A Novel Greedy Bayesian Network Structure Learning Algorithm for Limited Data

Feng Liu[1], Fengzhan Tian[2], and Qiliang Zhu[1]

[1] Department of Computer Science, Beijing University of Posts and Telecommunications,
Xitu Cheng Lu 10, 100876 Beijing, China
lliufeng@hotmail.com

[2] Department of Computer Science, Beijing Jiaotong University,
Shangyuan Cun 3, 100044 Beijing, China

Abstract. Existing algorithms for learning Bayesian network (BN) require a lot of computation on high dimensional itemsets, which affects accuracy especially on limited datasets and takes up a large amount of time. To alleviate the above problem, we propose a novel BN learning algorithm MRMRG, Max Relevance and Min Redundancy Greedy algorithm. MRMRG algorithm is a variant of K2 algorithm for learning BNs from limited datasets. MRMRG algorithm applies Max Relevance and Min Redundancy feature selection technique and proposes Local Bayesian Increment (LBI) function according to the Bayesian Information Criterion (BIC) formula and the likelihood property of overfitting. Experimental results show that MRMRG algorithm has much better efficiency and accuracy than most of existing BN learning algorithms when learning BNs from limited datasets.

1 Introduction

There are many problems in fields as diverse as medical diagnosis, weather forecast, fault diagnosis, where there is a need for models that allow us to reason under uncertainty and take decisions, even when our knowledge is limited. To model this type of problems, AI community has proposed Bayesian network which allows us to reason under uncertainty. [1] During the last two decades, many BN learning algorithms have been proposed. But, the recent explosion of high dimensional and limited datasets in the biomedical realm and other domains has induced a serious challenge to these BN learning algorithms. The existing algorithms must face higher dimensional and smaller datasets.

In general, BN learning algorithms take one of the two approaches: the constraint-based method and the search & score method. The constraint-based approach [2],[3],[15] estimates from the data whether certain condition independences hold between variables. Typically, this estimation is performed using statistical or information theoretical measure. The search & score approach [4],[5],[6],[9],[12],[13] attempts to find a graph that maximizes the selected score. Score function is usually defined as a measure of fitness between the graph and

R. Alhajj et al. (Eds.): ADMA 2007, LNAI 4632, pp. 412–421, 2007.

the data. These algorithms use a score function in combination with a search method in order to measure the goodness of each explored structure from the space of feasible solutions. During the exploration process, the score function is applied in order to evaluate the fitness of each candidate structure to the data.

Although encouraging results have been reported, the two approaches both suffer some difficulties in accuracy on limited datasets. A high order statistical or information theoretical measure may become unreliable on limited datasets. At the same time, the result of selected score function may also be unreliable on limited datasets.

To further enhance learning efficiency and accuracy, this paper proposes Max-Relevance and Min-Redundancy Greedy BN learning algorithm. MRMRG algorithm applies Max-Relevance and Min-Redundancy feature selection technology to obtain better efficiency and accuracy on limited datasets, and proposes Local Bayesian Increment function according to BIC approximation formula and the likelihood property of overfitting for limited datasets.

This paper is organized as follows. Section 2 provides a brief review of some basic concepts and theorems. Section 3 describes K2 algorithm. In Section 4, we propose Local Bayesian Increment function. Section 5 represents the details of MRMRG algorithm. At the same time, we also analyze the time complexity of MRMRG. Section 6 shows an experimental comparison among K2 and MRMRG. Finally, we conclude and present future work.

2 Concepts and Theorems

2.1 Bayesian Network

A Bayesian network is defined as a pair $B = \{G, \Theta\}$, where G is a directed acyclic graph $G = \{V(G), A(G)\}$, with a set of nodes $V(G) = \{V_1, \ldots, V_n\}$ representing a set of random variables and a set of arcs $A(G) \subseteq V(G) \times V(G)$ representing causal independence/dependence relationships that exist among the variables. Θ represents the set of parameters that quantifies the network. It contains a parameter $\theta_{v_i|\pi_i} = P(v_i \mid \pi_i)$ for each possible value v_i of V_i, and π_i of Π_i. Here Π_i denotes the set of parents of V_i in G and π_i is a particular instantiation of Π_i.

2.2 Max-Dependence and MRMR

Definition 1. *In feature selection, **Max-Dependence scheme** [7] is to find a feature set S with m features, which jointly have the largest dependency on the target class C; $S = \arg \max_{\{X_i, i=1,\ldots,m\}} I(\{X_i, i = 1, \ldots, m\}; C)$.*

Definition 2. *In feature selection, **Max-Relevance criterion** [7] is to select a feature set S with m features satisfying $S = \arg\max_{S} \left(\frac{1}{|S|} \sum_{X_i \in S} I(X_i; C) \right)$, which approximates $I(\{X_i, i = 1, \ldots, m\}; C)$ with the mean value of all mutual information values between individual features $X_i, i = 1, \ldots, m$ and class C.*

Definition 3. *In feature selection,* ***Min-Redundancy criterion*** *[7] is to select a feature set S with m features such that they are mutually minimally similar (mutually maximally dissimilar): $S = \arg\min_{S} \left(\frac{1}{|S|^2} \sum_{X_i, X_j \in S} I(X_i; X_j) \right)$.*

Definition 4. *In feature selection,* ***Max-Relevance and Min-Redundancy criterion*** *[7] is to find a feature set S with m features obtained by optimizing the Max-Relevance criterion and the Min-Redundancy criterion simultaneously. Assume that the two conditions equally important, and consider the following criteria: $S = \arg\max_{S} \left(\sum_{X_i \in S} I(X_i; C) - \frac{1}{|S|} \sum_{X_i, X_j \in S} I(X_i; X_j) \right)$.*

We select the feature set $S_m = \{X_1, X_2, \ldots, X_m\}$, the classification variable C. Using the standard multivariate mutual information $MI(X_1, \ldots, X_m) = \int \int p(x_1, \ldots, x_m) \log \frac{p(x_1, \ldots, x_m)}{p(x_1) \ldots p(x_m)} dx_1 \ldots dx_n$, we can get the following formula:

$$I(S_m; C) = \int \int p(S_m, c) \log \frac{p(S_m, c)}{p(S_m) p(c)} dS_m dc = MI(S_m, C) - MI(S_m). \quad (1)$$

Equation (1) is similar to the MRMR feature selection criterion: The second term requires that the features S_m are maximally independent of each other(that is, minimum redundant), while the first term requires every feature to be maximally dependent on C. In practice, the authors have shown that if one feature is selected at one time, then MRMR criterion is almost optimal implementation scheme of Max-Dependence scheme on limited datasets. [8]

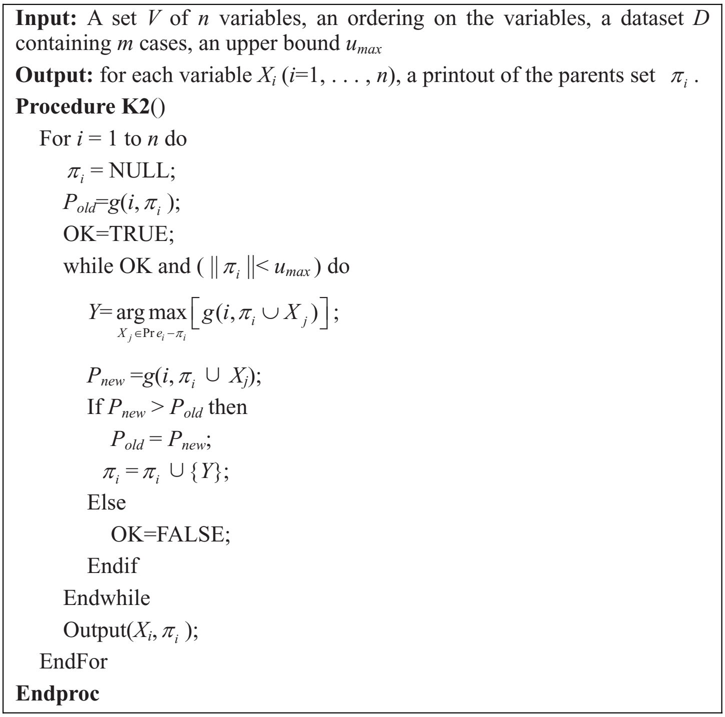
3 K2 Algorithm

Given a complete dataset D, K2 searches for the Bayesian network G^* with maximal $P(G, D)$.

Let D be a dataset of m cases, where each case contains a value for each variable in V. D is sufficiently large. Let V be a set of n discrete variables, where x_i in V has r_i possible values $(v_{i1}, v_{i2}, \ldots, v_{ir_i})$. Let G denote a Bayesian network structure containing just the variables in V. Each variable x_i in G has the parents set π_i. Let $\phi_i[j]$ denote the j^{th} unique instantiation of π_i relative to D. Suppose there are q_i such unique instantiation of π_i. Define $N_{ijk}$to be the number of cases in D in which variable x_i is instantiated as v_{ik} and π_i is instantiated as $\phi_i[j]$. Let $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Given a Bayesian network model, cases occur independently. Bayesian network prior distribution is uniform. It follows that $g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!$.

It starts by assuming that a node has no parents, and then in every step it adds incrementally the node which can most increase the probability of the resulting BN, to the parents set. K2 stops adding nodes to parents set when the addition



```
Input: A set $V$ of $n$ variables, an ordering on the variables, a dataset $D$
containing $m$ cases, an upper bound $u_{max}$
Output: for each variable $X_i$ ($i$=1, . . . , $n$), a printout of the parents set $\pi_i$ .
Procedure K2()
  For $i$ = 1 to $n$ do
    $\pi_i$ = NULL;
    $P_{old}$=$g(i, \pi_i)$;
    OK=TRUE;
    while OK and ( $\|\pi_i\| < u_{max}$ ) do
      $Y = \arg\max_{X_j \in \mathrm{Pr}e_i - \pi_i} \left[ g(i, \pi_i \cup X_j) \right]$;
      $P_{new}$ =g($i$, $\pi_i \cup X_j$);
      If $P_{new} > P_{old}$ then
        $P_{old} = P_{new}$;
      $\pi_i = \pi_i \cup \{Y\}$;
      Else
        OK=FALSE;
      Endif
    Endwhile
    Output($X_i$, $\pi_i$ );
  EndFor
Endproc
```

Fig. 1. K2 algorithm

cannot increase the probability of the BN given the data. The pseudo-code of MRMRG algorithm sees Fig.1.

Pre_i denotes the set of variables that precede X_i. π_i denotes the current parents set of the variable X_i. u_{max} denotes an upper bound on the number of parents a node may have.

4 Local Bayesian Increment Function

Let X and Y be two discrete variables, $\mathbf{Z}$ be a set of discrete variables, and z be an instantiation for $\mathbf{Z}$. $X, Y \notin \mathbf{Z}$.

Definition 5. *According to Moore's recommendation [10] about the chi-squared test, the dataset D satisfying the following condition is* ***sufficiently large*** *for* $\{X \cup Y\}$ *: All cells of* $\{X \cup Y\}$ *in the contingency table have expected value greater than 1, and at least* 80% *of the cells in the contingency table about* $\{X \cup Y\}$ *have expected value greater than 5.*

Definition 6. *According to Moore's recommendation [10] about the chi-squared test, the sub-dataset* $D_{\mathbf{Z}=z}$ *satisfying the following condition is* ***locally sufficiently large*** *for* $\{X \cup Y\}$ *given* $\mathbf{Z} = z$ *: All cells of* $\{X \cup Y\}$ *in the contingency* *table conditioned on $\boldsymbol{Z} = z$ have expected value greater than 1, and at least 80% of the cells in the contingency table about $\{X \cup Y\}$ on $\boldsymbol{Z} = z$ have expected value greater than 5.*

Learning on limited datasets, we loose the "locally sufficiently large" condition: If the number of cases in $D_{\mathbf{Z}=z}$ is much larger than the number of values for $\{X \cup Y\}$, for example $\|D_{\mathbf{Z}=z}\| \geq 4 \times (\|X\| \times \|Y\|)$; then we assume that the sub-dataset $D_{\mathbf{Z}=z}$ is "locally sufficiently large" for $\{X \cup Y\}$ given $\mathbf{Z} = z$.

Let D be a dataset of m cases. Let V be a set of n discrete variables, where X_i in V has r_i possible values $(v_{i1}, v_{i2}, \ldots, v_{ir_i})$. B_P and B_S denote BN structures containing just the variables in V. B_S exactly has one edge $Y \rightarrow X_i$ more than B_P. X_i has the parents set π_i in B_P and the parents set $\pi_i \cup Y$ in B_S. N_{ijk} is the number of cases in D, which variable X_i is instantiated as v_{ijk} and π_i is instantiated as $\phi_i[j]$. Let $N_{ijk} = \sum\limits_{y} N_{i,\{j \cup y\},k}, N_{ij} = \sum\limits_{k=1}^{r_i} N_{ijk}$. $\hat{\Theta}_i, \hat{\Theta}$ denote the maximum likelihoods of Θ_i, Θ. π_i^l denotes the instantiation of π_i in the lth case.

Cases occur independently. The prior distribution of possible Bayesian networks is uniform. Given a Bayesian network model, there exist two properties: Parameter Independence and Parameter Modularity. [5]

We apply the BIC formula also used by Steck in [11] : $BIC(B_S) = \log L\left(\hat{\Theta}\right) - \frac{1}{2}\log(m)dim\left(\hat{\Theta}\right) \approx \log(P(D \mid B_S))$ to control the complexity of BN model. BIC adds the penalty of structure complexity to LBI function to avoid overfitting.

Definition 7 (Local Bayesian Increment Function)

$$Lbi(Y, i, \pi_i) = \log\left(P(B_S, D)/P(B_P, D)\right) \approx BIC(B_S) - BIC(B_P)$$
$$= \log\left(L\left(\hat{\Theta}^{B_S}\right)/L\left(\hat{\Theta}^{B_P}\right)\right) - \frac{1}{2}\log(m)\left[dim\left(\hat{\Theta}^{B_S}\right) - dim\left(\hat{\Theta}^{B_P}\right)\right]$$
$$\log\left(L\left(\hat{\Theta}^{B_S}\right)/L\left(\hat{\Theta}^{B_P}\right)\right) = \log\left(P\left(D \mid \hat{\Theta}^{B_S}\right)\right) - \log\left(P\left(D \mid \hat{\Theta}^{B_P}\right)\right)$$
$$= \sum_{l=1}^{m} \log\left(P\left(x_i^l \mid \hat{\Theta}_i^{B_S}, \pi_i^l \cup y\right)/P\left(x_i^l \mid \hat{\Theta}_i^{B_P}, \pi_i^l\right)\right)$$

According to the likelihood property of overfitting(the marginal likelihood of overfitting for the training dataset is usually no less than non-overfitting), we assume that the log-likelihood does not change on the sub-dataset $D_{\pi_i=\phi_i[*]}$ which are not "locally sufficiently large" for $\{X \cup Y\}$ (that is, to assume that there is overfitting between X and the parents set $\pi_i \cup Y$ on the $D_{\pi_i=\phi_i[*]}$),

$$\sum_{d_l \in D_{\pi_i=\phi_i[*]}} \log P\left(x^l \mid \hat{\Theta}^{B_S}, \pi_l \cup y\right) = \sum_{d_l \in D_{\pi_i=\phi_i[*]}} \log P\left(x^l \mid \hat{\Theta}^{B_P}, \pi_l\right). \quad (2)$$

According to (2), we infer the following results:

$$\log\left(L\left(\hat{\Theta}^{B_S}\right)\right) - \log\left(L\left(\hat{\Theta}^{B_P}\right)\right)$$

$$= \sum_j N_{ij} \times I_j(X,Y), \text{for } j, D_{\pi_i=\phi_i[j]} \text{ is “locally sufficiently large”}$$

$$\dim\left(\hat{\Theta}^{B_S}\right) - \dim\left(\hat{\Theta}^{B_P}\right) = (r_y - 1)(r_i - 1)q_i$$

$$Lbi(Y,i,\pi_i) = \sum_j N_{ij} \times I_j(X,Y) - \frac{1}{2}(r_y - 1)(r_i - 1)q_i \log(m),$$

$$\text{for } j, D_{\pi_i=\phi_i[j]} \text{ is “locally sufficiently large”}.$$

Note: $I_j(X,Y)$ is the mutual information between X and Y on $D_{\pi_i=\phi_i[j]}$.

5 MRMRG Algorithm

MRMRG algorithm initializes the current parents set π_i of the variable X_i to NULL, and then adds the variables one by one, which acquire the maximal value for Local Bayesian Increment (LBI) function, into the parents set π_i from $Pre_i - \pi_i$, until the result of LBI function is no more than 0. Repeating the above steps for every variable, we can obtain an approximately optimal Bayesian network. The pseudo-code of MRMRG algorithm sees Fig.2.

Pre_i denotes the set of variables that precede X_i. π_i denotes the current parents set of the variable X_i. $(k < 5)$.

Given an ordering on the variables, MRMRG algorithm improves greedy BN learning algorithms (such as K2 algorithm [4]) in the following two ways in order to learn more accurately and efficiently on limited datasets.

Firstly, on limited datasets, the results of traditional scoring functions (such as K2 score [4],MDL score [6],BDe score [5], etc) $score(C, \pi_i \cup X_j)$ have less and less reliability and robustness with the dimension increase of $\pi_i \cup X_j$, so that the formula $Y = \arg \max_{X_j \in Pre_i - \pi_i} score(C, \pi_i \cup X_j)$ cannot obtain the variable Y with the maximal score, even cannot acquire a variable with approximately maximal score sometimes. Since MRMR technology only uses 2-dimensional computation, it has much higher reliability and robustness than traditional scoring functions on limited datasets. Furthermore, according to the discussion in section 2.2, we know that if one feature is selected at one time (that is Greedy search), MRMR technology is nearly optimal implementation scheme of Max-Dependence scheme, which is equivalent to the maximal score method, on limited datasets. We consider that for some variable $X_j \in Pre_i - \pi_i$, if the value of $\{I(X_j;C) - \frac{1}{|\pi_i|+1} \sum_{X \in \pi_i} I(X_j;X)\}$ is the largest, then it is the most probable that the value of the formula $score(C, \pi_i \cup X_j)$ is the largest. Thus, MRMRG algorithm applies Max-Relevance and Min-Redundancy (MRMR) feature selection technology and replaces $score(C, \pi_i \cup X_j)$ with the formula $\{I(X_j;C) - \frac{1}{|\pi_i|+1} \sum_{X \in \pi_i} I(X_j;X)\}$ to obtain the variable Y which gets the maximal score. Firstly, MRMRG algorithm selects the top k variables from the sorted variables set $Pre_i - \pi_i$ according to the value of the formula $\{I(X_j;C) - \frac{1}{|\pi_i|+1} \sum_{X \in \pi_i} I(X_j;X)\}$ by descendant order.

```
Input: A set V of n variables, an ordering on the variables, a dataset D containing m cases.
Output: for each variable $X_i$ ($i$=1, . . . , n), a printout of the parents set $\pi_i$ .
Procedure MRMRG()
  For i = 1 to n do
    Initialize $\pi_i$ to NULL and OK to TRUE;
    while OK
      For every variable $X \in Pre_i - \pi_i$, compute the formula $\left\{ I(X;C) - \frac{1}{|\pi_i|+1} \sum_{X_j \in \pi_i} I(X;X_j) \right\}$ (1);
      Sort the variables in $Pre_i - \pi_i$ by descendant order according to the value of (1);
      Obtain the top k variables $\{Y_1, Y_2, . . . , Y_k\}$ from the sorted variables set $Pre_i - \pi_i$ ;
      $Y_{\max} = \underset{Y_j \in \{Y_1, Y_2, ... Y_k\}}{\arg\max} \left[ Lbi(Y_j, i, \pi_i) \right]$ ;
      If $Lbi(Y_{\max}, i, \pi_i) > 0$ then
        $\pi_i = \pi_i \cup \{Y_{\max}\}$ ;
      Else
        OK=FALSE;
      Endif
    Endwhile
    Output ( $X_i$, $\pi_i$ );
  Endfor
Endproc
```

Fig. 2. MRMRG Bayesian network learning algorithm

Then, it take the variable Y with the largest value of LBI function among the k variables as the variable with the maximal score.

Secondly, MRMRG algorithm proposes LBI function to replace traditional score increment functions (such as K2 [4],MDL [6],BDe [5]) to control the complexity of Bayesian network and to avoid overfitting. When the dataset D is “sufficiently large” for $\{X \cup Y \cup \pi_i\}$, LBI function is equivalent to K2 increment function. When the dataset D is not ”sufficiently large” for $\{X \cup Y \cup \pi_i\}$, but there exist sub-datasets $D_{\pi_i = \phi_i[*]}$ are ”locally sufficiently large” for $\{X \cup Y\}$ given $\pi_i = \phi_i[*]$, MRMRG algorithm can also apply LBI function to improve accuracy and avoid overfitting (see section 4). The technique also makes it unnecessary to set the maximal parents number u_{max} of a node.

5.1 The Time Complexity of MRMRG

$r = max(r_i), i = 1, \ldots, n$, where r_i is the number of values for the variable X_i. The complexity of the formula $I(X;C) - \frac{1}{|\pi_i|+1} \sum_{X_j \in \pi_i} I(X;X_j)$ is O(n). The complexity of computing $Lbi(Y, i, \pi_i)$ is O(mnr). The while statement loops at most O(n) times, each time it is entered. The for statement loops n times. So, in the worst case, the complexity of **MRMRG()** is O(kmn^3r). On the other hand, the worst-case time complexity of K2 algorithm is O(mn^4r). Therefore, if

$n >> k$, then MRMRG has much better efficiency (more than one magnitude) than K2 algorithm.

6 Experimental Results

We implemented MRMRG algorithm, K2 algorithm, TPDA algorithm [3] and presented the comparison of the experimental results for 3 implementations.

Tests were run on a PC with Pentium4 1.5GHz and 1GB RAM. The operating system was Windows 2000. These programs were developed under Matlab 7.0. 5 Bayesian networks were used. Table 1 shows the characteristics of these networks. The characteristics include the number of nodes, the number of arcs, the maximal number of node parents/children(Max In/Out-Degree), and the minimal/maximal number of node values(Domain Range).

From these networks, we performed these experiments with 200, 500, 1000, 5000 training cases each. For each network and sample size, we sampled 20 original datasets and recorded the average results by each algorithm. Let $u_{max} = 7$ in Fig.1 and $k = 3$ in Fig.2.

Table 1. Bayesian networks

BN	Nodes Num	Arcs Num	Max In/Out-Degree	Domain Range
Insur	27	52	3/7	2-5
Alarm	37	46	4/5	2-4
Barley	48	84	4/5	2-67
Hailf	56	66	4/16	2-11
Munin	189	282	3/15	1-21

6.1 Comparison of Runtime Among Algorithms

A summary of the time results of the execution of all the 3 algorithms is in Table 2. We normalized the times reported by dividing by the corresponding running time of MRMRG on the same datasets and reported the averages over sample sizes. [14] Thus, a normalized running time of greater than 1 implies a slower algorithm than MRMRG on the same learning task. A normalized running time of lower than 1 implies a faster algorithm than MRMRG.

From the results, we can see that MRMRG has better efficiency than other 3 algorithms K2 and TPDA. In particular, for smaller sample sizes (200, 500, 1000), MRMRG runs several times faster than K2 and TPDA. For larger sample sizes (5000), MRMRG performs nearly one magnitude faster than K2.

6.2 Comparison of Accuracy Among Algorithms

We compared the accuracy of Bayesian networks learned by these 3 algorithms according to the BDeu score. The BDeu score corresponds to the posteriori probability of the learned structure.[5] The BDeu scores in our experiments

Table 2. Normalized Runtime

Size	MRMRG	K2	TPDA
200	1.0	2.39	8.82
500	1.0	5.57	7.61
1000	1.0	9.18	4.57
5000	1.0	12.03	1.22

Table 3. Average BDeu(Insur)

Size	MRMRG	K2	TPDA
200	-21.915	-22.993	-31.507
500	-19.032	-19.583	-28.786
1000	-19.386	-19.533	-25.111
5000	-18.177	-18.152	-20.857

Table 4. Average BDeu(Alarm)

Size	MRMRG	K2	TPDA
200	-15.305	-16.069	-24.456
500	-13.858	-13.950	-18.097
1000	-13.319	-13.583	-15.429
5000	-13.021	-12.979	-14.496

Table 5. Average BDeu(Barley)

Size	MRMRG	K2	TPDA
200	-81.794	-83.972	-106.782
500	-76.327	-79.194	-103.783
1000	-76.846	-77.375	-111.069
5000	-75.710	-76.281	-116.892

Table 6. Average BDeu(Hailf)

Size	MRMRG	K2	TPDA
200	-72.138	-73.361	-106.135
500	-71.217	-72.662	-101.382
1000	-70.955	-71.734	-97.374
5000	-70.277	-71.105	-84.300

Table 7. Average BDeu(Munin)

Size	MRMRG	K2	TPDA
200	-65.971	-68.393	-135.103
500	-62.483	-63.250	-125.625
1000	-61.967	-62.837	-140.476
5000	-59.392	-60.943	-145.635

were calculated on a seperate test set sampled from the true Bayesian network containing 50000 samples. Table 3-7 reports the results.

From the results, we can see that MRMRG can learn more accurately than TPDA on limited datasets. In particular, MRMRG has better accuracy than K2 on limited datasets. The accuracy of MRMRG is almost the same as K2 on larger datasets relative to the true Bayesian network, such as Insur(5000), Alarm(5000).

7 Conclusion

Efficiency and accuracy are two main indices in evaluating algorithms for learning Bayesian network. MRMRG algorithm greatly reduces the number of high dimensional computations and improves scalability of learning. The experimental results indicate that MRMRG has better performance on efficiency and accuracy than most of existing algorithms on limited datasets.

We are interesting in incorporate ordering-based search method [9] into our MRMRG algorithm to implement MRMRG without the information of the order between nodes. In addition, much more experimentation is needed on different network structures.

References

1. Heckerman, D.: Bayesian Networks for Data Mining, 1st edn. Microsoft Press, Redmond (1997)
2. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search, 2nd edn. MIT Press, Massachusetts (2000)
3. Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W.: Learning Belief Networks form Data: An Information Theory Based Approach. Artificial Intelligence 137(1-2), 43–90 (2002)
4. Cooper, G., Herskovits, E.: A Bayesian Method for Constructing Bayesian Belief Networks from Databases. In: Ambrosio, B., Smets, P. (eds.) UAI '91. Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence, pp. 86–94. Morgan Kaufmann, San Francisco (1991)
5. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian Networks: the Combination of Knowledge and Statistical Data. Machine Learning 20(3), 197–243 (1995)
6. Wai, L., Fahiem, B.: Learning Bayesian Belief Networks An approach based on the MDL Principle. Computational Intelligence 10(4), 269–293 (1994)
7. Hanchuan, P., Chris, D., Fuhui, L.: Minimum redundancy maximum relevance feature selection. IEEE Intelligent Systems 20(6), 70–71 (2005)
8. HanChuan, P., Fuhui, L., Chris, D.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on PAMI 27(8), 1226–1238 (2005)
9. Teyssier, M., Koller, D.: Ordering-Based Search: A Simple and Effective Algorithm for Learning Bayesian Networks. In: Chickering, M., Bacchus, F., Jaakkola, T. (eds.) UAI '05. Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, pp. 584–590. Morgan Kaufmann, San Francisco CA (2005)
10. Moore, D.S.: Goodness-of-Fit Techniques, 1st edn. Marcel Dekker, New York (1986)
11. Steck, H.: On the Use of Skeletons when Learning in Bayesian Networks. In: Boutilier, C., Goldszmidt, M. (eds.) UAI '00. Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, pp. 558–565. Morgan Kaufmann, San Francisco (2000)
12. Moore, A.W., Wong, W.K.: Optimal Reinsertion: A New Search Operator for Accelerated and More Accurate Bayesian Network Structure Learning. In: Fawcett, T., Mishra, N. (eds.) ICML 2003 – Machine Learning. Proceedings of the Twentieth International Conference, pp. 552–559. AAAI Press, Washington DC (2003)
13. Friedman, N., Nachman, I., Peter, D.: Learning Bayesian Network Structure from Massive Datasets: The Sparse Candidate algorithm. In: Laskey, K.B. (ed.) UAI '99. Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 196–205. Morgan Kaufmann, San Francisco (1999)
14. Ioannis, T., Laura, E.B.: The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. Machine Learning 65(1), 31–78 (2006)
15. Margaritis, D., Thrun, S.: Bayesian Network Induction via Local Neighborhoods. In: Solla, S.A., Leen, T.K., Muller, K. (eds.) NIPS '99. Advances in Neural Information Processing Systems 12, pp. 505–511. MIT Press, Cambridge (1999)