simplicity and efficiency, researchers have proposed many effective methods [7], [23] and [28]. Sahami [23] proposed the $k-$dependence BNs to construct the feature dependence with a given number, value of k. In this algorithm, each feature could have a maximum of k feature variables as parents, and these parents are obtained by using mutual information. The value of k in this algorithm is initially chosen before applying it, $k = 0, 1, 2, ...$ Friedman et al. [7] introduced the Tree Augment Naive Bayes (TAN) based on the tree structure. It approximates the interactions between features by using a tree structure imposed on the NB structure. In the TAN, each feature has the class and at most one other feature 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 initially chosen k for the k-dependence BNs. In fact, by setting k, i.e., the maximum number of parent nodes that any feature may have, we can construct the structure of BNs. 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. In this paper, we propose a new algorithm to identify the limitations of each of these methods while also capturing much of the computational efficiency of the NB. In the proposed algorithm, the number k is defined by the algorithm internally, and it is an unrestricted dependency algorithm.

The rest of the paper is organized as follows. In the next section, we provide a brief description of BNs. In Section III, we introduce a new algorithm for structure learning of BNs from binary classification data. Section IV presents a brief review of the Topological Traversal algorithm. The results of numerical experiments are given in Section V. Section VI concludes the paper.

II. Representation of Bayesian Networks

A BN consists of a directed acyclic graph connecting each variables into a network structure and a collection of conditional probability tables, where each variable in the graph is denoted by a conditional probability distribution given its parent variables. The nodes in the graph correspond to the variables in the domain, and the arcs (edges) between nodes represent causal relationships among the corresponding variables. The direction of the arc indicates the direction of causality. When two nodes are joined by an arc, the causal node is called the parent of the other node, and another one is called the child. How one node influences another is defined by conditional probabilities for each node given its parents [21]. Suppose a set of variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, where X_i denotes both the variable and its corresponding node. Let $Pa(X_i)$ denotes a set of parents of the node X_i in $\mathbf{X}$. When there is an edge from X_i to X_j, then X_j is called the child variable for a parent variable X_i. A conditional dependency connects a child variable with a set of parent variables. The lack of possible edges in the structure encodes conditional independencies.

In particular, given a structure, the joint probability distribution for $\mathbf{X}$ is given by

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

here, $P(X_i|Pa(X_i))$ is the conditional probability of X_i given its parents $Pa(X_i)$, where

$$P(X_i|Pa(X_i)) = \frac{P(X_i, Pa(X_i))}{P(Pa(X_i))} = \frac{n_{X_i,Pa(X_i)}}{n_{Pa(X_i)}},$$

where $n_{Pa(X_i)}$ denotes the number of items in the set $Pa(X_i)$, and $n_{X_i,Pa(X_i)}$ is the number of items in $X_i \cap Pa(X_i)$.

However, accurate estimation of $P(X_i|Pa(X_i))$ is non trivial. Finding such an estimation requires searching the space of all possible network structures for one that best describes the data. Traditionally, this is done by employing some search mechanism along with an information criterion to measure goodness and differentiate between candidate structures met while traversing the search space. The idea would be to try and maximize this information measure or score by moving from one structure to another. The associated structure is then chosen to represent and explain the data. Finding an optimal structure for a given set of training data is a computationally intractable problem. 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 exponentially as every possible subset of edges could represent the final model. Due to this exponential growth in graph structure, learning an optimal BNs has been proven to be NP-hard [11].

During the last decades a good number of algorithms whose aim is to induce the structure of the BN that better represents the conditional dependence and independence relationships underlying have been developed [4], [5], [7], [12], [16], [24] and [25]. In our opinion, the main reason for continuing the research in the structure learning problem is that mendelizing the expert knowledge has become an expensive, unreliable and time consuming job. We introduce a new algorithm for structure learning of BNs in the following section.

III. The Proposed Algorithm for Bayesian Networks

In this section, we propose a new algorithm to learn the structure of BNs for binary classification problems. Since the learning process in BNs is based on the correlations of children and parent nodes, we propose a combinatorial optimization model to find the dependencies between features. However, some features could be independent which is considered by intruding a threshold K. Let us consider an optimization model (2):

$$\max \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (K_{ij} - K) w_{ij}, \tag{2}$$

$$\text{subject to} \quad w_{ij} + w_{ji} \leq 1,$$

where $1 \leq i,j \leq n,\ i < j$ and $w_{ij} \in \{0,1\}$. w_{ij} is the association weight (to be found), given 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 for $1 \leq i,j \leq n, \quad i \neq j$,

$$K_{ij} = \sum_{q_2=1}^{|X_j|} \sum_{q_1=1}^{|X_i|} \max\{P(X_{q_2 j}|C_1, X_{q_1 i}), P(X_{q_2 j}|C_2, X_{q_1 i})\}. \tag{4}$$

Here, $|X_j|$ and $|X_i|$ are the number of values of features X_j and X_i, respectively, and X_{ql} shows the qth value of the feature $X_l,\ 1 \leq l \leq n$. We assume binary classification; $C_1 = 1$ and $C_2 = -1$ are class labels. K is a threshold such that $K \geq 0$.

From the 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 presented by

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

(i,j) shows the arc from X_i to X_j. If we have set of arcs $A(W)$, then we have the corresponding matrix W that satisfies (6). 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).

The best value for K will be found based on the maximum training accuracy for different values of $w_{ij}(K)$, where $0 \leq K \leq K^{max}$, and

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

More precisely, we find the values of $w_{ij}(K_r)$ for different $K_r = K^{max} - \varepsilon r,\ r = 0, 1, ...$ until $K_r < 0$, and we set $W(K_r) = [w_{ij}(K_r)]_{n \times n}$. With the matrix $W(K_r)$, the set of arcs $A(W(K_r))$ and, therefore, a network will be learnt. Based on the obtained network, the conditional probabilities will be found:

$$P(C|\mathbf{X}) \equiv \prod_{i=1}^{n} P(X_i|C, Pa(X_i))P(C), \tag{8}$$

where $Pa(X_i)$ denotes the set of parents of the variable X_i to be found with $W(K_r)$. Now, based on these conditional probabilities, we calculate:

$$C(\mathbf{X}) = \begin{cases} 1 & \text{if } P(C_1 = 1|\mathbf{X}) > P(C_2 = -1|\mathbf{X}), \\ \\ -1 & \text{otherwise,} \end{cases}$$

and then the maximum training accuracy will be found using the following formula:

$$\textit{accuracy}(A(W(K_r))) = \frac{100}{ntr} \sum_{i=1}^{ntr} \delta(C(\mathbf{X}_i), C_i), r = 0, 1, ... \tag{9}$$

where

$$\delta(\alpha, \beta) = \begin{cases} 1 & \text{if } \alpha = \beta \\ \\ 0 & \text{otherwise.} \end{cases}$$

We will choose that value of r corresponding to the highest training accuracy. Here, ntr stands for the number of instances in the training set.

Since BNs are directed acyclic graphs, we should not have any cycle in the structure obtained by $A(W(K_r))$. Therefore, the maximum training accuracy subject to no cycles will give the best value of K_r, denoted by K^*, and consequently, the best structure $A(W(K^*))$. Here, we apply the topological traversal algorithm to test if the corresponding graph to the obtained network is acyclic.

According to explanations above, the proposed algorithm constructs unrestricted dependencies between features based on the structure of the NB. The proposed algorithm eliminates the strong assumptions of independencies between features in the NB, yet at the same time maintains its robustness. It is clear that $r = 0$ in the proposed algorithm gives the structure of the NB. In our algorithm, some features could have a large number of dependencies, whereas others just have a few. The number of these dependencies will be defined by the algorithm internally. The steps of our algorithm is presented in the following:

```
Step 1. Compute $\{K_{ij},\ 1 \le i,j \le n,\ i \ne j\}$ using (4).
Step 2. Determine $K^{max}$ using (7). Set $r = 0$, and $p_0 = 0$.
Step 3. while $K^{max} - \varepsilon r \ge 0$ do
    3.1. Calculate $K_r = K^{max} - \varepsilon r$.
    3.2. Compute $w_{ij}(K_r),\ 1 \le i,j \le n,\ (i \ne j)$ using (5), and let $W(K_r) = [w(K_r)_{ij}]_{n \times n}$.
    3.3. Find dependencies between features by a set of arcs $A(W(K_r))$ using (6).
    3.4. Apply the topological traversal algorithm to test the network obtained by $A(W(K_r))$ for possible cycles. If any cycle is founds, then go to Step 4.
    3.5. Compute the training accuracy, $p = accuracy(A(W(K_r))$, using (9). If $p > p_0$ then set $p_0 = p$, $K^* = K_r$, $r = r + 1$.
end
Step 4. Construct the optimal structure based on the basic structure of the NB and applying the set of arcs $A(W(K^*))$ between features.
Step 5. Compute the conditional probability tables inferred by the new structure.
```

Algorithm 1: Unrestricted Dependency BNs Algorithm

In this paper, we limit ourselves to binary classification, though a brief discussion on multiple class classification is warranted. The most straightforward approach in these classification problems is finding maximum of m conditional probabilities in the formula (4), where m is the number of classes. Moreover, the one-versus-all classification paradigm will be used to find either in the training accuracy, (9), or the test accuracy in the experiments.

IV. Topological Traversal Algorithm

The topological traversal algorithm [8] is applied for testing a directed graph if there exists any cycle. The degree of a node in a graph is the number of connections or edges the node has with other nodes. If a graph is directed, meaning that edges point in one direction from one node to another node. Then nodes have two different degrees, the in-degree, which is the number of incoming edges to this node, and the out-degree, which is the number of outgoing edges from this edge.

The topological traversal algorithm begins by computing the in-degrees of the nodes. At each step of the traversal, a node with in-degree of zero is visited. After a node is visited, the node and all the edges emanating from that node are removed from the graph, reducing the in-degree of adjacent nodes. This is done until the graph is empty, or no node without incoming edges exists. The presence of the cycle prevents the topological order traversal from completing. Therefore, the simple way to test whether a directed graph is cyclic is to attempt a topological traversal of its nodes. If all nodes are not visited, the graph must be cyclic.

V. Experiments

We have employed 12 well-known binary classification data sets. A brief description of the data sets is given in Table I. The detailed description of the data sets used in this experiments are downloadable in the UCI repository of machine learning databases [2] and the tools page of the LIBSVM [3]. The reason that we have chosen these data sets is: they are the most frequently binary classification data sets considered in the literature.

All continue features in data sets are discretized using two different methods. In the first one, we apply a mean value of each feature to discretize values to binary, $\{0, 1\}$. In the second one, we use the discretization algorithm using sub-optimal agglomerative clustering (SOAC) [30] to get more than two values for discretized features.

We conduct an empirical comparison for the Naive Bayes (NB), the Tree Augmented Naive Bayes (TAN), the $k-$Dependency Bayesian Networks ($k-$DBN), and the proposed algorithm (UDBN) in terms of test set accuracy. We have compared our algorithm with the mentioned algorithms because the basic structure of all, the TAN, the $k-$DBN and the UDBN, is based on the the structure of the NB. In all the cases we have used $10-$fold cross validation. We report the averaged accuracy over the ten test folds.

Table II presents the averaged test set accuracy obtained by the NB, the TAN, the $k-$DBN and the UDBN on 12 data sets, where continuous features are discretized using mean values for discretization. The results presented in this table demonstrate that the accuracy of the proposed algorithm (UDBN) is much better than that of the NB, and the TAN in all data sets. The UDBN also works better than the $k-$DBN in most of data sets. In 10 cases out of 12, the UDBN has higher accuracy than the $k-$DBN. The accuracy of this method almost ties with the $k-$DBN in data sets Phoneme CR and German.numer.

The averaged test set accuracy obtained by the NB, the TAN, the $k-$DBN and the UDBN on 12 data sets using discretization algorithm SOAC summarized in Table III. The results from this table show that the accuracy obtained by the proposed algorithm in all data sets are higher than those obtained by the NB, the TAN, and the $k-$DBN.

According to the results explained, the proposed algorithm, UDBN, works well. It yields good classification compared to the NB, the TAN and the $k-$DBN. In addition, our algorithm is more general than the $k-$DBN. In the $k-$DBN, the number k is a priori chosen. In fact, by setting k, i.e., the maximum number of parent nodes that any feature may have, the structure of BNs could be 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. However, in the proposed algorithm, the number k is defined by the algorithm internally, and it is an unrestricted dependency algorithm. It might be various for different data sets, and even for each fold in the calculations. The computational times are not presented in Tables II and III. It is clear that the proposed algorithm needs more computational time than the others, since for example, the NB appears as a special case of UDBN when $r = 0$.

Table I
A BRIEF DESCRIPTION OF DATA SETS

Data sets	# Instances	# Features
Breast Cancer	699	10
Congressional Voting Records	435	16
Credit Approval	690	15
Diabetes	768	8
Haberman's Survival	306	3
Ionosphere	351	34
Phoneme CR	5404	5
Spambase	4601	57
Fourclass	862	2
German.numer	1000	24
Svmguide1	7089	4
Svmguide3	1284	21

Table II
TEST SET ACCURACY AVERAGED OVER $10-$FOLD CROSS VALIDATION FOR DATA SETS USING MEAN VALUES FOR DISCRETIZATION. NB STANDS FOR NAIVE BAYES, TAN FOR TREE AUGMENTED NAIVE BAYES, $k-$DBN FOR $k-$DEPENDENCY BAYESIAN NETWORKS, $k = 2$, AND UDBN FOR THE PROPOSED ALGORITHM

Data sets	NB	TAN	$k-$DBN	UDBN
Breast Cancer	97.18	96.52	97.31	97.66
Congressional Voting Records	90.11	93.21	94.62	95.48
Credit Approval	86.10	84.78	86.87	87.46
Diabetes	74.56	75.14	75.03	75.98
Haberman's Survival	75.09	74.41	76.43	77.86
Ionosphere	88.62	89.77	88.35	89.98
Phoneme CR	77.56	78.31	80.58	80.16
Spambase	90.41	89.78	89.27	92.37
Fourclass	77.46	77.61	77.94	79.06
German.numer	74.50	73.13	76.35	76.27
Svmguide1	92.39	91.61	92.98	94.17
Svmguide3	81.23	82.47	83.64	85.41

Table III
TEST SET ACCURACY AVERAGED OVER $10-$FOLD CROSS VALIDATION FOR DATA SETS USING DISCRETIZATION ALGORITHM SOAC. NB STANDS FOR NAIVE BAYES, TAN FOR TREE AUGMENTED NAIVE BAYES, $k-$DBN FOR $k-$DEPENDENCY BAYESIAN NETWORKS, $k = 2$, AND UDBN FOR THE PROPOSED ALGORITHM

Data Sets	NB	TAN	$k-$DBN	UDBN
Breast Cancer	96.12	95.60	96.76	97.65
Congressional Voting Records	90.11	91.42	92.61	94.16
Credit Approval	85.85	84.98	86.53	87.17
Diabetes	75.78	75.90	75.82	76.22
Haberman's Survival	74.66	73.78	75.64	77.31
Ionosphere	85.92	86.18	85.94	88.62
Phoneme CR	77.01	78.53	80.41	81.01
Spambase	89.30	89.04	90.69	92.54
Fourclass	78.58	79.52	78.97	79.96
German.Numer	74.61	74.01	75.31	76.15
Svmguide1	95.61	94.91	96.32	97.54
Svmguide3	77.25	79.99	80.75	82.92

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new algorithm for learning of the structure in Bayesian Networks. An important property of this algorithm is adding some numbers of arcs between features that captures unrestricted dependency among them. The number of arcs has been defined by the proposed algorithm internally. We have carried out a number of experiments on some binary classification data sets from the UCI machine learning repository and LIBSVM. The values of features in data sets are discritized by using mean value of each feature and applying discretization algorithm using sub-optimal agglomerative clustering. We have presented results of numerical experiments. These results clearly demonstrate that the proposed algorithm achieves comparable or better performance in comparison with traditional Bayesian Networks.

Our future work is applying the proposed algorithm to more complicated problems for learning BNs, e.g., problems with incomplete data, hidden variables, and multi class data sets.

REFERENCES

[1] H. Akaike, *Analysis of Cross Classified Data by AIC.* Ann. Inst. Statist. pp. 185-197, 1978.

[2] A. Asuncion and D. Newman, *UCI machine learning repository.* School of Information and Computer Science, University of California, 2007.

http://www.ics.uci.edu/mlearn/MLRepository.html, accessed May 2012.

[3] C. Chang and C. Lin, *LIBSVM: A library for support vector machines*. http://www.csie.ntu.edu.tw/cjlin/libsvm, accessed May 2012.

[4] J. Cheng, D. Bell, and W. Liu, *Learning Belief Networks from Data*. An Information Theory Based Approach. Artificial Intelligence, 137. pp. 43-90, 2002.

[5] G. F. Cooper and E. Herskovits, *A Bayesian Method for Constructing Bayesian Belief Networks from Databases*. Conference on Uncertainty in AI. pp. 86-94, 1990.

[6] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, *From data mining to knowledge discovery: An overview*. In Advances in Knowledge Discovery in Data Mining. AAAI Press, Menlo Park, CA. pp. 1-34, 1996.

[7] N. Friedman, D. Geiger, and M. Goldszmidt, *Bayesian network classifiers*. Machine Learning 29. pp. 131-163, 1997.

[8] B. Haeupler, T. Kavitha, R. Mathew, S. Sen, and R. E. Tarjan, *Incremental Cycle Detection, Topological Ordering, and Strong Component Maintenance*. 35th ACM Transactions on Algorithms (TALG). pp. 1-33, 2012.

[9] D. Heckerman, A. Mamdani, and W. Michael, *Real-World Applications of Bayesian Networks*. Communications of the ACM. pp. 38-68, 1995.

[10] D. Heckerman, *Bayesian Networks for Data Mining*. Data Mining and Knowledge Discovery1. pp. 79-119, 1997.

[11] D. Heckerman, D. Chickering, and C. Meek, *Large-Sample Learning of Bayesian Networks is NP-Hard*. Journal of Machine Learning Research. pp. 1287-1330, 2004.

[12] E. Herskovits and G. F. Cooper, *An Entropy-Driven System for Construction of Probabilistic Expert Systems from Databases*. 6th Internatial Conference on Uncertainty in Artificial Intelligence (UAI90), Cambridge, MA, USA, Elsevier Science, New York. pp. 54-62, 1991.

[13] Y. Jing, V. Pavlovic, and J. Rehg, *Efficient discriminative learning of Bayesian network classifier via Boosted Augmented Naive Bayes*. The 22 nd International Conference on Machine Learning, Bonn, Germany. pp. 369 - 376, 2005.

[14] A. Jonsson and A. Barto, *Active Learning of Dynamic Bayesian Networks in Markov Decision Processes*. Springer-Verlag Berlin Heidelberg. pp. 273284, 2007.

[15] D. Kitakoshi, H. Shioya, and R. Nakano, *Empirical analysis of an on-line adaptive system using a mixture of Bayesian networks*. Information Sciences, Elsevier. pp. 2856-2874, 2010.

[16] M. Koivisto and K. Sood, *Exact Bayesian structure discovery in Bayesian networks*. Journal of Machine Learning 5. pp. 549573, 2004.

[17] P. Kontkanen, T. Silander, T. Roos, and P. Myllymki, *Bayesian Network Structure Learning Using Factorized NML Universal Models*. Information Theory and Applications Workshop (ITA-08), IEEE Press. pp. 272 - 276, 2008.

[18] P. Langley, W. Iba, and K. Thompson, *An Analysis of Bayesian Classifiers*. In 10th International Conference Artificial Intelligence, AAAI Press and MIT Press. pp. 223-228, 1992.

[19] W. Liaoa and Q. Ji, *Learning Bayesian network parameters under incomplete data with domain knowledge*. Pattern Recognition, Elsevier. pp. 3046-3056, 2009.

[20] P. Myllymaki, *Advantages of Bayesian networks in data mining and knowledge discovery*. http://www.bayesit.com/docs/advantages.html, 2005, acessed April 2012.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Networks of Plausible Inference, Morgan Kaufmann, 1988.

[22] J. Pearl, *Causality: Models, Reasonings and Inference*. Cambridge University Press. pp. 675685, 2003.

[23] M. Sahami, *Learning Limited Dependence Bayesian Classifiers*. In the 2nd International Conference. Knowledge Discovery and Data mining (KKD). pp. 335-338, 1996.

[24] T. Silander and P. Myllymaki, *A simple approach for finding the globally optimal Bayesian network structure*. In Proceedings of UAI-06. pp. 445-452, 2006.

[25] A. Singh and A. Moore, *Finding optimal Bayesian networks by dynamic programming*. Technical Report CMU-CALD-05-106, Carnegie Mellon University, 2005.

[26] G. Schwarz, *Estimating the Dimension of a Model*. Annals of Stastics. pp. 461-464, 1978.

[27] P. Spirtes, C. Glymour, and R. Sheines, *Causation, Prediction, and Search*. 1993.

[28] S. Taheri, M. Mammadov, and A. M. Bagirov, *Improving Naive Bayes Classifier Using Conditional Probabilities*. In the proceedings of Ninth Australasian Data Mining Conference (AusDM), Ballarat, Australia. Vol. 125. pp. 63-68, 2011.

[29] P. Weber, G. Medina-Oliva, C. Simon, and B. Iung, *Overview on Bayesian networks applications for dependability*. Risk-analysis and maintenance areas, Engineering Applications of Artificial Intelligence. pp. 671-682, 2010.

[30] A. Yatsko, A. M. Bagirov, and A. Stranieri, *On the Discretization of Continuous Features for Classification*. In the proceedings of Ninth Australasian Data Mining Conference (AusDM 2011), Ballarat, Australia. Vol. 125, 2011.

[31] C. Yuan, B. Malone, and X. Wu, *Learning Optimal Bayesian Networks Using A* Search*. In the proceedings of twenty second international joint conferenceon Artificial intelligence. pp. 2186-2191, 2011.