Bayesian Network Structure Ensemble Learning

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. Bayesian networks (BNs) have been widely used for learning model structures of a domain in the area of data mining and knowledge discovery. This paper incorporates ensemble learning into BN structure learning algorithms and presents a novel ensemble BN structure learning approach. Based on the Markov condition and the faithfulness condition of BN structure learning, our ensemble approach proposes a novel sample decomposition technique and a components integration technique. The experimental results reveal that our ensemble BN structure learning approach can achieve an improved result compared with individual BN structure learning approach in terms of accuracy.

1 Introduction

Bayesian network is an efficient tool to represent a joint probability distribution and causal independence relationships among a set of variables. Therefore, there has been great interest in automatically inducing Bayesian networks from datasets. [6] During the last two decades, two kinds of BN learning approaches have emerged. The first is the search & score method [2],[9], which uses heuristic search methods to find the Bayesian network that maximizes some given score function. Score function is usually defined as a measure of fitness between the graph and the data. The second approach, which is called the constraint-based approach, estimates from the data whether certain conditional independences hold among the variables. Typically, this estimation is performed using statistical or information theoretical measures [1],[11].

Although encouraging results have been reported, both of the approaches done so far suffer some computational difficulties in accuracy and cannot overcome the local maxima problem. A statistical or information theoretical measure may become unreliable on small sample datasets. At the same time, the computation of selected score function may also be unreliable on small sample datasets. Moreover, the CI-testing space and structure-searching space are so vast that heuristic methods have to be used. So, the two approaches are usually limited to find a local maxima.

To further enhance the accuracy and to try to overcome the local maxima problem in BN leaning, this paper proposes an ensemble BN structure learning

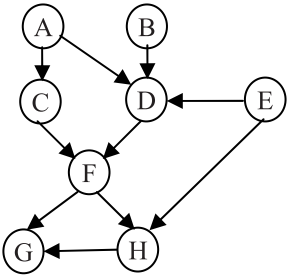
R. Alhajj et al. (Eds.): ADMA 2007, LNAI 4632, pp. 454–465, 2007.

approach that aims to achieve a better result in BN induction. In Section 2, we briefly introduces Bayesian network. In Section 3, we propose the overall process of our learning method. We present the details of our learning approach in sections 4, 5 and 6. In section 7, experimental results are compared and analyzed. Finally, we conclude our work in section 8.

2 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.

For example, Fig.1 shows the Bayesian network called *World* and the parameter table $\Theta_{H|\{E,F\}}$ of the node H. In the *World* network, the nodes A,B and E are root nodes which have not inarcs in the *World*.



E	F	H	P(H|E,F) $\theta_{x_i\|\pi_i}$
0	0	0	0.9
0	0	1	0.1
1	0	0	0.7
1	0	1	0.3
0	1	0	0.8
0	1	1	0.2
1	1	0	0.1
1	1	1	0.9

Fig. 1. An example of Bayesian network (*World*)

3 Ensemble BN Learning Overview

The overall process of our ensemble BN structure learning approach is shown in Fig.2.

Our approach belongs to the category of ensemble methods, "sub-sampling the training dataset".[3] Given the original training dataset D, our algorithm applies sample decomposition technique to generate several training sub-datasets D_i. From each generated training sub-dataset D_i, the component learner learns a component (Bayesian network) BN$_i$. Then, using components integration technique, these learned components (BNs) are combined into a result Bayesian network.

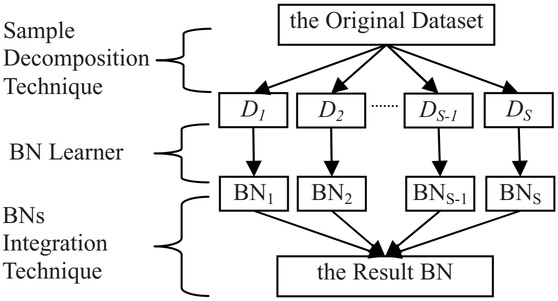


Fig. 2. Process of ensemble BN structure learning

4 Root Nodes Based Sample Decomposition Method

BN structure learning is to estimate conditional independences and dependences among the variables in the training dataset. The two most important sufficient conditions for BN structure learning are the Markov condition and the faithfulness condition. The joint probability distribution P over the training dataset and the true BN G satisfy the Markov condition if and only if under the distribution P, a node is independent of its non-descendant nodes given its parent nodes in the true BN G. The faithfulness condition means that all and only the conditional independence relations true in P are entailed by the Markov condition applied to G. [10]

Bootstrap sampling used by Bagging methods is a powerful tool for model selection. When learning the structure of a graphical model from limited datasets, such as the gene-expression datasets, Bagging methods [3] which use the bootstrap [4] sampling have been applied to explore the model structure [14],[13],[15], [16]. However, Bagging methods have several disadvantages over BN structure learning. On the one hand, the distributions generated from the sub-datasets obtained using Bagging methods and the true BN G may be unsatisfied with the Markov condition and the faithfulness condition. On the other hand, although Bagging methods may asymptotically converge to the true BN by re-sampling a large number of times, they require some convergence conditions. For example, the bagging method using non-parametric bootstrap sampling requires uniform convergence in the distribution of the bootstrap statistic as well as a continuity condition in the parameters.[14], [5]

To solve the above problems, we propose a novel sample decomposition method, which sub-samples the training dataset according to the values of root nodes in the true BN, for ensemble BN structure learning. The sample decomposition method is called Root Nodes based Sample Decomposition(RNSD).

4.1 Root Nodes Based Sample Decomposition

The detail of RNSD method is shown in Fig.3. The idea behind this decomposition method is based on the following 3 facts:

1. Learning BN from the sub-dataset sampled from the original training dataset by limitting the range of values for some root node to be a part of all the possible values for the root node, we expect to get all the Markov independences implied by the true BN.
2. Any arc, which is not connected with some root node in the true BN, can be learned on at least one sub-dataset sampled from the original training dataset by specifying values for the root node. So, learning on sampled sub-datasets that all the possible values of the specified root node are considered, we expect to learn all the "dependences" (arcs) of the true BN.
3. The joint marginal probability distributions of some nodes vary with the different sub-datasets sampled from the original training dataset by limitting the different ranges of the values for some root node.

Therefore, RNSD method can guarantee that all the Markov independences implied by the true BN can be learned from the sampled sub-datasets. Moreover, RNSD method can also hold the diversity in marginal probability distributions during components (BNs) learning.

```
1) Search root nodes $H_i$ ($i$=1,...,$L$) on the original dataset $D$ (Note: $L$ is the number of found root nodes);
2) Compute marginal probability tables of every found root node $\{P(H_i)|1 \leq i \leq L\}$;
3) Construct probability tables $T_s$ for all pairs and triples of the found root nodes $\{P(H_i,H_k), P(H_i,H_j,H_k) \mid 1 \leq i, j, k \leq L\}$,
   where $P(H_i,H_k) = P(H_i) * P(H_k), P(H_i,H_j,H_k) = P(H_i) * P(H_j) * P(H_k)$;
4) For each probability table $T_s$, sort the probability values by ascendant order;
5) For each probability table $T_s$, obtain the group $Gr(H_i,...,H_k)$ of the sub-datasets $D_{\{H_i \neq h_i \wedge ... \wedge H_k \neq h_k\}}$,..., by
   sub-sampling the dataset $D$ given $\{H_i \neq h_i \wedge ... \wedge H_k \neq h_k\}$, where $(h_i,...,h_k) \in T_s$;
6) For each group $Gr(H_i,...,H_k)$ of the sub-datasets, prune the sub-datasets which sampling rate is smaller than $\sigma$;
7) Prune the groups of sub-datasets which have few sub-datasets (such as the groups which only have no more
   than 2 sub-datasets)
```

Fig. 3. Root Nodes based Sample Decomposition method

Take the *World* network in Fig.1 as an example. Let $\sigma = 0.8$. Assume that the algorithm found the root nodes A, B and E in step 1. The sorted probability tables after step 4 are shown in Fig.4. In step 5, the groups of the sampled sub-datasets were generated. Finally, 4 groups of sub-datasets obtained after pruning in steps 6 and 7 are shown in Fig.5.

4.2 Correctness Proof

Definition 1. *Given a joint probability distribution P, X and Y are **conditional independent** given Z, denoted as $Ind(X{,}Y\ |Z)$, if and only if the following statement holds: $P(x \mid y, z) = P(x \mid z)$, $\forall x, y, z$ such that $P(yz) > 0$, where x, y and z denote an instantiation of the subsets of variables X, Y and Z, respectively.[11]*

Definition 2. *Given a joint probability distribution P, X and Y are **conditional dependent** given Z, denoted as $Dep(X{,}Y\ |Z)$, if and only if the following statement holds: $P(x \mid y, z) \neq P(x \mid z)$, $\exists x, y, z$ such that $P(yz) > 0$,*

A	$P(A)$
0	0.2
1	0.8

B	$P(B)$
1	0.1
0	0.9

E	$P(E)$
0	0.2
1	0.2
2	0.6

A	B	$P(A,B)$
0	1	0.02
1	1	0.08
0	0	0.18
1	0	0.72

A	B	E	$P(A,B,E)$
0	1	0	0.004
0	1	1	0.004
0	1	2	0.012
1	1	0	0.016
1	1	1	0.016
0	0	0	0.036
0	0	1	0.036
1	1	2	0.048
0	0	2	0.108
1	0	0	0.144
1	0	1	0.144
1	0	2	0.432

B	E	$P(B,E)$
1	0	0.02
1	1	0.02
1	2	0.06
0	0	0.18
0	1	0.18
0	2	0.54

A	E	$P(A,E)$
0	0	0.04
0	1	0.04
0	2	0.12
1	0	0.16
1	1	0.16
1	2	0.48

Fig. 4. Sorted probability tables

$Gr(A, B)$	Sampling rate
$D_{\{A\neq 0\wedge B\neq 1\}}$	0.98
$D_{\{A\neq 1\wedge B\neq 1\}}$	0.92
$D_{\{A\neq 0\wedge B\neq 0\}}$	0.82

$Gr(A, E)$	Sampling rate
$D_{\{A\neq 0\wedge E\neq 0\}}$	0.96
$D_{\{A\neq 0\wedge E\neq 1\}}$	0.96
$D_{\{A\neq 0\wedge E\neq 2\}}$	0.88
$D_{\{A\neq 1\wedge E\neq 0\}}$	0.84
$D_{\{A\neq 1\wedge E\neq 1\}}$	0.84

$Gr(B, E)$	Sampling rate
$D_{\{B\neq 1\wedge E\neq 0\}}$	0.98
$D_{\{B\neq 1\wedge E\neq 1\}}$	0.98
$D_{\{B\neq 1\wedge E\neq 2\}}$	0.94
$D_{\{B\neq 0\wedge E\neq 0\}}$	0.82
$D_{\{B\neq 0\wedge E\neq 1\}}$	0.82

$Gr(A, B, E)$	Sampling rate
$D_{\{A\neq 0\wedge B\neq 1\wedge E\neq 0\}}$	0.996
$D_{\{A\neq 0\wedge B\neq 1\wedge E\neq 1\}}$	0.996
$D_{\{A\neq 0\wedge B\neq 1\wedge E\neq 2\}}$	0.988
$D_{\{A\neq 1\wedge B\neq 1\wedge E\neq 0\}}$	0.984
$D_{\{A\neq 1\wedge B\neq 1\wedge E\neq 1\}}$	0.984
$D_{\{A\neq 0\wedge B\neq 0\wedge E\neq 0\}}$	0.964
$D_{\{A\neq 0\wedge B\neq 0\wedge E\neq 1\}}$	0.964
$D_{\{A\neq 1\wedge B\neq 1\wedge E\neq 2\}}$	0.952
$D_{\{A\neq 0\wedge B\neq 0\wedge E\neq 2\}}$	0.892
$D_{\{A\neq 1\wedge B\neq 0\wedge E\neq 0\}}$	0.856
$D_{\{A\neq 1\wedge B\neq 0\wedge E\neq 1\}}$	0.856

Fig. 5. Groups of sampled sub-datasets ($\sigma = 0.8$)

where x, y and z denote an instantiation of the subsets of variables X, Y and Z, respectively.[11]

Assume that the original training dataset D is data faithful to the true BN G.[11] We take the *World* network in Fig.1 as an example to prove the correctness.

Proposition 1. *Learning a BN from the sub-dataset sampled from the original training dataset by limitting the range of the values for some root node to be a part of all the possible values for the root node, we expect to obtain all the Markov independences implied by the true BN from the learned BN.*

Proof. Assume that we obtain the sub-dataset $D_{E\neq 0}$ from the original training dataset D by limitting the range of the values for the root node E to be $E \neq 0 \Leftrightarrow \{(E = 1) \cup (E = 2)\}$.

Assume that P denotes the distribution faithful to the true BN and P' denotes the distribution over the sub-dataset $D_{E\neq 0}$.

We take 2 cases to consider whether there exists the Markov independences implied by the true BN in the distribution P' over the sub-dataset $D_{E\neq 0}$.

1. For the nodes of which the parents set contains the root node E, for example the node D, there exists the Markov independence $Ind(D, C \mid A, B, E)$ in the true BN.

   According to the definition of conditional independence, there exists:

$$P(d \mid c, a, b, E = 1) = P(d \mid a, b, E = 1)$$
$$P(d \mid c, a, b, E = 2) = P(d \mid a, b, E = 2)$$

We can infer $P'(d \mid c, a, b, E = 2) = P'(d \mid a, b, E = 2)$.

According to the definition of conditional independence, we can infer that there exists $Ind(D, C \mid A, B, E)$ in the distribution P' over the sub-dataset $D_{E \neq 0}$.

2. For the nodes of which the parents set does not contain the root node E, for example the node G, there exists the Markov independences $Ind(G, C \mid F, H)$ and $Ind(G, E \mid F, H)$ in the true BN.

   According to the definition of conditional independence, there exists:

$$P(g \mid c, f, h, E = 1) = N_{gcfh1}/N_{cfh1} = P(g \mid f, h)$$
$$P(g \mid c, f, h, E = 2) = N_{gcfh2}/N_{cfh2} = P(g \mid f, h)$$
$$P(g \mid f, h, E = 1) = N_{gfh1}/N_{fh1} = P(g \mid f, h)$$
$$P(g \mid f, h, E = 2) = N_{gfh2}/N_{fh2} = P(g \mid f, h)$$

   We can infer that $P'(g \mid c, f, h) = P'(g \mid f, h)$.

   According to the definition of conditional independence, we can infer that there exists $Ind(G, C \mid F, H)$ in the distribution P' over the sub-dataset $D_{E \neq 0}$.

According to case 1, 2, we infer that Proposition 1 is correct.

Lemma 1. *Learning a BN from the sub-dataset sampled using RNSD method, we can obtain all the Markov independences implied by the true BN from the learned BN.*

Proof. Using the same way as Proposition 1, we can infer it.

Proposition 2. *Learning BNs from the sub-datasets which are sampled by by limitting the range of values for some root node to be a part of all the possible values for the root node, if all the possible values of the root node can be included in the sub-datasets, then we can obtain all the edges of the true BN from the learned BNs.*

Proof. There is an edge between the node C and the node F in Fig.1. We can get $Dep(C, F \mid A)$ and $Dep(C, F \mid A, E)$. According to the definition of conditional dependence, we can infer the following formula:

$$Dep(C, F \mid A, E) \Leftrightarrow \exists e \in E, Dep(C, F \mid A, E = e) \quad (1)$$

According to $Dep(C, F \mid A)_{D_{E \neq 0}} \Leftrightarrow Dep(C, F \mid A, E \neq 0)_D$ and the formula (1), we can infer the proposition.

According the above inferences, we can conclude that the BNs learned from the sub-datasets, which are sampled using our RNSD method, include all the Markov independences and edges implied by the true BN.

4.3 Root Nodes Search Method

The search method sees Fig.6.

1) Conduct order-0 CI tests for each pair nodes, then build the undirected graph UG_0;

2) Conduct order-1 CI tests for any three nodes X, Y and Z that in the undirected graph UG_0, X and Y, and Y and Z, are directly connected; and X and Z are not directly connected, if $Dep(X, Z \mid Y)$, then direct the edges $X \rightarrow Y$ and $Z \rightarrow Y$;

3) Find the maximal cliques $\{G_{a1},...,G_{ak}\}$ consisting of the nodes which have no inarcs, that every clique is undirected complete graph and has at least one outarc;

4) Order the maximal cliques by the number of nodes in ascendant order and prune the cliques G_{ai} that $\|G_{ai}\| > \delta$;

5) For every maximal clique, use the exhaustive search & Bayesian score function method to learn the root node in the maximal clique;

6) Detect and delete pseudo root nodes.

Fig. 6. Root nodes search method

The idea behind the search root nodes method is based on the following assumption, which is correct in most situations both for synthetic datasets and for real-life datasets:

Assumption 1. *If there exists a directed path $X \longmapsto Y$ between node X and node Y in a Bayesian network, then $Dep(X, Y \mid NULL)$.*

Under the above assumption, we can obviously infer that every clique obtained after the 4th step of the method has one and only one root node.

In most cases, Assumption 1 is satisfied. However, some exceptions may occur when there are many nodes (normally, the number of nodes including the two nodes on the path $\gg 5$) on the directed path between two nodes, that is, the two nodes may be independent conditional on NULL. Moreover, even if Assumption 1 is satisfied in any situation, some results after step 5 in Fig.6 may be pseudo root nodes on limited datasets.

We take one step to solve the pseudo root nodes problem. The step is to detect pseudo root nodes and delete them (see step 6 in Fig.6). The detection for pseudo root nodes is based on 2 kinds of independences. One kind of independences is the Markov independences given the obtained root node of other nodes in the maximal clique. The other kind of independences is the independences among root nodes. Firstly, if the first kind of independences given some found root node is not satisfied, then the found root node is pseudo root node and prune the pseudo root node. Secondly, if the second kind of independences is not satisfied by the two found root nodes, then at least one found root node is pseudo root node, and we prune the two root nodes.

For example, during the *World* network learning, the running result for every step of our root nodes searching method sees Fig.7.

Our root nodes search method does not have to find all the root nodes in the true BN, it is enough to find several root nodes in the true BN for our ensemble BN learning in terms of accuracy. We can also use other methods to search root nodes, such as the RAI algorithm [12].

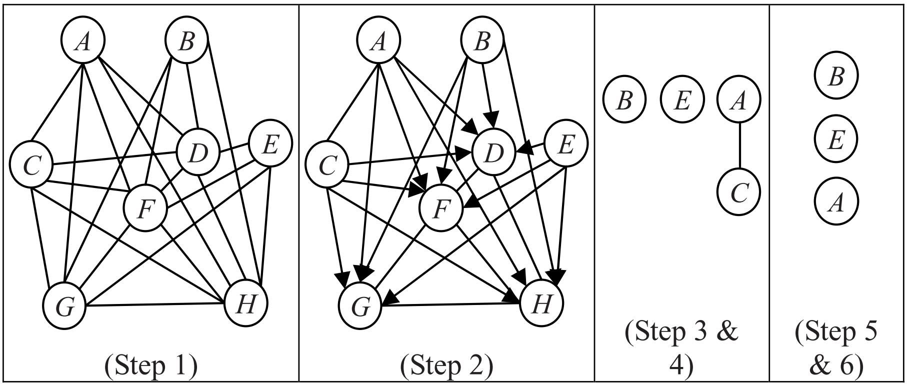


Fig. 7. Search root nodes in the *World* network

5 Bayesian Network Learner

Bayesian network learner is an individual BN learning algorithm and a building block of ensemble BN structure learning algorithm. Normally, it needs to be computationally efficient. Therefore, we selected OR algorithm [9], TPDA algorithm [1], and other algorithms using fast heuristic search (such as Greedy search) methods [8] as our BN learners. In our implementations for these algorithms, we applied partial nodes order information which was acquired by our root nodes search method.

6 Bayesian Networks Integration Method

Our integration method includes 2 parts: the integration of the Bayesian networks in the same group; the integration of the intergroup undirected networks.

6.1 Intragroup Bayesian Networks Integration

The method for intragroup Bayesian network integration is shown in Fig.8.

For any edge $\widetilde{e}$, ($\widetilde{e}$ is undirected edge of the arc e) in the BNs, consider the quantity:

$$P(\widetilde{e}) = \frac{1}{L} \sum_{i=1}^{L} 1\{\widetilde{e} \in BN_i\}$$

If $P(\widetilde{e}) > P(\widetilde{e}')$, then it is more probable that $\widetilde{e}$ exists in the true BN than $\widetilde{e}'$ does. Furthermore, if $P(\widetilde{e}) > 1/2$, we classify edge $\widetilde{e}$ as "true". Therefore, we can obtain the most probable BNs represented in the form of undirected network UG_i which every edge $\widetilde{e}$ has a probability table $g(e)$ of $\{\rightarrow, \leftarrow, \leftrightarrow\}$.

In the group Gr_i, assume we have learned L Bayesian networks $BN_1,\ldots,BN_L$:

1) For the Bayesian Networks $BN_1,\ldots,BN_L$, according to the principle of simple voting, compute the probability of every edge $\tilde{e}$ without considering orientation in the Bayesian Networks
2) Prune the edges whose probabilities are less than 1/2, and create a undirected network UG_i, which every edge $\tilde{e}$ has a probability table $g(e)$ about $\{\rightarrow,\leftarrow,\leftrightarrow\}$

Fig. 8. Intra-group Bayesian Networks Integration

6.2 Intergroup Undirected Networks Integration

Assume there are m groups of sub-datasets sampled using RNSD method. After intragroup Bayesian networks integration for every group $Gr_i (i = 1, \ldots, m)$, we obtained m undirected networks UG_i. For any possible arc e, consider the quantity:

$$P'(e) = \frac{1}{m}\sum_{i=1}^{m} p(e)1\{\tilde{e} \in UG_i, i = 1, \ldots, m\}$$

Note: Weight $p(e)$ is the probability value of $\{\rightarrow, \leftarrow\}$, where $p(\rightarrow) = g(\rightarrow)+g(\leftrightarrow)$ and $p(\leftarrow) = g(\leftarrow) + g(\leftrightarrow)$.

Finally, we take search method and score function to generate the result Bayesian network of our ensemble BN structure learning method. The process sees Fig.9.

Assume we have learned m undirected networks UG_i (i=1,...,m):

1) For the undirected networks UG_i (i=1,...,m), according to the principle of weighted voting, compute the probability value of every possible arc which exists in the undirected networks
2) Order arcs by the probability values in ascendant order, and apply exhaustive or heuristic search method and Bayesian score function to learn the maximal score Bayesian network by ‘adding arc’, ‘deleting arc’ and ‘reversing arc’ operators

Fig. 9. Inter-group Bayesian Networks Integration

7 Experimental Results

We implemented OR algorithm, OR-BWV algorithm, OR-HNSD algorithm, TPDA algorithm, TPDA-BWV algorithm, and TPDA-HNSD algorithm. OR-BWV and TPDA-BWV algorithms use Bagging sampling method and weighted voting integration method. Tests were run on a PC with Pentium4 1.5GHz and 1GB RAM. The operating system was Windows 2000. 4 Bayesian networks were used. From these networks, we performed experiments with 500, 1000, 5000

Table 1. Bayesian Networks

BN	Nodes Num	Arcs Num	Roots Num	Max In/Out-Degree	Domain Range
Alarm	37	46	12	4/5	2-4
Barley	48	84	10	4/5	2-67
Insur	27	52	2	3/7	2-5

Table 2. Average BDe(Alarm-OR)

SIZE	OR	OR-BSW	OR-RNSD
500	-15.1000	-14.9235	-13.8901
1000	-14.8191	-14.1917	-13.7610
5000	-13.9041	-13.8941	-13.1002

Table 3. Average BDe(Alarm-TPDA)

SIZE	TPDA	TPDA-BSW	TPDA-RNSD
500	-18.0973	-17.8045	-17.3650
1000	-15.4286	-15.0012	-14.2455
5000	-14.4960	-14.4731	-13.2682

Table 4. Average BDe(Barley-OR)

SIZE	OR	OR-BSW	OR-RNSD
500	-82.3448	-81.4104	-80.1943
1000	-78.9655	-78.2544	-76.2538
5000	-76.7081	-75.2273	-73.3371

Table 5. Average BDe(Barley-TPDA)

SIZE	TPDA	TPDA-BSW	TPDA-RNSD
500	-103.7931	-117.5443	-110.5973
1000	-111.0690	-112.2151	-103.5482
5000	-116.8919	-106.7328	-99.8341

Table 6. Average BDe(Insur-OR)

SIZE	OR	OR-BSW	OR-RNSD
500	-24.0167	-23.3812	-24.0010
1000	-22.6077	-21.5445	-22.5077
5000	-19.4286	-18.9523	-19.2286

Table 7. Average BDe(Insur-TPDA)

SIZE	TPDA	TPDA-BSW	TPDA-RNSD
500	-28.7857	-28.3471	-28.5172
1000	-25.1111	-24.8157	-25.0111
5000	-20.8571	-20.7916	-20.5571

training cases each. For each network and sample size, we sampled 10 original datasets and record the average results by each algorithm. Moreover, we applied Bagging sampling with 200 times in OR-BWV and TPDA-BWV algorithms. Let $\sigma = 0.8$ in Fig.3 and $\delta = 5$ in Fig.6.

We compared the accuracy of Bayesian networks learned by these algorithms according to the average BDeu score. The BDeu score corresponds to the posteriori probability of the structure learned.[7] The BDeu scores in our experiments were calculated on a seperate testing dataset sampled from the true BN containing 50000 samples. Tables 2-7 report the results.

There are several noticeable trends in these results. Firstly, as expected, as the number of instances grow, the quality of learned Bayesian network improves, except to TPDA for Barley network (500, 1000, 5000). It is due to that constraint-based method is unstable for limited datasets (500, 1000, 5000) relative to Barley network. At the same time, we can see that TPDA-BWV algorithm and TPDA-RNSD algorithm improve the stability of TPDA, that is, the quality of learned Bayesian networks by TPDA-BWV and TPDA-RNSD improves with the increase of sample size. Secondly, our RNSD based ensemble algorithms OR-RNSD and TPDA-RNSD are almost better than or at least equal to the individual Bayesian network learning algorithms in terms of accuracy on limited datasets.

Thirdly, in most cases, our ensemble algorithms have better performance than BWV ensemble algorithms. Finally, for Bayesian networks with few root node (such as Insur network), our ensemble algorithms have little improvement on learning accuracy. So, they are ineffective for these Bayesian networks.

8 Conclusion

We proposed a novel sampling technique and a components integration technique to incorporate ensemble learning into BN structure learning. Our results are encouraging in that they indicate that the our method achieved a more accurate result BN than individual BN learning algorithms.

References

1. 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)
2. 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)
3. Dietterich, T.G.: Machine Learning Research: Four Current Directions. AI Magazine 18(4), 745–770 (1997)
4. Davison, A.C., Hinkley, D.V.: Bootstrap Methods and Their Application, 1st edn. Cambridge Press, New York (1997)
5. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap, 1st edn. Chapman Hall, New York (1993)
6. Heckerman, D.: Bayesian Networks for Data Mining, 1st edn. Microsoft Press, Redmond (1997)
7. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian Networks: the Combination of Knowledge and Statistical Data. Machine Learning 20(3), 197–243 (1995)
8. Chickering, D.M., Heckerman, D., Geiger, D.: Learning Bayesian Network: Search Methods and Experimental Results. In: Fisher, D., Lenz, H. (eds.): Learning from Data: Artificial Intelligence and Statistics 5, Lecture Notes in Statistics 112, 143–153. Springer, New York (1996)
9. 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)
10. Pearl, J.: Causality: Models, Reasoning, and Inference, 1st edn. Cambridge Press, London (2000)
11. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search, 2nd edn. MIT Press, Massachusetts (2000)
12. Yehezkel, R., Lerner, B.: Recursive Autonomy Identification for Bayesian Network Structure Learning. In: Cowell, R.G., Ghahramani, Z. (eds.) AISTATS05. Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pp. 429–436. Society for Artificial Intelligence and Statistics, London (2005)

13. Friedman, N., Goldszmidt, M., Wyner, A.: On the Application of the Bootstrap for Computing Confidence Measures on Features of Induced Bayesian Networks. In: Heckerma, D., Whittaker, J. (eds.) Learning from Data: Artificial Intelligence and Statistics VII. Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics, pp. 197–202. Morgan Kaufmann, San Francisco (1999)
14. Friedman, N., Goldszmidt, M., Wyner, A.: Data Analysis with Bayesian Networks: A Bootstrap Approach. 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)
15. Friedman, F., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian Networks to Analyze Expression Data. Journal of Computational Biology 7, 601–620 (2000)
16. Pe'er, D., Regev, A., Elidan, G., Friedman, N.: Inferring Subnetworks from Perturbed Expression Profiles. Bioinformatics 17(Suppl. 1), 1–9 (2001)