Improvement of CB & BC Algorithms (CB* Algorithm) for Learning Structure of Bayesian Networks as Classifier in Data Mining

Benhard Sitohang[1] & G.A. Putri Saptawati[2]

[1 & 2] Data & Software Engineering Research Division
School of Electrical Engineering & Informatics, Institut Teknologi Bandung
[1] benhard@stei.itb.ac.id, [2]putri@informatika.org

Abstract. There are two categories of well-known approach (as basic principle of classification process) for learning structure of Bayesian Network (BN) in data mining (DM): scoring-based and constraint-based algorithms. Inspired by those approaches, we present a new CB* algorithm that is developed by considering four related algorithms: K2, PC, CB, and BC. The improvement obtained by our algorithm is derived from the strength of its primitives in the process of learning structure of BN. Specifically, CB* algorithm is appropriate for incomplete databases (having missing value), and without any prior information about node ordering.

Keywords: *Classification, Constraint-based Algorithm, Missing Value, Node Ordering, Scoring-based Algorithm.*

1 Introduction

Using Bayesian Networks (BN) approach as basic principle of classifier process is one of the promising and effective classifiers since this approach performs competitively with other known methods [1], [2]. In fact, learning the structure of BN from data is one of the active research topics in data mining. Unlike other classes of BN approach (Naïve-Bayes, tree augmented Naïve-Bayes (TAN), Bayesian Network augmented Naïve-Bayes [BAN], and general BNs [GBN]), the basic idea in CB* algorithm is to present *probabilistic dependences and independences derived from the data* in a concise way. In this way, the structure can be employed as a classifier, and it can give then the posterior probability distribution of the classification variable, given the values of other variables [1], [3].

Since the concept of data mining becomes popular in 1992, there were many algorithms developed for learning effective BN classifiers, and most of the algorithms assume that data are complete (i.e., there is no missing value). This assumption is in fact not realistically correct. So, this unrealistic assumption becomes one of the main reasons for performance degradation.

The rest of this paper is organized as follows. The next section presents BN in general, and describes approaches on how to learn BN from database, using various algorithms proposed by different researchers (K2, PC, BC, and CB algorithm). In Section 3, an experimental result from the case of *Visit to Asia* is presented. Section 4 presents our BC* algorithm for learning structure of BN from incomplete database.

2 Bayesian Network (BN)

Principally, Bayesian Network (BN) is a graphical representation of joint probability distribution of a set of random variables [4]. In case of $U = \{x_1, \ldots x_n\}$, joint probability distribution for U is the probability over all states of U, where $x_1, \ldots x_n$ are domain variables. In graphical representation, the structure of a BN is a directed acyclic graph (DAG) where each node corresponds to one domain variable x_i. An arc between nodes represents direct dependency between variables. In case there is no arc between two domain variables, those two domain variables are independent one of another.

The chain rule for BN is as follows [5], [6]:

Let BN is a Bayesian Network over $U = \{x_1, \ldots x_n\}$. Then, the joint probability distribution $P(U)$ is the product of all probabilities specified in BN;

$$P(x_1,...,x_N) = \prod_{i=1}^{N} P(x_i | pa_i) \quad (1)$$

where pa_i is the parent set of x_i.

Global joint probability distribution in BN approach is constructed by combining a set of local conditional probability distributions, with a set of assertion of conditional independencies [5]. The statement of conditional probability is expressed as follow:

> Given the event B, the probability of event A is x.
>
> The notation for the statement above is: $P(A \mid B) = x$.

Conditional independence is very important property in Bayesian networks [7], [6], [8]. If variables A and C are independent given B, it can be expressed as:

$$P(a_i \mid b_j) = P(a_i \mid b_j, c_k) \quad (2)$$

This expression means that if the state B is known, then the presence of C will not alter the probability of A.

Basically, BN is defined by two components. The DAG is the qualitative part of BN. The conditional probability table (CPT) for each variable is the quantitative part. Figure 1 provides an example of BN.

A class label attribute can be any node within the network. As a result, there may be more than one class label attribute. The classification process does not return a single class label. Instead, it can return a probability distribution for the class label attribute, i.e., predicting the probability of each class.

Referring to the description of BN above, Bayesian Network is one of the fundamental methods for prediction in classification category of data mining [1], [9], [10]. As a classifier, BN defines *posterior probability distribution* of a class based on the variable value. The classification process is composed of two sub-processes: (i) learning process, and (ii) BN inferences process. The goal of learning BN is to construct the BN structure. The BN Inference is for instance classification.

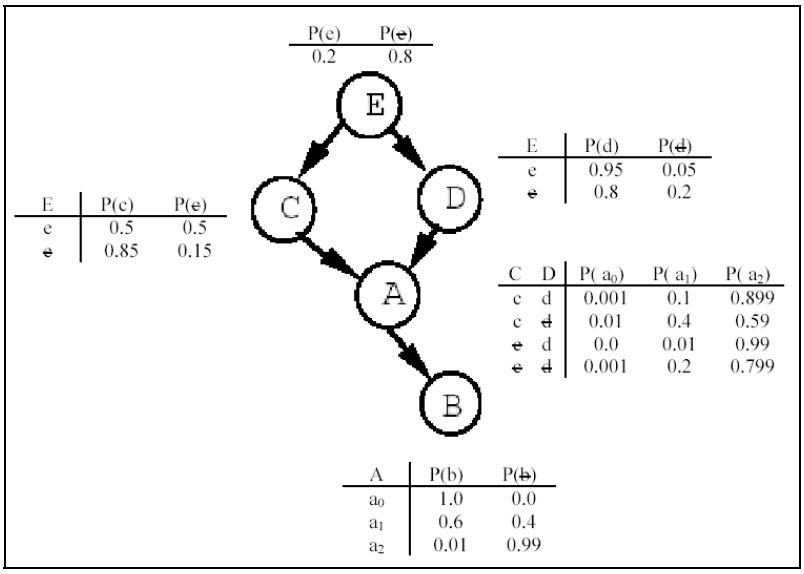

Figure 1 DAG of BN.

As a probabilistic graphical model (represented as Directed Acyclic Graph or DAG), the BN structure consists of nodes and arcs between nodes, so called probabilistic relation, that can be supposed as representation of uncertain knowledge. Therefore, it can be used to manipulate probability of attribute as a basic element of knowledge [11], [1], [12], [13], [14]. Pearl in [15] concludes that BN is a Directed Graph Acyclic (DAG), composed by node (as

representation of variable) and edge (representation of any relation between variable/relation of variables). In general, components of BN can be defined as:

a. Node: representing a set of variable of data in database, which can be discrete, continue, or categorical.
b. Edge: representing relationship between nodes that can be interpreted as dependency of a set of variables (conditional dependency).
c. Conditional Probability Table (CPT): representing prior probability. CPT consists of probabilistic distribution of variable to its parent node, and is a part of BN parameter. Each node in BN has CPT.
d. Directed Acyclic Graph (DAG): representing the whole structure of BN.

2.1 Learning the Structure of BN

There are two approaches to learning structure of Bayesian Network [1]:

a. *Scoring-based algorithms*. In this approach, BNs are perceived as joint probability distribution of variables and the algorithms seek for the Bayesian structures that best fit the data.
b. *Constraint-based algorithms*. These approaches seek for Bayesian structures by identifying the conditional independence relationships among the variables. Then, the relationships are used as constraints to construct a BN. These approaches perceive BN as a set of conditional independence relationships among variables.

There are several learning algorithms of BN structures derived from those principles. K2 algorithm, PC algorithm, Bound & Collapse algorithm, and CB algorithm will be presented in the following sections.

2.1.1 K2 Algorithm

K2 is an algorithm based on scoring-based approach for GBN [11], [8]. The basic idea of K2 is a greedy search algorithm. It seeks for DAG that maximizes Bayesian scoring method. The objective of K2 algorithm is to find the most probable network structure B, given a data set D, that is, maximizing the probability $P(B \mid D)$. In this case, K2 algorithm constructs BN structure using prior information of node ordering, and this algorithm works only on complete database (without any missing value).

2.1.2 PC Algorithm

PC is one of the famous constraint-based algorithms, which is developed by Spirtes & Glymour [8]. Given the set of conditional independencies in a

probability distribution, the PC algorithm tries to find a DAG for which the Markov condition entails all and only those conditional independencies. The complete algorithm is described in [8]. The algorithm consists of three phases: (i) identifying undirected graph, (ii) identifying uncoupled head-to-head meeting, and (iii) orient edges [16].

Identifying undirected graph: The aim of the first phase is to find the graph framework for which all pairs of dependent nodes are connected by an edge. All pairs of independent nodes are not connected by an edge.

Identifying uncoupled head-to-head meeting: Based on the result of previous phase, this second phase seeks for all uncoupled meeting (X–Z–Y) potentially being uncoupled head-to-head meeting (X→Z←Y). Then, an edge on that uncoupled meeting is oriented to form uncoupled head-to-head meeting.

Orient Edges: finally, the algorithm orients the rest of edges to produce an acyclic graph.

PC algorithm constructs BN structure without any prior information of node ordering, and works on complete database.

2.1.3 CB Algorithm

CB is developed by Singh & Valtorta as a hybrid algorithm that employs PC and K2 algorithm approaches. The basic idea is to avoid node ordering requirement [11]. To achieve that idea, CB employs a modified version of PC to find a node order, and then modified version of K2 is employed to learn the Bayesian Network structure. The first phase of CB is aimed to generate an undirected graph and to orient edges. Then, node ordering is determined and taken as input for the second phase. Referring to this node ordering, the second phase learns the network structure [17]. The complete algorithm can be found in [18].

All algorithms discussed above are based on the assumption that data are complete. However, this is an unrealistic assumption since missing values are often found in real world. Some well-known methods address this issue by using approximation methods to approximate the marginal likehood of the data being used [19]. These methods perform the task based on the Missing Information Principle (i.e., filling in the missing data on the basis of available information, using appropriate method). However, the quality of processes is highly dependent on the number of missing data, and the applied method.

2.1.4 Bound and Collapse (BC) Method

To minimize the problem of missing value, Bound and Collapse (BC) method is proposed by Ramoni et *al.* [11], [6], [19]. This method starts by bounding the set of possible estimates (interval estimate) based on the available observations in a database. Then, it collapses the resulting interval estimate to a unique point estimate by combining extreme values of the interval estimate. Based on this unique point estimate, dependency in data is calculated for which it becomes the basis for learning the structure. Node ordering & scoring functions are required for the learning process. In real applications, there is an expectation that the BC method is better because it handles any missing value. BC algorithm requires prior information of node ordering in order to construct BN structure on incomplete database (containing missing value).

3 Experimental Results

This section briefly discusses prior experimental results. There was an experimental study conducted on different algorithms described above. In this experiment, a well known case "Visit to Asia" from Lauritzen et *al*. is employed [9]. The case, which has 8 binary variables and 5,000 records as depicted in Table 1, represents dyspnoea diagnosis. K2, PC, CB and BC algorithms were run on this case.

Table 1 Variables in "Visit to Asia".

Variable Name	Description
Visit to Asia ?	Has patient ever visited Asia ?
Smoking ?	Is patient a smoker ?
Tuberculosis ?	Does patient have tuberculosis ?
Lung cancer ?	Does patient have lung cancer ?
Either tub or lung Cancer ?	Hybrid variables of tuberculosis & lung cancer. True/yes if patient has tuberculosis or lung cancer
Bronchitis ?	Does patient have bronchitis ?
Positive X-ray ?	Does test X-ray show positive result ?
Dyspnoea ?	Does patient have dsypnoea ?

The first experiment was mainly focused on the comparison of performances among K2, PC and CB algorithms [16]. The final result shows that CB outperforms others (K2, PC) in that:

a. CB can generate node ordering resulting in a structure that is Markov equivalence to original structure.

b. CB is efficient by reducing search space.

The experiment result with BC algorithm shows that [20]:

a. The number of missing data does not have any influence in the generated structure. The BC algorithm in the first phase has already addressed the problem of missing value.
b. For two cases with the same numbers of missing data, the generated structures can be different if their distributions of missing data are also different. This is expected because the algorithm may use different strategy on how to handle the missing values.

4 Proposed Algorithm

Scoring-based and constraint-based algorithms as described above show their advantages and disadvantages. The experiment results indicate that hybrid algorithm is superior (i.e. BC method) than the others. However, all of those algorithms work only on complete database. There is no such an algorithm that can learn the structure of BN from incomplete database.

This paper proposes a so-called CB* algorithm, which is developed as an improved version of CB algorithm. Figure 2 describes the complete algorithm of CB*. It combines the 2-phase of BC algorithm (CB* is composed by 2 phases) and is especially developed for incomplete database (in addition to complete database as in its predecessor) by applying the BC algorithm.

The first phase of CB* consists of seven steps (steps 1 to 7) derived from (as part of) CB algorithm [1]. This phase mainly constructs DAG of BN.

The second phase of algorithm is composed by three steps (steps 8 to 10). This phase learns the structure of BN from incomplete data, exactly the same as applied by BC algorithm.

Let $A_G ab$ be the set of vertices adjacent to a or b in graph G not including a and b. Also, let u be a bound on the degree of the undirected graph generated by step 2. ord is the order of CI relations being tested. Let π_i be the set of parents of node i, $1 \le i \le n$.

1. Start with the complete graph G_1 on the set of vertices Z. //step 1
   $ord \leftarrow 0$.
   $old\_\pi_i \leftarrow \{\ \}$ $\forall i$, $1 \le i \le n$, and $old\_Prob \leftarrow 0$.

2. (based on **[11]**) //step 2
   Modify G_1 as follows:
   For each pair of vertices a, b that are adjacent in G_1, $A_{G_1}ab$ has a cardinality greater than or equal to ord, and $I(a, S_{ab}, b)$ where $S_{ab} \subseteq A_{G_1}ab$ of cardinality ord, remove the edge $a-b$, and store S_{ab}.
   If for all pairs of adjacent vertices a,b in G_1, $A_{G_1}ab$ has cardinality < ord, goto step 10.
   If degree of $G_1 > u$, then
   $ord \leftarrow ord + 1$
   Goto beginning of step 2

3. Let G be the copy of G_1. //step 3
   For each pair of non adjacent variables a,b in G, if there is a node c that is not in S_{ab} and is adjacent to both a and b, then orient the edges from $a \rightarrow c$ and $b \rightarrow c$ **(see [2],[6])** unless such an orientation leads to the introduction of directed cycle in the graph.
   If an edge has already been oriented in the reverse direction, make that edge bidirectional.

4. Try to assign directions to the yet undirected edges in G by applying the following four rules **[2],[5]** if this can be done without introducing directed cycles in the graph: //step 4
   Rule 1: if $a \rightarrow b$ and $b-c$ are not adjacent, then direct $b \rightarrow c$
   Rule 2: if $a \rightarrow b$, $b \rightarrow c$ and $a-c$, then direct $a \rightarrow c$
   Rule 3: if $a-b$, $b-c$, $b-d$, $a \rightarrow d$ and $c \rightarrow d$, then direct $b \rightarrow d$
   Rule 4: if $a-b$, $b-c$, $a-c$, $c-d$ and $d-a$, then direct

$a \rightarrow b$ and $c \rightarrow b$

Moreover, if $a \rightarrow b$, $b \rightarrow c$ and $a \leftrightarrow c$, then $a \rightarrow c$

5. Let $\pi_i \leftarrow \{\}$ $\forall i, 1 \le i \le n$. //step 6
For each node i, add to π_i the set of vertices j such that for each such j, there is an edge $j - i$ in the pdag G.

6. For each undirected or bidirected edge in the pdag G choose an orientation as described below. //step 6
If $i - j$ in a directed edge, and π_i and π_j are the corresponding parent sets in G, then calculate the following products

$$i_{val} = g(i, \pi_i) \times g(j, \pi_j \cup \{i\})$$

$$j_{val} = g(j, \pi_j) \times g(i, \pi_i \cup \{j\})$$

If $i_{val} > j_{val}$, then $\pi_j \leftarrow \pi_i \cup \{i\}$ unless the addition of this edge, i.e. $i - j$ leads to a cycle in the pdag. In that case, choose the reverse orientation, and change π_i (instead of π_j). Do similar thing in case $j_{val} > i_{val}$

7. The sets $\pi_i, 1 \le i \le n$ obtained by step 6 define a DAG since for each node i, π_i consists of those nodes that have a directed edge to a node i.
Generate a total order on the nodes from this DAG by performing a topological sort on it. //step 7

8. Apply bound and collapse phases of BC method to find a unique point estimate for all possible pairs of node based on the node ordering in step 7. Then, find the set of parents of each node using the order in step 7. Let π_i be the set of parents, found by BC, of node i, $\forall i \quad 1 \le i \le n$
Let $new\_Prob = \Pi_{i=1}^{n} g(i, \pi_i)$

9. If *new_Prob* > *old_Prob*, then //step 9
old_Prob $\leftarrow$ *new_Prob*

```
          ord ← ord + 1
          old_π_i ← π_i  ∀i  1 ≤ i ≤ n
   Discard G
   Goto Step 2
      Else goto Step 10

10. Output old_π_i ∀i  1 ≤ i ≤ n //step 10
      Output old_Prob
```

Figure 2 CB* Algorithm.

As originally proposed as part of BC algorithm [6], the algorithm for finding the set of parents is as follows:

```
For each node X_i, 1 ≤ i ≤ n, find Π_i as
follows
   Π_i ← ϕ
   P(ϕ) ← ĝ(X_i | Π_i)
      For each node P_i, P_i ∈ Pred(X_i), do
         P_new ← ĝ(X_i | P_i)
         If P_new > P(ϕ) then
               Π_i ← Π_i ∪ { P_i }
         end{if}
      end {for}
end {for}
```

The first phase of CB* Algorithm uses CI test to generate undirected graph, followed by the process to define the direction of an edge as part of node ordering. This phase consists of seven steps as in the following:

a. Like CB algorithm, steps 1 to 4 are derived from algorithm introduced by Verma & Pearl [6] and Spirtes & Glymour [4]. In step 3, if CI test is not applied completely, then the edge can be bidirectional. As a consequence, some CI relations may not be found (there are some extra edges). Step 2 is mainly applied to avoid network structure produced by a dense-graph. The bound test applied in this step does not have any impact on the structure constructed, but it can be useful to reduce the process run time.

b. Step 5 searches for potential parent set from each node, using the direction of edge. The tail node of an undirected edge will become a Parent node.

c. To define the direction of undirected or bidirectional edge, step 6 calculates the value of $i_{val} = g(i, \pi_i) \times g(j, \pi_j \cup \{i\})$ and $j_{val} = g(j, \pi_j) \times g(i, \pi_i \cup \{j\})$, where π_i and π_j are potential parent sets found in step 5. If $i_{val} > j_{val}$, then the direction of node becomes i → j and $\pi_j \leftarrow \pi_j \cup i$. If this arc creates a directed cycle, then the reverse direction j → i is selected and $j_{val} > i_{val}$. Step 6 produces a complete DAG.

d. The last step performs topological sort of DAG and defines total ordering. It generates node ordering of DAG.

The first phase as described above is exactly the same as CB algorithm, which is mainly utilized to produce node ordering.

The second phase of CB* algorithm uses node ordering as input, and performs step 8 (modified BC step) followed by steps 9 and 10. This step constructs network structure in the presence of incomplete database. The modified BC step is specially introduced in order to be able to analyse an incomplete database as utilized in the BC algorithm.

The main advantages of the proposed CB* algorithm are derived from the advantages of its primitives (especially BC and CB algorithms), i.e.:

a. Generate node ordering resulting a structure that is Markov equivalence to original structure.

b. Construct BN structure from incomplete database (can be applied in the presence of missing value).

c. The number of missing data does not influence the BN structure.

d. Construct BN structure without prior information.

5 Conclusion & Future Work

CB* algorithm as proposed deserves more attention in the research field of learning BN structure. Theoritically, its performance is expected superior because it combines the advantages from both the scoring-based algorithms and the constraint-based algorithms. Moreover, CB* algorithm may also have the capability to learn structure of BN from incomplete databases without relying on the Missing Information Principles. Consequently, CB* algorithm is a promising method.

Future work will be focused on empirically evaluating the performance of CB* algorithm in terms of the quality of BN structure it generates. There is a preliminary judgement that it may lead to low performance (compared to its primitives, i.e. BC and/or CB algorithm) due to the complexity of algorithm, especially if the test is performed on limited volume of data. However, the algorithm may show a better performance on high volume of data.

References

[1] Cheng, J. & Geiner, R., *Comparing Bayesian Network Classifiers*, Proceedings of 15th International Conference on Uncertainty in AI, 1999.

[2] Friedman, N. & Koller, D., *Learning Bayesian Networks from Data*, 1998.

[3] Steck, H. & Tresp, V., *Bayesian Belief Networks for Data Mining*, Siemens AG, Corporate Technology Information and Telecommunications, Munich, Germany.

[4] Madden, M.G., *A New Bayesian Network Structure for Classification Tasks*.

[5] Heckerman, D., *A Tutorial on Learning Bayesian Network*, Technical Reports, 1995.

[6] Jensen, F.V., *Bayesian Networks and Decision Graphs*, Springer, 2001.

[7] Cheng, Jie & Greiner R., *Learning Bayesian Network Classifier: Algorithms dan System*, Proceeding of 14th Biennial conference of the Canadian Society for Computational Studies of Intelligence, 2001.

[8] Neapolitan, R.E., *Learning Bayesian Networks*, Pearson Prentice Hall, 2004.

[9] Han, J. & Kamber, M., *Data Mining: Concepts and Techniques*, Morgan Kaufman, 2001.

[10] Kruse, Rudolf & Borgelt, C., *Graphical Models: Analysis Tool for Data Mining*, Dept. of Knowledge Processing and Language Engineering, Otto-von-Guericke-University of Magdeburg, Germany, 2000.

[11] Cheng, J., Bell, D. & Liu, W., *Learning Bayesian Networks from Data: A Efficient Approach Based on Information Theory*, 1997.

[12] Heckerman, Chickering, *A Comparison of Scientific and Engineering Criteria for Bayesian Model Selection*, Microsoft Research, 1997.

[13] Hirsalmi, M., *Method Feasibility Study: Bayesian Networks*, Research Report, 2000.

[14] Murphy, Kevin P., *A Brief Introduction to Graphical Model and Bayesian Networks*, UC Berkeley, 2003.

[15] Pearl, J. *Graphical Models for Probabilistic and Causal Reasoning*, Computer Science Department, University of California, 1997.

[16] Sophia, A., *Algoritma PC sebagai Alternatif Pendekatan Analisis Dependensi untuk Konstruksi Struktur Bayesian Network dalam Data*

Mining, Final Project Dept of Informatics, Institute of Technology Bandung, 2005.

[17] Sandhyaduhita P.I., *Algoritma CB: Algoritma yang Dibangun dengan Dua Pendekataan untuk Konstruksi Bayesian Network dalam Data Mining*, Final Project Dept of Informatics, Institute of Technology Bandung, 2005.

[18] Singh, M. & Valtorta, M., *Construction of Bayesian Network Structures from Data: a Brief Survey and Efficient Algorithm*, Department of Computer Science, Univ. of South Carolina, Columbia, USA, 2005.

[19] Ramoni, M. & Sebastiani P., *Learning Bayesian Networks from Incomplete Databases*, KMI-TR-43, 1997.

[20] Maharani, H., *Konstruksi Struktur Bayesian Network dalam Data Mining untuk Basis Data Incomplete dengan Metode Bound and Collapse*, Final Project Dept of Informatics, Institute of Technology Bandung, February 2005.