Research Article

A Bayesian Classifier Learning Algorithm Based on Optimization Model

Sanyang Liu, Mingmin Zhu, and Youlong Yang

Department of Mathematics, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Sanyang Liu; liusanyang@126.com

Received 6 September 2012; Accepted 10 December 2012

Academic Editor: Cesar Cruz-Hernandez

Copyright © 2013 Sanyang Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Naive Bayes classifier is a simple and effective classification method, but its attribute independence assumption makes it unable to express the dependence among attributes and affects its classification performance. In this paper, we summarize the existing improved algorithms and propose a Bayesian classifier learning algorithm based on optimization model (BC-OM). BC-OM uses the chi-squared statistic to estimate the dependence coefficients among attributes, with which it constructs the objective function as an overall measure of the dependence for a classifier structure. Therefore, a problem of searching for an optimal classifier can be turned into finding the maximum value of the objective function in feasible fields. In addition, we have proved the existence and uniqueness of the numerical solution. BC-OM offers a new opinion for the research of extended Bayesian classifier. Theoretical and experimental results show that the new algorithm is correct and effective.

1. Introduction

With the development of information technology, in particular the progress of network technology, multimedia technology and communication technology, massive data analysis, and processing become more and more important. Since Bayesian network as classifier has a solid mathematical basis and takes the prior information of samples into consideration, it is now one of the hottest areas in machine learning and data mining fields. Moreover, it has been applied to a wide range of tasks such as natural spoken dialog systems, vision recognition, medical diagnosis, genetic regulatory network inference, and so forth [1–8]. Naive Bayes (NB) [9–11] is a simple and effective classification model. Although its performance can be comparable with other classification methods, such as decision trees and neural network, its attribute of independence assumption limits its real application. Extending its structure is a direct way to overcome the limitation of naive Bayes [12–14], since attribute dependencies can be explicitly represented by arcs. Tree-augmented naive Bayes (TAN) [9] is an extended tree-like naive Bayes in which the class node directly points to all attribute nodes and an attribute node can have only one parent from another attribute node. On this basis, Cheng et al.

presented Bayesian-network-Augmented naive Bayes (BAN) [15, 16] which further expanded the tree-like structure of TAN classifier and allowed the dependency relation between any two attribute nodes. In constructing BAN, they use a scoring function based on the minimum description length principle. Unfortunately, the search for the best network is performed in the space of all possible networks, and the number of elements in this space increases exponentially with the number of nodes, finding the best structure is NP-hard [17, 18].

Based on above analysis, this paper presents a Bayesian classifier learning algorithm based on optimization model (BC-OM) for the first time, inspired by constraint-based Bayesian network structure learning method [19–22]. We discuss the classification principles of Bayesian classifier from a new view. Because chi-squared tests are a standard tool for measuring the dependency between pairs of variables [23], BC-OM first introduces the chi-squared statistic to define the dependence coefficients of variables. Then, it uses the dependence coefficients to construct an overall measure of the dependence in a classifier structure, from which the objective function for our optimization model can be derived. Therefore, a problem of searching for an optimal classifier can be turned into finding the maximum value of the objective

FIGURE 1: Illustration of NB, TAN, and BAN structures.

function in feasible fields. The function extremum corresponds to the best classifier. Finally, BC-OM improves the efficiency of classification and delete irrelevant or redundant attributes by using the d-separation rule of Bayesian network. Theoretical and experimental results show the proposed algorithm is not only effective in improving the accuracy, but also has a high learning speed and simple solving procedure.

The remainder of this paper is organized as follows. Section 2 reviews the existing Bayesian network classifiers. We describe our algorithm and its theoretical proofs in Section 3. Section 4 details the experimental procedures and results of the proposed algorithm. Finally, in Section 5, we conclude and outline our future work.

2. Background

In this section, we discuss previous work that is relevant to this paper, and describe some of the notations used firstly. We use boldface capital letters such as $\mathbf{V}$, $\mathbf{E}$, $\mathbf{X}$ for sets of variables. General variables are denoted by italic capital letters or index italic capital letters X , R , V_{-} ; specific values taken by these variables are denoted x_{-} , r_{-} , v_{-} . Specially, we use the same italic letters X , Y , V_{-} for graph nodes which corresponds with the random variables.

Classification is a basic task in data analysis and pattern recognition that requires the construction of a classifier, that is, a function that assigns a class label to instances described by a set of attributes. The induction of classifiers from data sets of preclassified instances is a central problem in machine learning. Let $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$ represent the variable set which corresponds with the training data set D . We assume that V_1 is the class variable and $\{V_2, V_3, \dots, V_n\}$ is the set of attribute variables. Bayesian networks are often used for classification problems, in which the main task is to construct the classifier structure G from a given set of training data with class labels and then compute the posterior probability $P_G(v_1 | V_2, V_3, \dots, V_n)$, where v_1 is the value that V_1 takes. Thus, it only needs to predict the class with the highest value of probability $P_G(v_1 | V_2, V_3, \dots, V_n)$, that is,

$$v_1^* = \arg \max_{v_1 \in \text{dom}(V_1)} P_G(v_1 | V_2, V_3, \dots, V_n). \quad (1)$$

According to Bayes theorem, maximizing $P_G(v_1 | V_2, V_3, \dots, V_n)$ is equivalent to maximizing $P(v_1) \cdot P_G(V_2, V_3, \dots, V_n | v_1)$. The difference between the existing Bayesian classifiers is the computing mode of $P_G(\cdot)$.

Figure 1 schematically illustrates the structures of the Bayesian classifiers considered in this paper. In naive Bayes, each attribute node has the class node as its parent, but does not have any parent from attribute nodes. Computing $P_G(V_2, V_3, \dots, V_n | v_1)$ is equal to $\prod_{i=2}^n P_G(V_i | v_1)$. Because the values of $P_G(v_1)$ and $P_G(V_i | v_1)$ can be easily estimated from training examples, naive Bayes is easy to construct. However, its prerequisite of condition independence assumption and data completeness limit its real application. TAN takes the naive Bayes and adds edges to it in which the class node directly points to all attribute nodes and an attribute node can have only one parent from another attribute node. Computing $P_G(V_2, V_3, \dots, V_n | v_1)$ is equivalent to $\prod_{i=2}^n P_G(V_i | V_j, v_1)$, where $V_j \in \{V_1, V_2, \dots, V_{(i-1)}\}$. It is an efficient extend of naive Bayes. BAN is a specific case of general Bayesian network classifier, in which the class node also directly points to all attribute nodes, but there is no limitation on the arcs among attribute nodes (except that they do not form any directed cycle). It is clear that TAN and BAN are useful to model correlations among attribute nodes that cannot be captured by naive Bayes. They embody a good tradeoff between the equality of the approximation of correlations among attributes and the computational complexity in the learning stage. In addition, existing algorithms using the same idea to construct the structure of Bayesian classifier which first learn the dependent relationships among attribute variables using Bayesian network structure learning algorithm, then add the class variable as the root node of the network. It is equivalent to learning the best Bayesian network among those in which V_1 is a root. Thus, even if we could improve the performance of a naive Bayes classifier in this way, the computational effort required may not be worthwhile.

Based on above analysis, this paper presents an optimization model to learn the structure of Bayesian classifier, which inspired by constraint-based Bayesian network structure learning method. It is the first time that a problem of structural learning for a Bayesian classifier is transformed into its related mathematical programming problem by defining objective function and feasible region. And, we also propose a new method to measure the dependent relationships between attributes. The theoretical basis of this method is established by Theorem 1 [24].

Theorem 1. Given a data set D and a variable set $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$, if the hypothesis that V_i and V_j are conditionally

independent given V_k is true, then the statistics $U_{ij|k}^2 = 2 \sum_{a,b,c} N_{ijk}^{abc} \log[N_{ijk}^{abc} N_k^c / (N_{ik}^{ac} N_{jk}^{bc})]$ approximates to a distribution $\chi^2(l)$ with $l = (r_i - 1)(r_j - 1)r_k$ degrees of freedom, where r_i , r_j , and r_k represent the number of configurations for the variables V_i , V_j , and V_k , respectively. N_{ijk}^{abc} is the number of cases in D where $V_i = a$, $V_j = b$, and $V_k = c$. N_{ik}^{ac} is the number of cases in D where $V_i = a$ and $V_k = c$ and N_k^c is the number of cases in D where $V_k = c$.

3. A Bayesian Classifier Learning Algorithm Based on Optimization Model

3.1. Optimization Model Design. In this subsection, we give some basic concepts and theorems which is the foundation of the method proposed in this paper.

A Bayesian classifier is a graphical representation of a joint probability distribution that includes two components. One is a directed acyclic graph $G = (\mathbf{V}, \mathbf{E})$, where the node set $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$ represents the class and attribute variables, and the edge set $\mathbf{E}$ represents direct dependency relationships between variables. The other is a joint probability distribution $\Theta = \{\theta_i \mid \theta_i = P(V_i \mid \text{pa}(V_i)), i = 1, 2, \dots, n\}$ that quantifies the effects of $\text{pa}(V_i)$ has on the variable V_i in G , where $\text{pa}(V_i) = \{V_j \mid V_j \rightarrow V_i \in \mathbf{E}\}$. We assume that V_1 is the class node and $\{V_2, V_3, \dots, V_n\}$ is the set of attribute nodes. The structure of G reflects the underlying probabilistic dependence relations among the nodes and a set of assertions about conditional independencies. The problem of data classification can be stated as follows: the learning goal is first to find the classifier structure that best matches D and estimate the parameters using the training data set D , then to assign class label to test instances. Since G is a directed acyclic graph, it can be represented by a binary node-node adjacency matrix $X = (x_{ij})$. Entry (i, j) is 1 if there is a directed arc from node i to node j , and 0 otherwise. That is,

$$x_{ij} = \begin{cases} 1, & \text{if } V_i \rightarrow V_j \in \mathbf{E}, \\ 0, & \text{if } V_i \rightarrow V_j \notin \mathbf{E}. \end{cases} \quad (2)$$

Let $R = X + X^2 + \dots + X^n$ be the sum of powers of the adjacency matrix. Entry $r_{ij} \in R$ is equal to the number of directed paths from node V_i to node V_j in the graph [25].

We wish to be able to use a mathematical programming formulation, and this formulation requires that we are able to measure the impact of adding or removing a single arc from the network. In order to approximate the impact of adding such an arc, we define the dependence coefficient.

Definition 2. Given a data set D and a variable set $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$, we define the dependence coefficient $c_{ij\alpha}$ between variables V_i and V_j as $c_{ij\alpha} = \min_{k \neq i, j} \{U_{ij|k}^2 - \chi_{ij|k, \alpha}^2\}$, where $U_{ij|k}^2$ is the U^2 statistics of V_i and V_j given V_k , $\chi_{ij|k, \alpha}^2$ is the critical value at the significance level α of a χ^2 distribution with $(r_i - 1)(r_j - 1)r_k$ degrees of freedom.

Obviously, $c_{ij\alpha}$ is a conservative estimate of the degree of dependence between two nodes. If $c_{ij\alpha} > 0$, then, regardless

of the other variable involved, there is statistically significant dependence between V_i and V_j , so there should be an arc between them. If $c_{ij\alpha} < 0$, then there is at least one way of conditioning the relationship so that significant dependence is not present. We define $C = (c_{ij})$ the dependence coefficient matrix corresponding to the variable set $\mathbf{V}$, that is,

$$c_{ij} = \begin{cases} c_{ij\alpha}, & \text{if } i \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Lemma 3. Given a data set D and a variable set $\mathbf{V}$, V_i and V_j are locally conditionally independent at the significance level α if and only if there is a node $V_k \in \mathbf{V}$ such that $U_{ij|k}^2 < \chi_{ij|k, \alpha}^2$.

The proof of Lemma 3 can be obtained directly by Definition 2 and chi-square hypothesis test. According to Lemma 3, V_i and V_j are locally conditionally independent at the significance level α if and only if there is a node $V_k \in \mathbf{V}$ such that $c_{ij} < 0$. Further, V_i and V_j are globally conditionally independent at the significance level α if and only if $U_{ij|k}^2 < \chi_{ij|k, \alpha}^2$ for any $k \neq i, j$. Based on this, we use the dependence coefficients to construct an overall measure of the dependence which will be treated as the objective function for our mathematical program.

Definition 4. For a Bayesian classifier with adjacency matrix $X = (x_{ij})$, the global dependence measure of the network is given by

$$\mathcal{F}(X, \alpha) = \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij}. \quad (4)$$

According to the measure of Definition 4, if V_i and V_j are conditionally independent, by Lemma 3 $c_{ij} < 0$, and hence, adding an arc between V_i and V_j decreases the value of $\mathcal{F}$. Thus, we wish to find the feasible solution which increases $\mathcal{F}$. The optimal solution corresponds to the best classifier structure. We next explain what constitutes feasible network.

Given the variable set $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$, V_1 is the class node and $\{V_i, i = 2, \dots, n\}$ is the set of attribute nodes. A directed network is a feasible classifier structure if and only if the following conditions are satisfied:

- (1) for any attribute node $V_i \in \mathbf{V}$, $i = 2, \dots, n$, there is no directed edge from V_i to V_1 ;
- (2) for any node $V_i \in \mathbf{V}$, $i = 1, 2, \dots, n$, there is no directed path from V_i to V_i , namely, the graph is acyclic;
- (3) there exists at least one attribute node $V_i \in \mathbf{V}$, $i = 2, \dots, n$ which is dependent with class node V_1 , namely, there is an attribute node V_i such that V_i can be reached from V_1 by a directed path.

In order to incorporate the requirements of the above three conditions into a mathematical programming formulation, we express them by the following constraints:

- (1) $\sum_{i=2}^n x_{i1} = 0$;

- (1) **Input:** Data set D ; Variable set $\mathbf{V} = (V_1, V_2, \dots, V_n)$ (V_1 is the class node and others are attribute nodes).
- (2) For any pair of variable V_i and V_j contained in $\mathbf{V}$, calculate the dependence coefficient c_{ij} by Definition 2;
- (3) Solve the mathematical programming (OM) and obtain the optimal solution $X^* = (x_{ij})$;
- (4) Build the classifier structure $G^* = (\mathbf{V}, \mathbf{E}^*)$ by $X^* = (x_{ij})$;
- (5) For any variable $V_i \in \mathbf{V}$, $i \neq 1$, if there is a subset $\mathbf{Z} \subset \mathbf{V} \setminus \{V_i, V_j\}$ such that $\mathbf{Z}$ d-separates V_i and V_j , then delete V_i in G^* , update G^* as $\mathbf{V} = \mathbf{V} \setminus \{V_i\}$, $\mathbf{E}^* = \mathbf{E}^* \setminus \{V_i \rightarrow V_j \text{ or } V_j \rightarrow V_i, V_j \in \text{adj}(V_i)\}$;
- (6) **Output:** The built classifier $G^* = (\mathbf{V}, \mathbf{E}^*)$.

ALGORITHM 1: BC-OM.

- (2) $\sum_{i=1}^n r_{ii} = 0$;
- (3) $\sum_{j=2}^n r_{1j} \geq 1$.

The feasible classifiers are those that satisfy constrains (1)–(3). Thus, learning best Bayesian classifier can be transformed into the following related mathematical programming problem, where the objective function is a global dependence measure of the network, and the feasible region is the set of classifiers with reachability constrains (1)–(3), that is,

$$\begin{aligned}
 \max \quad & \mathcal{F}(X, \alpha) = \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij}, \\
 \text{s.t.} \quad & \sum_{i=2}^n x_{i1} = 0, \\
 & \sum_{i=1}^n r_{ii} = 0, \\
 & \sum_{j=2}^n r_{1j} \geq 1, \\
 & x_{ij} \in \{0, 1\}.
 \end{aligned} \tag{OM}$$

3.2. BC-OM Algorithm and Its Correctness. In this subsection, we present the main algorithm of this paper. Our method starts with finding the best Bayesian classifier by solving the above optimization model. Second, we use the d-separation rule of Bayesian network to delete irrelevant or redundant attributes in the network which have low dependence degree with the class variable. The parameters of modified network can be estimated. Third, classification is done by applying obtained classifier to predict the class label of test data. We prove the correctness of proposed method under the faithfulness assumption for the data distribution.

Given a directed acyclic graph $G = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ is the node set and $\mathbf{E}$ the set of directed edges. A path ρ between two distinct nodes V_1 and V_l is a sequence of distinct nodes in which the first node is V_1 , the last one is V_l and two consecutive nodes are connected by an edge, that is $\rho = V_1 e_1 V_2 \cdots e_{l-1} V_l$ where e_i denotes $V_i \rightarrow V_{i+1}$ or $V_i \leftarrow V_{i+1}$ for $i = 1, 2, \dots, (l-1)$.

Definition 5. A path ρ is said to be d-separated by a set $\mathbf{Z}$ in a directed acyclic graph G if and only if (1) ρ contains a “head-to-tail meeting”: $V_i \rightarrow V_j \rightarrow V_k$ or a “tail-to-tail meeting”: $V_i \leftarrow V_j \leftarrow V_k$ such that the middle node V_j is in $\mathbf{Z}$, or (2) ρ

contains a “head-to-head meeting”: $V_i \rightarrow V_j \leftarrow V_k$ such that the middle node V_j is not in $\mathbf{Z}$ and no descendant of V_j is in $\mathbf{Z}$. Specially, two distinct sets of nodes $\mathbf{X}$ and $\mathbf{Y}$ are said to be d-separated by a set $\mathbf{Z}$ in G if $\mathbf{Z}$ d-separates every path from any node in $\mathbf{X}$ to any node in $\mathbf{Y}$ [26].

In this paper, we assume that all the distributions are compatible with G [27]. We also assume that all independencies of a probability distribution of variables in $\mathbf{V}$ can be checked by d-separations of G , called the faithfulness assumption [26]. The faithfulness assumption means that all independencies and conditional independencies among variables can be represented by G . Now we formally describe our method in the following Algorithm 1.

From the detailed steps of BC-OM, we can see that BC-OM classifier relaxes the restrictions on condition variable and further meets the need of practical application. Since its network structure is similar to that of BAN's, BC-OM does not need to build all possible networks in which class node is a root and removes irrelevant or redundant nodes from the network before the process of estimating the network parameters, which greatly reduces the calculation for posterior probability of class variable. In fact, the training process of BC-OM is different from other BN classifiers. Its main task is to solve the mathematical programming (OM). To create the dependence coefficient matrix corresponding to (OM), BC-OM needs to compute the conditional statistics $U_{ij|k}^2$. Moreover, just as other constraint based algorithms, the main cost of BC-OM is the number of conditional independence tests for computing the dependence coefficients of any two variables in step 2. The number of conditional independence tests is $C_n^2 \cdot C_{n-2}^1$ and the computing complexity is $O(n^3)$. The total complexity of BC-OM is bound by $O(n^3 \cdot N)$, where n is the number of variables in the network and N is the number of cases in data set D . In principle, BC-OM is a structure-extension-based algorithm. In BC-OM, we essentially extend the structure of TAN by relaxing the parent set of each attribute node. Thus, the resulting structure is more complex than TAN, but more simple than BAN. Therefore, BC-OM is a good tradeoff between the model complexity and accuracy compared with TAN and BAN. Next, we prove the correctness of BC-OM algorithm under the faithfulness assumption.

The next two results establish the existence and uniqueness properties of solution to (OM).

Theorem 6. Let $\mathcal{X} = \{(x_{ij})_{n \times n} \mid x_{ij} \in \{0, 1\}\}$. There always exists an $X \in \mathcal{X}$ such that X is a feasible point of (OM).

Proof. Given the set of variables $V = \{V_1, V_2, \dots, V_n\}$ where V_1 is the class variable and $\{V_2, \dots, V_n\}$ are the attribute variables. We give a matrix X as follows:

$$x_{ij} = \begin{cases} 1, & i = 1, j = 2, \dots, n, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Obviously, the adjacency matrix X always satisfies the constraints (1)–(3). In fact, the graph represented by X is the Naive Bayes classifier. Thus, X is a feasible solution of (OM). $\square$

According to Theorem 6, we can prove that there exists a feasible classifier which satisfy constraints (1)–(3). Theorem 7 further shows that such classifier is unique under certain condition.

Theorem 7. Let X^* be the optimal solution of (OM), $C_1 = \{c_{ij} \mid x_{ij} = 1\}$ and $C_2 = \{c_{ij} \mid x_{ij} = 0\}$ be the coefficient sets where x_{ij} is the element of X^* . X^* is the unique solution of (OM) if and only if any element in C_1 cannot be expressed as the sum of any number of elements in C_2 .

Proof. Without loss of generality, we suppose, by reduction to absurdity, that $X^{(1)}$ and $X^{(2)}$ are two optimal solutions of (OM). The values of the objective function is the same in both solutions, that is,

$$\sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij}^{(1)} = \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij}^{(2)} \quad (6)$$

$$\implies \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} (x_{ij}^{(1)} - x_{ij}^{(2)}) = 0. \quad (7)$$

Let $C_1^{(1)} = \{c_{ij} \mid x_{ij}^{(1)} = 1\}$, $C_2^{(1)} = \{c_{ij} \mid x_{ij}^{(1)} = 0\}$. According to the assumption of $X^{(1)} \neq X^{(2)}$ and $x_{ij}^{(1)}, x_{ij}^{(2)} \in \{0, 1\}$, there must exist $1 \leq i_0, j_0 \leq n$ such that $x_{i_0 j_0}^{(1)} - x_{i_0 j_0}^{(2)} = 1$, namely, $x_{i_0 j_0}^{(1)} = 1, x_{i_0 j_0}^{(2)} = 0$. Then, by (7),

$$c_{i_0 j_0} + \sum_{i \neq i_0} \sum_{j \neq j_0} c_{ij} (x_{ij}^{(1)} - x_{ij}^{(2)}) = 0. \quad (8)$$

Since

$$x_{ij}^{(1)} - x_{ij}^{(2)} = \begin{cases} 1, & x_{ij}^{(1)} = 1, x_{ij}^{(2)} = 0, \\ 0, & x_{ij}^{(1)} = x_{ij}^{(2)}, \\ -1, & x_{ij}^{(1)} = 0, x_{ij}^{(2)} = 1, \end{cases} \quad (9)$$

then, by (8), there must exist $c_{i_1 j_1}, \dots, c_{i_k j_k} \in C_2^{(1)}$ such that $x_{i_r j_r}^{(1)} - x_{i_r j_r}^{(2)} = -1, r = 1, \dots, k$ and $c_{i_0 j_0} + \sum_{r=1}^k c_{i_r j_r} (x_{i_r j_r}^{(1)} - x_{i_r j_r}^{(2)}) = 0$, namely,

$$c_{i_0 j_0} = \sum_{r=1}^k c_{i_r j_r}, \quad (10)$$

where $x_{i_0 j_0}^{(1)} = 1, x_{i_r j_r}^{(1)} = 0, r = 1, \dots, k$. This contradicts with the known condition that any element in $C_1^{(1)}$ cannot be expressed as the sum of any number of elements in $C_2^{(1)}$. $\square$

Theorem 8. Let $G = (V, E), V = (V_1, V_2, \dots, V_n)$ be the classifier structure obtained by step 4 of BC-OM, where V_1 is the class variable and $V_2, \dots, V_n$ are attribute variables. $G^* = (V^*, E^*)$ denotes the final output of BC-OM, then the classification results obtained by G and G^* are consistent.

Proof. Without loss of generality, suppose $(d_{i_2}, \dots, d_{i_n})$ is an example to be classified. The classifier represented by G is given as follows:

$$v_1^* = \arg \max_{v_1 \in \text{dom}(V_1)} p(v_1 \mid v_2 = d_{i_2}, \dots, v_n = d_{i_n}). \quad (11)$$

We write the right side of (11) as $p(v_1 \mid v_2, \dots, v_n)$ in short. We can suppose that k redundant variables were deleted in step 5 of BC-OM, say the last k variables $V_{n-k+1}, V_{n-k+2}, \dots, V_n$. Then, $V^* = \{V_1, V_2, \dots, V_{n-k}\}$. According to step 5, $\{V_2, \dots, V_{n-k}\}$ d-separates V_1 and $\{V_{n-k+1}, \dots, V_n\}$. Thus, V_1 is conditional independent with $\{V_{n-k+1}, \dots, V_n\}$ given $\{V_2, \dots, V_{n-k}\}$. Equation (11) can be reduced as follows:

$$\begin{aligned} v_1^* &= \arg \max_{v_1 \in \text{dom}(V_1)} \frac{p(v_1, v_2, \dots, v_n)}{p(v_2, \dots, v_n)} \\ &\iff v_1^* = \arg \max_{v_1 \in \text{dom}(V_1)} p(v_1, v_2, \dots, v_n) \\ &\iff v_1^* = \arg \max_{v_1 \in \text{dom}(V_1)} p(v_1, v_{n-k+1}, \dots, v_n \mid v_2, \dots, v_{n-k}) \\ &\quad \cdot p(v_2, \dots, v_{n-k}) \\ &\iff v_1^* = \arg \max_{v_1 \in \text{dom}(V_1)} p(v_1 \mid v_2, \dots, v_{n-k}) \\ &\quad \cdot p(v_{n-k+1}, \dots, v_n \mid v_2, \dots, v_{n-k}) \\ &\iff v_1^* = \arg \max_{v_1 \in \text{dom}(V_1)} p(v_1 \mid v_2, \dots, v_{n-k}). \end{aligned} \quad (12)$$

We get the results. $\square$

Theorem 8 reveals that it is effective and correct to remove redundant or irrelevant attributes using d-separation rule, and the performance of Bayesian classifier can be improved.

4. Experimental Results

We run our experiments on 20 data sets from the UCI repository of Machine Learning datasets [28], which represent a wide range of domains and data characteristics. Table 1 shows the description of the 20 data sets which are ordered by ascending number of samples. In our experiments, missing values are replaced with the modes and means of the corresponding attribute values from the available data. For example, if the sex of someone is missing, it can be replaced by the mode (the value with the highest frequency) of the sexes of all the others. Besides, we manually delete three

TABLE 1: Descriptions of UCI datasets used in the experiments.

NO.	Dataset	# Instances	# Attributes	# Classes	Missing value (N/Y)
1	Shuttle	15	6	2	N
2	Balloon	16	4	2	N
3	Lenses	24	4	3	N
4	Soybean (small)	47	35	4	N
5	Post-patient	90	8	3	Y
6	Zoo	101	17	7	N
7	Hayes-roth	132	5	3	N
8	Iris	150	4	3	N
9	Glass	214	10	7	N
10	Heart	267	22	2	N
11	Ionosphere	351	34	2	N
12	Monk's	432	7	2	N
13	Vote	435	16	2	Y
14	Balance-scale	625	5	3	N
15	Tic-tac-toe	958	9	2	N
16	Car-evaluation	1728	6	4	N
17	Chess (kr-vs-kp)	3196	36	2	N
18	Mushroom	8124	22	2	Y
19	Nursery	12960	8	5	N
20	Letter	20000	16	26	N

useless attributes: the attribute “ID number” in the dataset “Glass”, the attribute “name” in the dataset “Hayes-roth”, and the attribute “animal name” in the dataset “Zoo”.

The experimental platform is a personal computer with Pentium 4, 3.06 GHz CPU, 0.99 GB memory, and Windows XP. Our implementation is based on the BayesNet Toolbox for Matlab [29], which provides source code to perform several operations on Bayesian networks. The purpose of these experiments is to compare the performance of the proposed BC-OM with Naive Bayes, TAN and BAN in terms of classifier accuracy. The accuracy of each model is based on the percentage of successful predictions on the test sets of each data set. In all experiments, the accuracy of each model on each data set are obtained via 10 runs of 5-fold cross validation. Runs with the various algorithms are 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 the experiments on each data set. Finally, we compared related algorithms via two-tailed t -test with a 95 percent confidence level. According to the statistical theory, we speak of two results for a data set as being “significantly different” only if the probability of significant difference is at least 95 percent [30].

Table 2 shows the accuracy (and standard deviation of accuracy) of each model on each data set, and the average values and standard deviation on all data sets are summarized at the bottom of the table. In each row, the best of the four classifier results are displayed in bold. If another's performance is not significantly different from the best, it is also highlighted, but if the differences between all four classifiers are not statistically significant, then none of them

is highlighted. From our experiments, we can see that BC-OM is best in 6 cases. NB, TAN, and BAN are best in 6, 5, and 3 cases, respectively. When the number of samples is larger than 400, the performance of TAN and BAN is better than that of NB, and BC-OM is best. Although it can be seen that the performance of BC-OM and TAN becomes similar as the sample size increase, BC-OM has a higher accuracy on average. From a general point of view, we can see that from the first data set to the last one, the highlighted numbers change from few to even more in the sixth column of Table 2. It means the advantage of BC-OM is more evident with the increase of data size.

Table 3 shows the compared results of two-tailed t -test, in which each entry $w/t/l$ means that the model in the corresponding row wins in w data sets, ties in t data sets and loses in l data sets, compared to the model in the corresponding column. From Table 3, we can see that BC-OM significantly outperforms NB (9 wins and 4 losses), TAN (12 wins and 5 losses) and BAN (11 wins and 5 losses) in accuracy. Figures 2, 3, and 4 show two scatter-plots comparing BC-OM with NB, TAN, and BAN, respectively. In the scatter plot, each point represents a data set, where the x coordinate of a point is the percentage of misclassifications according to NB or TAN or BAN, and the y coordinate is the percentage of misclassifications according to BC-OM. Thus, points below the diagonal line correspond to data sets on which BC-OM performs better. From Figures 2 and 3, we can see that BC-OM generally outperforms NB and TAN as is also demonstrated in Table 3. It provides strong evidence that BC-OM is performing well against the other two classifiers both in terms of accuracy as well as

TABLE 2: The detailed experimental results on accuracy and standard deviation.

NO.	Dataset	NB	TAN	BAN	BC-OM
1	Shuttle	83.52 $\pm$ 5.85	85.33 $\pm$ 7.03	84.43 $\pm$ 6.83	82.33 $\pm$ 5.94
2	Balloon	89.15 $\pm$ 3.06	85.16 $\pm$ 3.15	87.16 $\pm$ 2.92	83.75 $\pm$ 3.57
3	Lenses	92.56 $\pm$ 2.44	93.25 $\pm$ 2.85	89.67 $\pm$ 3.16	91.12 $\pm$ 3.02
4	Soybean (small)	90.91 $\pm$ 2.35	84.98 $\pm$ 3.16	87.36 $\pm$ 2.44	89.74 $\pm$ 3.23
5	Post-patient	80.29 $\pm$ 3.39	79.94 $\pm$ 3.47	81.26 $\pm$ 3.56	80.12 $\pm$ 3.48
6	Zoo	93.46 $\pm$ 5.12	95.21 $\pm$ 4.78	94.34 $\pm$ 5.16	94.88 $\pm$ 5.09
7	Hayes-roth	81.82 $\pm$ 5.27	87.34 $\pm$ 4.89	84.58 $\pm$ 3.74	84.55 $\pm$ 3.56
8	Iris	94.13 $\pm$ 6.12	93.47 $\pm$ 6.65	93.81 $\pm$ 6.80	89.07 $\pm$ 7.25
9	Glass	66.02 $\pm$ 9.88	59.67 $\pm$ 9.52	58.32 $\pm$ 9.98	56.96 $\pm$ 9.81
10	Heart	82.41 $\pm$ 6.62	82.53 $\pm$ 7.01	82.65 $\pm$ 7.03	81.53 $\pm$ 6.91
11	Ionosphere	90.54 $\pm$ 4.03	90.61 $\pm$ 3.98	91.45 $\pm$ 4.14	90.99 $\pm$ 4.82
12	Monk's	84.28 $\pm$ 1.14	83.29 $\pm$ 1.29	83.79 $\pm$ 1.23	86.64 $\pm$ 1.02
13	Vote	94.88 $\pm$ 3.57	90.37 $\pm$ 3.34	92.63 $\pm$ 3.98	90.97 $\pm$ 2.95
14	Balance-scale	90.91 $\pm$ 1.51	84.97 $\pm$ 2.44	87.94 $\pm$ 2.31	90.77 $\pm$ 1.85
15	Tic-tac-toe	70.11 $\pm$ 4.45	73.16 $\pm$ 4.47	75.74 $\pm$ 4.83	76.21 $\pm$ 3.82
16	Car-evaluation	84.95 $\pm$ 2.74	93.44 $\pm$ 1.89	89.2 $\pm$ 1.02	95.42 $\pm$ 1.03
17	Chess (kr-vs-kp)	87.88 $\pm$ 1.91	91.68 $\pm$ 1.66	90.27 $\pm$ 1.12	92.65 $\pm$ 1.4
18	Mushroom	92.85 $\pm$ 0.56	99.89 $\pm$ 0.1	96.37 $\pm$ 0.76	99.91 $\pm$ 0.1
19	Nursery	90.54 $\pm$ 1.14	94.07 $\pm$ 1.29	92.31 $\pm$ 1.23	89.95 $\pm$ 1.09
20	Letter	69.32 $\pm$ 3.97	77.58 $\pm$ 1.83	73.45 $\pm$ 1.67	77.66 $\pm$ 2.01
	Average	86.85 $\pm$ 4.81	86.63 $\pm$ 3.7	85.69 $\pm$ 3.69	86.76 $\pm$ 3.57

TABLE 3: The compared results of two-tailed t -test on accuracy with the 95 percent confidence level.

$w/t/l$	NB	TAN	BAN
TAN	13/3/4	—	—
BAN	12/4/4	10/5/5	—
BC-OM	9/7/4	12/3/5	11/4/5

the percentage of misclassifications. Figure 4 also shows BC-OM outperforming BAN, though the difference in performance is not as marked as in the results of Figures 2 and 3. In other words, the performance of BC-OM and BAN is similar in terms of the percentage of misclassifications. However, BC-OM has a higher accuracy and a more simple graph structure, which suggests that BC-OM is able to handle very large data sets and is a more promising classifier.

5. Conclusions

In many real-world applications, classification is often required to make optimal decisions. In this paper, we summarize the existing improved algorithms for naive Bayes and propose a novel Bayesian classifier model: BC-OM. We conducted a systematic experimental study on a number of UCI datasets. The experimental results show that BC-OM has a better performance compared to the other state-of-the-art models for augmenting naive Bayes. It is clear that in some situations, it would be useful to model correlations among

FIGURE 2: Relative errors of BC-OM and NB.

attributes. BC-OM is a good tradeoff between the quality of the approximation of correlations among attributes and the computational complexity in the learning stage. Considering its simplicity, BC-OM is a promising model that could be used in many field.

In addition, we use the chi-squared statistic to estimate the dependence coefficients among attributes from dataset. We believe that the use of more sophisticated methods could improve the performance of the current BC-OM and make

FIGURE 3: Relative errors of BC-OM and TAN.

FIGURE 4: Relative errors of BC-OM and BAN.

its advantage stronger. This is the main research direction for our future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 60974082 and 61075055), the National Funds of China for Young Scientists (no. 11001214), and the Fundamental Research Funds for the Central Universities (no. K5051270013).

References

- [1] Z. Cai, S. Sun, S. Si, and B. Yannou, "Identifying product failure rate based on a conditional Bayesian network classifier," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5036–5043, 2011.
- [2] D. C. Kim, X. Wang, C. R. Yang, and J. Gao, "Learning biological network using mutual information and conditional independence," *BMC Bioinformatics*, vol. 11, no. 3, article S9, 2010.
- [3] N. C. Hsieh and L. P. Hung, "A data driven ensemble classifier for credit scoring analysis," *Expert Systems with Applications*, vol. 37, no. 1, pp. 534–545, 2010.
- [4] S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.
- [5] Y. Sun, Y. Y. Tang, S. X. Ding, S. P. Lv, and Y. F. Cui, "Diagnose the mild cognitive impairment by constructing Bayesian network with missing data," *Expert Systems with Applications*, vol. 38, no. 1, pp. 442–449, 2011.
- [6] V. Aquaro, M. Bardoscia, R. Bellotti, A. Consiglio, F. de Carlo, and G. Ferri, "A Bayesian networks approach to operational risk," *Physica A*, vol. 389, no. 8, pp. 1721–1728, 2010.
- [7] S. Nikolajewa, R. Pudimat, M. Hiller, M. Platzer, and R. Backofen, "BioBayesNet: a web server for feature extraction and Bayesian network modeling of biological sequence data," *Nucleic acids research*, vol. 35, pp. W688–693, 2007.
- [8] I. Ben-Gal, A. Shani, A. Gohr et al., "Identification of transcription factor binding sites with variable-order Bayesian networks," *Bioinformatics*, vol. 21, no. 11, pp. 2657–2666, 2005.
- [9] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.
- [10] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proceedings of the 10th National Conference on Artificial Intelligence*, P. Rosenbloom and P. Szolovits, Eds., pp. 223–228, AAAI Press, Menlo Park, Calif, USA, 1992.
- [11] M. Minsky, "Steps toward artificial intelligence," *Proceedings of the Institute of Radio Engineers*, vol. 49, pp. 8–30, 1961.
- [12] L. Jiang, H. Zhang, and Z. Cai, "A novel bayes model: hidden naive bayes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1361–1371, 2009.
- [13] A. A. Balamurugan, R. Rajaram, S. Pramala, S. Rajalakshmi, C. Jeyendran, and J. Dinesh Surya Prakash, "NB+: an improved Naive Bayesian algorithm," *Knowledge-Based Systems*, vol. 24, no. 5, pp. 563–569, 2011.
- [14] L. Jiang, Z. Cai, D. Wang, and H. Zhang, "Improving tree augmented naive Bayes for class probability estimation," *Knowledge-Based Systems*, vol. 26, pp. 239–245, 2012.
- [15] J. Cheng and R. Greiner, "Comparing Bayesian network classifiers," in *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, K. B. Laskey and H. Prade, Eds., pp. 101–108, Morgan Kaufmann, San Francisco, Calif, USA, 1999.
- [16] M. G. Madden, "On the classification performance of TAN and general Bayesian networks," *Knowledge-Based Systems*, vol. 22, no. 7, pp. 489–495, 2009.
- [17] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Francisco, Calif, USA, 1988.
- [18] D. M. Chickering, "Learning Bayesian networks is NP-Complete," in *Learning From Data: Artificial Intelligence and Statistics V*, D. Fisher and H. Lenz, Eds., pp. 121–130, Springer, New York, NY, USA, 1996.
- [19] C. Borgelt, "A conditional independence algorithm for learning undirected graphical models," *Journal of Computer and System Sciences*, vol. 76, no. 1, pp. 21–33, 2010.
- [20] A. M. Mart'inez-Rodr'iguez, J. H. May, and L. G. Vargas, "An optimization-based approach for the design of Bayesian networks," *Mathematical and Computer Modelling*, vol. 48, no. 7-8, pp. 1265–1278, 2008.
- [21] L. M. de Campos, "A scoring function for learning Bayesian networks based on mutual information and conditional independence tests," *Journal of Machine Learning Research*, vol. 7, pp. 2149–2187, 2006.

- [22] X. W. Chen, G. Anantha, and X. T. Lin, "Improving bayesian network structure learning with mutual information-based node ordering in the K2 algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 628–640, 2008.
- [23] W. J. Conover, *Practical Nonparametric Statistics*, John Wiley & Sons, New York, NY, USA, 1999.
- [24] S. Kullback, *Information Theory and Statistics*, Dover, 1968.
- [25] G. Chartrand and P. Zhang, *Introduction to Graph Theory*, McGraw-Hill, New York, NY, USA, 2005.
- [26] R. E. Neapolitan, *Learning Bayesian Networks*, Prentice Hall, Englewood Cliffs, NJ, USA, 2004.
- [27] D. M. Chickering and C. Meek, "On the incompatibility of faithfulness and monotone DAG faithfulness," *Artificial Intelligence*, vol. 170, no. 8-9, pp. 653–666, 2006.
- [28] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, Calif, USA, 2010, <http://archive.ics.uci.edu/ml/>.
- [29] K. P. Murphy, *BaysNetToolbox For Matlab*, 2007, <https://code.google.com/p/bnt/>.
- [30] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.