Fast Markov Blanket Discovery Algorithm Via Local Learning within Single Pass

Shunkai Fu and Michel C. Desmarais

Ecole Polytechnique de Montreal,
C.P.6079, Succ. Centre-ville, Montreal, Quebec, Canada
{shukai.fu,michel.desmarais}@polymtl.ca

Abstract. Learning of Markov blanket (MB) can be regarded as an optimal solution to the feature selection problem. In this paper, an efficient and effective framework is suggested for learning MB. Firstly, we propose a novel algorithm, called Iterative Parent-Child based search of MB (IPC-MB), to induce MB without having to learn a whole Bayesian network first. It is proved correct, and is demonstrated to be more efficient than the current state of the art, PCMB, by requiring much fewer conditional independence (CI) tests. We show how to construct an AD-tree into the implementation so that computational efficiency is further increased through collecting full statistics within a single data pass. We conclude that IPC-MB plus AD-tree appears a very attractive solution in very large applications.

Keywords: Markov blanket, local learning, feature selection, single pass, AD-tree.

1 Introduction

Classification is a fundamental task in data mining and machine learning that requires learning a classifier through the observation of data. Basically, a classifier is a function that maps instances described by a set of attributes to a class label. How to identify the minimal, or close to minimal, subset of variables that best predicts the target variable of interest is known as feature (or variable) subset selection (FSS). In the past three decades, FSS for classification has been given considerable attention, and it is even more critical today in many applications, like biomedicine, where high dimensionality but few observations are challenging traditional FSS algorithms.

A principle solution to the feature selection problem is to determine a subset of attributes that can render the rest of all attributes independent of the variable of interest [8,9,16]. Koller and Sahami (KS) [9] first recognized that the Markov blanket (see its definition below) of a given target attribute is the theoretically optimal set of attributes to predict the target's value, though the Markov blanket itself is not a new concept and can be traced back to 1988 [11].

A Markov blanket of a target attribute T renders it statistically independent from all the remaining attributes, that is, given the values of the attributes in the Markov

S. Bergler (Ed.): Canadian AI 2008, LNAI 5032, pp. 96–107, 2008.

blanket, the probability distribution of T is completely determined and knowledge of any other variable(s) becomes superfluous [11].

Definition 1 (**Conditional independent**). Variable X and T are conditionally independent given the set of variables $\boldsymbol{Z}$ (bold symbol is used for set), iff. $P(T \mid X, \boldsymbol{Z}) = P(T \mid \boldsymbol{Z})$, denoted as $T \perp X \mid \boldsymbol{Z}$.

Similarly, $T \not\perp X \mid \boldsymbol{Z}$ is used to denote that X and T are **NOT** conditionally independent given $\boldsymbol{Z}$.

Definition 2 (**Markov blanket, *MB***). Given all attributes $\boldsymbol{U}$ of a problem domain, a Markov blanket of an attribute $T \in \boldsymbol{U}$ is any subset $\boldsymbol{MB} \subseteq \boldsymbol{U} \backslash \{T\}$ for which

$$\forall X \in \boldsymbol{U} \setminus \{T\} \setminus \boldsymbol{MB},\ T \perp X \mid \boldsymbol{MB}$$

A set is called **Markov boundary** of T if it is a minimal Markov blanket of T.

Definition 3 (**Faithfulness**). A Bayesian network G and a joint distribution P are faithful to one another, if and only if every conditional independence encoded by the graph of G is also present in P, i.e., $T \perp_G X \mid \boldsymbol{Z} \Leftrightarrow T \perp_P X \mid \boldsymbol{Z}$ [12].

Pearl [11] points out that: if the probability distribution over $\boldsymbol{U}$ can be faithfully represented by a Bayesian network (BN), which is one kind of graphical model that compactly represent a joint probability distribution among $\boldsymbol{U}$ using a directed acyclic graph, then the Markov blanket of an attribute T is unique, composing of the T's parents, children and spouses (sharing common children with T). So, given the faithfulness assumption, learning an attribute's Markov blanket actually corresponds to the discovery of its Markov boundary, and therefore can be viewed as selecting the optimal minimum set of feature to predict a given T. In the remaining text, unless explicitly mentioned, Markov blanket of T will refer to its Markov boundary under the faithfulness assumption, and it is denoted as $\boldsymbol{MB}(T)$.

$\boldsymbol{MB}(T)$ is trivial to obtain if we can learn a BN over the $\boldsymbol{U}$ first, but the BN's structure learning is known as NP-complete, and readily becomes non-tractable in large scale applications where thousands of attributes are involved. Until now, none of existing known BN learning algorithms claims to scale correctly over more than a few hundred variables. For example, the publicly available versions of the PC [12] and the TPDA (also known as PowerConstructor) [2] algorithms accept datasets with only 100 and 255 variables respectively.

The goal of this paper is to develop an efficient algorithm for the discovery of Markov blanket from data without having to learn a BN first.

2 Related Work

A reasonable compromise to learning the full BN is to discover only the local structure around an attribute T of interest. We refer to the conventional BN learning as *global learning* and the latter as *local learning*. Local learning of $\boldsymbol{MB}(T)$ is expected to remain a viable solution in domains with thousands of attributes.

Local learning of $\boldsymbol{MB}$ began to attract attention after the work of KS [9]. However, the KS algorithm is heuristic, and provides no theoretical guarantee of success. Grow-Shrink (GS) algorithm [10] is the first provably correct one, and, as indicated by its name, it contains two sequential phases, growing first and shrinking secondly. To improve the speed and reliability, several variants of GS, like IAMB, InterIAMB [15,16] and Fast-IAMB[17], were proposed. They are proved correct given the faithfulness assumption, and indeed make the $\boldsymbol{MB}$ discovery more time efficient, but none of them are data efficient. In practice, to ensure reliable independence tests, which is essential for this family of algorithm, IAMB and its variants decide a test is reliable when the number of instances available is at least five times the number of degree of freedom in the test. This means that the number of instances required by IAMB to identify $\boldsymbol{MB}(T)$ is at least exponential in the size of $\boldsymbol{MB}(T)$, because the number of degrees of freedom in a test is exponential with respect to the size of conditioning set, and the test to add a new node in $\boldsymbol{MB}(T)$ will be conditioned on at least the current nodes in $\boldsymbol{MB}(T)$ (Line 4, Table 1 in [8]).

Several trials were made to overcome this limitation, including MMPC/MB[14], HITON-PC/MB[1] and PCMB[8]. All of them have the same two assumptions as IAMB, i.e. faithfulness and correct independence test, but they differ from IAMB by taking into account the graph topology, which helps to improve data efficiency through conditioning over a smaller set instead of the whole $\boldsymbol{MB}(T)$ as done by IAMB. However, MMPC/MB and HITON-PC/MB are shown not always correct by the authors of PCMB since false positives will be wrongly learned due to non-complete conditional independence tests [8]. So, based on our knowledge, PCMB is the only one proved correct, scalable and represents a truly data-efficient means to induce the MB.

In this paper, we propose a novel MB local learning algorithm, called Iterative Parent-Child based search of Markov Blanket (IPC-MB). It is built on the same two assumptions of IAMB and PCMB. IPC-MB algorithm is compared with two of the algorithms discussed above: IAMB and PCMB. IAMB is a well known algorithm and referred to as MB local discovery. PCMB is the most successful break over IAMB to our knowledge and our own work is based on this algorithm.

Akin to PCMB, IPC-MB is designed to execute an efficient search by taking the topology into account to ensure a data efficient algorithm. We believe this approach is an effective means to conquer the data inefficiency problem occurring in GS, IABM and their variants. As its name implies, IPC-MB starts the search of $\boldsymbol{MB}(T)$ from its neighbors first, which actually are the parents and children of T, denoted as $\boldsymbol{PC}(T)$. Then, given each $X \in \boldsymbol{PC}(T)$, it further searches for $\boldsymbol{PC}(X)$ and checks each $Y \in \boldsymbol{PC}(X)$ to determine if it is the spouse of T or not. So, our algorithm is quite similar to PCMB, but it finds the $\boldsymbol{PC}$ of an attribute in a much more efficient manner. More detail about the algorithm can be found in Section 3. Considering that the discovery of $\boldsymbol{PC}(X)$ is a common basic operation for PCMB and IPC-MB, its efficiency will directly influence the overall performance of algorithm. Experiment results of algorithms comparison are reported and discussed in Section 5.

3 Local Learning Algorithm of Markov Blanket: IPC-MB

3.1 Overall Design

As discussed in Section 1 and 2, the IPC-MB algorithm is based on two assumptions, faithfulness and correct conditional test, from which the introduction and proof of this algorithm will be given.

Table 1. IPC-MB Algorithm

```
RecognizePC (T: target,
  ADJ_T :Adjacency set to search
  D: Dataset, ε:threshold)
{
1  NonPC = φ;
2  cutSetSize = 0;
3  do
4    for(each X ∈ ADJ_T) do
5      for(each S ⊆ ADJ_T \{X}
6          with |S| =cutSetSize) do
7        if(I_D(X_i,T|S)≤ε )then
8          NonPC = NonPC∪{X};
9          Sepset_{T,X} = S;
10         break;
11       end if
12     end for
13   end for
14   if(|NonPC|>0 )then
15     ADJ_T = ADJ_T \ NonPC ;
16     cutSetSize +=1;
17     NonPC = φ ;
18   else
19     break;
20   end if
21 while(|ADJ_T |> cutSetSize)
22 return ADJ_T ;
}
```

```
IPC-MB( D: Dataset, ε:threshold )
{
  // Recognize T'parents/children
1  CanADJ_T = U\{T};
2  PC =RecognizePC(T,CanADJ_T ,D ,ε);
3  MB = PC ;
4  for(each X ∈ PC ) do
//Recognize a true positive, and its
//parents/children as spouse candidates.
5    CanADJ_X = U\{X};
6    CanSP = RecognizePC(X,CanADJ_X, D,ε);
7    if(T ∉ CanSP )then
8      MB = MB\{X};
9      continue;
10   end if
     //Recognize true positives
11   for(each Y ∈ CanSP and Y ∉ MB )do
12     if(I_D(T,Y|Sepset_{T,Y} ∪ X) >ε) then
13       MB = MB∪{Y};
14     end if
15   end for
16 end for
17 return MB ;
}
```

On a BN over variables $\boldsymbol{U}$, the $\boldsymbol{MB}(T)$ contains parents and children of T, i.e. those nodes directly connected to T, and its spouses, i.e. parents of T's children. We denote these two sets as $\boldsymbol{PC}(T)$ and $\boldsymbol{SP}(T)$ respectively. With these considerations in mind, learning $\boldsymbol{MB}(T)$ amounts to deciding which nodes are directly connected

to T and which directly connect to those nodes adjacent to T (connect to T with an arc by ignoring the orientation).

As outline above, local learning of $\boldsymbol{MB}(T)$ amounts to (1) which nodes are adjacent to T among $\boldsymbol{U} \setminus \{T\}$, i.e. $\boldsymbol{PC}(T)$ here, and (2) which are adjacent to $\boldsymbol{PC}(T)$ and point to children of T in the remaining attributes $\boldsymbol{U} \setminus \{T\} \setminus \boldsymbol{PC}$, i.e. $\boldsymbol{SP}(T)$. This process is actually a breadth-first search procedure.

We need not care about the relations among $\boldsymbol{PC}(T)$, $\boldsymbol{SP}(T)$ and between $\boldsymbol{PC}(T)$ and $\boldsymbol{SP}(T)$, considering that we are only interested in which attributes belong to $\boldsymbol{MB}(T)$. Therefore, this strategy will allow us to learn $\boldsymbol{MB}(T)$ solely through local learning, reducing the search space greatly.

3.2 Theoretical Basis

In this section, we provide the theoretical background for the correctness of our algorithm.

Theorem 1. If a Bayesian network G is faithful to a probability distribution P, then for each pair of nodes X and Y in G, X and Y are adjacent in G iff. $X \not\perp Y \mid \boldsymbol{Z}$ for all $\boldsymbol{Z}$ such that X and $Y \notin \boldsymbol{Z}$. [12]

Lemma 1. If a Bayesian network G is faithful to a probability distribution P, then for each pair of nodes X and Y in G, if there exists $\boldsymbol{Z}$ such that X and $Y \notin \boldsymbol{Z}$, $X \perp Y \mid \boldsymbol{Z}$, then X and Y are **NOT** adjacent in G.

We get Lemma 1 from Theorem 1, and its proof is trivial. The first phase of IPC-MB, *RecognizePC* (Table 1), relies upon this basis. In fact, the classical structure learning algorithm PC [12, 13] is the first one designed on this basis.

Theorem 2. If a Bayesian network G is faithful to a probability distribution P, then for each triplet of nodes X, Y and W in G such that X and Y are adjacent to W, but X and Y are not adjacent, $X \rightarrow W \leftarrow Y$ is a sub-graph of G iff $X \not\perp Y \mid \boldsymbol{Z}$ for all $\boldsymbol{Z}$ such that X and $Y \notin \boldsymbol{Z}$, and $W \notin \boldsymbol{Z}$. [12]

Theorem 2 combined with Theorem 1 form the basis of IPC-MB's second phase, the discovery of T's spouses (Table 1). Given each $X \in \boldsymbol{PC}(T)$ learned via *RecognizePC*, we can learn $\boldsymbol{PC}(X)$ in a similar way as we learn $\boldsymbol{PC}(T)$. For each $Y \in \boldsymbol{PC}(X)$, if we known $T \not\perp Y \mid \boldsymbol{Z}$ for all $\boldsymbol{Z}$ such that T, $Y \notin \boldsymbol{Z}$ and $X \in \boldsymbol{Z}$, $T \rightarrow X \leftarrow Y$ is a sub-graph of G; therefore Y is a parent of X; since X is the common child between Y and T, Y is known as one spouse of T. This inference brings us Lemma 2.

Lemma 2. In a Bayesian network G faithful to a probability distribution P, given $X \in \boldsymbol{PC}(T)$, and $Y \in \boldsymbol{PC}(X)$, if $T \not\perp Y \mid \boldsymbol{Z}$ for all $\boldsymbol{Z}$ such that T, $Y \notin \boldsymbol{Z}$ and $X \in \boldsymbol{Z}$, then Y is a spouse of T.

3.3 Iterative Parent-Child Based Search of Markov Blanket

Learn parents/children

As the name of this algorithm indicates, the discovery of parent-child is the critical to the locality nature of this algorithm.

RecognizePC procedure (Table 1) is responsible for the search of parent/child candidates. It starts by connecting the current active target T (its first parameter) to all other nodes not visited by *RecognizePC* before, with non-oriented edges. Then, it deletes the edge (T, X_i) if there is any subset of $ADJ_T \setminus \{X_i\}$ conditioning on which T and X_i is independent based on the significance of a conditional independence test, I_D (line 5-12). X_i is removed finally at line 15.

In IPC-MB (discussed in the next section), *RecognizePC* appears at two different locations, line 2 and 6 respectively. This is designed to ensure that for each pair (X,Y), both *RecognizePC(X)* and *RecognizePC(Y)* will be called, and $X - Y$ is true only when $Y \in PC(X)$ and $X \in PC(Y)$, avoiding that any false nodes enter into $MB(T)$. Overall, this is similar to the conventional PC structure learning algorithm, but it limits the search to the neighbors of the target node. This is why local learning, instead of global learning, is possible and considerable time can be saved especially in applications with a large number of variables.

The correctness of our approach to find the parents and children of a specific node T is the basis for the whole algorithm, so the following theorem is defined.

Theorem 3. All parents and children of the node T of interest can be correctly recognized given the faithfulness assumption.

Proof. (i) A potential link between (T, X)*, where* X *is a candidate of* $\boldsymbol{PC}(T)$*, is kept only when there is no set* $\boldsymbol{S}$ *such that* T *and* $X \notin \boldsymbol{S}$*, and* $X \not\perp T \mid \boldsymbol{S}$*, i.e.* T *and* X *is conditional independent given* $\boldsymbol{S}$*. This is the direct application of Theorem 1, and this result guarantees that no false parent/child will be added into* $\boldsymbol{PC}(T)$ *given a sufficiently low* ε*; (ii) It is trivial to see that algorithm xxx above [provide a name or a reference because "our" is ambiguous] is exhaustive and covers all possible conditioning sets* $\boldsymbol{S}$*. (iii) Since algorithm RecognizePC always start by connecting* T *with all non-scanned nodes, it follows that no true positive that should be included will be missed by the algorithm. Therefore, all parents and children of* T *can be identified.*

Learn spouses

Learning of T's spouses involves two steps. For each candidate parent/child of T (line 4), *RecognizePC* (X) is called to collect X's parents and children, $\boldsymbol{PC}(X)$, as shown in lines 4-6 of *IPC-MB* procedure. If $T \notin \boldsymbol{PC}(X)$, then we just ignore the remaining part of current loop. If $T \in \boldsymbol{PC}(X)$, we know it is a true parent/child of T, and $\boldsymbol{PC}(X)$ contains the spouse candidates of T. Secondly, we begin to discover those true spouse candidates given Theorem 2 (lines 10-12).

Theorem 4. The result induced by *IPC-MB* is the complete Markov blanket of T .

Proof.(i) True parents/children can be returned by RecognizePC(T) if we use it correctly, as supported by Theorem 3 and our discussion above; (ii)Although some false spouses will be returned when we call Recognize(X), only true spouses that can satisfy the test of line 12 in IPC-MB, based on Theorem 2 and the underlying topology. Therefore, only true spouses will enter into ***MB*** *finally.*

4 All Dimension-Tree (AD-Tree)

Being a CI test-based algorithm, IPC-MB depends intensively on the tabulation of a joint frequency distributions, e.g. $C(X_1 = x_1 \wedge X_3 = x_3)$. At one extreme, we can look through the dataset to collect a specific co-occurrence table on demand, and another data pass for another a new query. This can be terribly time consuming considering thousands of CI tests are required (see our experiment example in Section 4), and it becomes worse quickly when the number of attributes increases, or the dataset becomes larger. In the implementation of IPC-MB, we try to cache as much statistics that can be expected given the current cutset size (see Table 1) as possible, aiming at reducing the data passes. It indeed works, but dozens of data passes still are necessary in our testing, which prohibits IPC-MB from being an economic candidate in large applications. An ideal solution we are looking for should be efficient not only in time, allowing all sufficient statistics to be collected in single data pass, but also in memory, at least scaling relative to the complexity of problems (i.e. number of attributes).

All Dimensions tree (AD-tree), proposed by Moore and Lee [18,19], represents such a solution. It is introduced for representing the cached joint frequency counting statistics for a categorical data set, from which we can query any co-occurrence table we need without having to go through the data repeatedly. Fig 1 is an example from [19], where attributes X_1, X_2 and X_3 have 2, 4 and 2 categories respectively. Each rectangular node in the tree stores the value of one conjunctive counting query, and they are called AD-nodes. The children of AD-node are called Vary nodes, displayed as ovals. Each corresponds to an attribute with index greater than that of its parent node.

A tree built in this way would be enormous for non-trivially sized problems, and its complexity increases quickly as the number of attributes and number of categories per attribute increase. However, considering that normally only a small percent of the all possible instances happens given attributes $\{ X_i \}$, the actual tree will be sparse very often, with many zero counts [18,19]. This characteristic allows a great reduction in memory consumption, and it is implemented in the IPC-MB algorithm. Readers can also consult the original references by the authors of AD-tree for alternative and potentially interesting techniques which we do not investigate here.

In this project, we refer IPC-MB with AD-tree as IPC-MB++, indicating that it is an enhanced version. Its algorithm specification is just same as IPC-MB (see Table 1)

Attributes	X_1	X_2	X_3
Case 1	1	1	1
Case 2	2	3	1
Case 3	2	4	2
Case 4	1	1	1
Case 5	2	3	1
Case 6	2	3	1

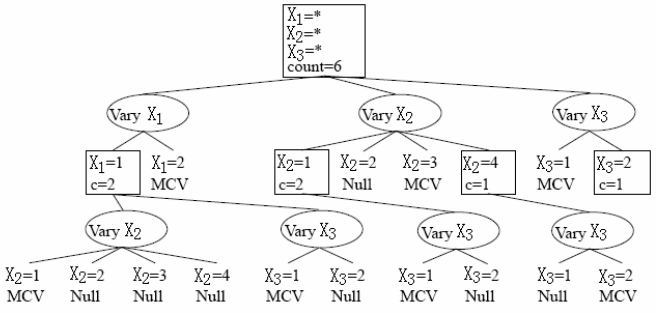

Fig. 1. Sample data with tree attributes and six records (Left), and its corresponding AD-tree (Right)

since we hide the details of tree construction and query, allowing readers focusing on the primary architecture.

5 Experiment and Analysis

5.1 Experiment Design

We only compare our algorithm with PCMB since interested readers can find the comparison of PCMB and IAMB in [8]. In the experiment, we use synthetic data sampled from known Alarm BN [7] which is composed of 37 nodes. The Alarm network is well-known as it has been used in a large number of studies on probabilistic reasoning. The network modeling situations arise from the medicine world. We run PCMB and IPC-MB with each node in the BN as the target variable T iteratively and, then, report the average performance when different size of data is given, including accuracy, data efficiency, time efficiency, scalability, and usefulness of information found.

5.2 Evaluation and Analysis

One of the basic assumptions of these three algorithms is that the independence tests are valid. To make them PCMB and IPC-MB, feasible in practice, we perform a test to check if the conditional test to do is reliable, and skip the result if not. As indicated in [15], IAMB considers a test to be reliable when the number of instances in D is at least five times the number of degrees of freedom in the test. PCMB follows this standard in [8], and so does our algorithm IPC-MB to maintain a comparable experiment result.

Accuracy and data efficiency

We measure the accuracy of induction through the precision and recall over all the nodes for the BN. **Precision** is the number of true positives in the returned output divided by the number of nodes in the output. **Recall** is the number of true positives in

the output divided by the number of true positives known in the true BN model. We also combine precision and recall as

$$\text{Distance} = \sqrt{(1-precision)^2 + (1-recall)^2}$$

to measure the Euclidean **distance** from precision and recall[8].

Table 2. Accuracy comparison of PCMB and IPC-MB over Alarm network

Instances	Algorithm	Precision	Recall	Distance
1000	PCMB	.76±.04	.83±.07	.30±.06
1000	IPC-MB	.92±.03	.84±.03	.18±.04
2000	PCMB	.79±.04	.91±.04	.23±.05
2000	IPC-MB	.94±.02	.91±.03	.11±.02
5000	PCMB	.80±.05	.95±.01	.21±.04
5000	IPC-MB	.94±.03	.95±.01	.08±.02
10000	PCMB	.81±.03	.95±.01	.20±.03
10000	IPC-MB	.93±.02	.96±.00	.08±.02
20000	PCMB	.81±.02	.96±.00	.20±.01
20000	IPC-MB	.93±.03	.96±.00	.08±.02

Table 2 shows the average precision, recall and distance performance about PCMB and IPC-MB given different size of data sampled from the Alarm network. From which, we notice that PCMB is worse than IPC-MB, which can be explained by its search strategy of minimum conditioning set. It needs to go through conditioning sets with size ranging from small to large, so PCMB has the similar problem like IAMB when conditioned on large set. However, IPC-MB's strategy, always conditioning on smallest conditioning set and removing as many as possible true negative ones first, prevents it from this weakness. Therefore, IPC-MB has higher accuracy rate compared with PCMB given the same size training data, and this also reflects IPC-MB's advantage on data efficiency.

Time efficiency

To measure time efficiency, we refer to the number of data pass and CI test occurring in PCMB, IPC-MB and the enhanced version, IPC-MB++ (IPC-MB plus AD-tree). One data pass corresponds to the scanning of the whole data for one time. In PCMB and IPC-MB, we only collect all the related statistics information (consumed by CI tests) that can be expected currently. However, in IPC-MB++, we collect the full statistics before the learning begins. In Table 3, "# rounds" refers to the total number of data passes we need to finish the MB induction on all the 37 nodes of Alarm BN. "# CI test" is defined similarly. Generally, the larger are these two numbers, the slower is the algorithm.

As Table 3 shows, in this study, IPC-MB requires less than 10% and 60% of the total amount of data passes and CI tests done by PCMB respectively. Compared with IPC-MB, IPC-MB++ needs only one data pass during the whole running procedure, but same CI tests. This is quite an attractive merit if we recognize the time spent in data scanning is quite consuming, especially when we have large observations and it is impossible to store them all in memory.

Table 3. Comparison of time complexity required by different MB induction algorithms, in terms of number of data pass and CI test

Instances	Algorithm	#rounds	#CI test
5000	PCMB	46702±6875	114295±28401
5000	IPC-MB	446±15	34073±1996
5000	IPC-MB++	1±0	34073±1996
10000	PCMB	46891±3123	108622±13182
10000	IPC-MB	452±12	37462±1502
10000	IPC-MB++	1±0	37462±1502
20000	PCMB	48173±2167	111100±9345
20000	IPC-MB	460±9	40374±1803
20000	IPC-MB++	1±0	40374±1803

Scalability

IAMB and its variants are proposed to do feature selection in microarray research [14, 15]. From our study, it is indeed a fast algorithm even when the number of features and number of cases become large. Reliable results are expected when there are enough data. PCMB is also shown scalable by its author in [8], where it is applied to a KDD-Cup'2001 competition problem with 139351 features. Due to the short of such large scale observation, we haven't tried IPC-MB(++) in the similar scenario yet. However, our empirical study, though there are only 37 variables, have shown that IPC-MB(++) runs faster than PCMB in terms of the amount of CI test and data pass. Therefore, we have confidence to do this inference that IPC-MB(++) can also scale to thousands of features as IAMB and PCMB claim. Besides, due to the relative advantage on data efficiency among the three algorithms, IPC-MB(++) is supposed to work with best results in challenging applications where there is large number of features but small amount of samples.

Usefulness of information found

Markov blanket contains the target's parents, children and spouses. IAMB and its variants only recognize that variables of MB render the rest of variables on the BN independent of target, which can be a solution to the feature subset selection. Therefore, IAMB only discovers which variables should fall into the Markov blanket, without further distinguishing among spouse/parents/children. PCMB and IPC-MB(++) goes further by discovering more topology knowledge. They not only learn MB, but also distinguish the parents/children from the spouses of target. Among parents/children, those children shared by found spouses and the target are also separated (the v-structures found).

6 Conclusion

In this paper, we propose a new Markov blanket discovery algorithm, called IPC-MB. It is based on two assumptions, DAG-faithful distribution and correct independence test. Like IAMB and PCMB, IPC-MB belongs to the family of local learning of MB, so it is scalable to applications with thousands of variables but few instances. It is

shown correct, and much more data-efficient than IAMB and PCMB, which allows it perform much better in learning accuracy than IAMB given the same amount of instances in practice. Compared with PCMB, IPC-MB(++) provides a more efficient approach for learning, requiring much fewer number of CI tests and data passes than PCMB. Therefore, we can state that IPC-MB(++) shows a high potential as a practical MB discovery algorithm, and is a good tradeoff between IAMB and PCMB.

References

1. Aliferis, C.F., Tsamardinos, I., Statnikov, A.: HITON, a Novel Markov blanket algorithm for optimal variable selection. In: Proceedings of the 2003 American Medical Informatics Association Annual Symposium, pp. 21–25 (2003)
2. Cheng, J., Greiner, R.: Learning Bayesian networks from data: An information-theory based approach. Artificial Intelligence 137, 43–90 (2002)
3. Cheng, J., Greiner, R.: Compared Bayesian Network classifiers. In: Proceedings of the 15th Conference on UAI (1999)
4. Cheng, J., Bell, D.A., Liu, W.: Learning belief networks from data: An information theory based approach. In: Proceedings of the sixth ACM International Conference on Information and Knowledge Management (1997)
5. Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence 42, 395–405 (1990)
6. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29, 131–163 (1997)
7. Herskovits, E.H.: Computer-based probabilistic-network construction. Ph.D Thesis, Stanford University (1991)
8. Pena, J.M., Nilsson, R., Bjorkegren, J., Tegner, J.: Towards scalable and data efficient learning of Markov boundaries. International Journal of Approximate Reasoning 45(2), 211–232 (2007)
9. Koller, D., Sahami, M.: Toward optimal feature selection. In: Proceedings of International Conference on Machine Learning, pp. 284–292 (1996)
10. Margaritis, D., Thrun, S.: Bayesian network induction via local neighborhoods. In: Proceedings of NIPS (1999)
11. Pearl, J.: Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, San Francisco (1988)
12. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Lecture Notes in Statistics. Springer, Heidelberg (1993)
13. Spirtes, P., Glymour, C.: An algorithm for Fast Recovery of Sparse Casual Graphs. Philosophy Methodology Logic (1990)
14. Tsamardinos, I., Aliferis, C.F., Statnikov, A.: Time and sample efficient discovery of Markov blankets and direct causal relations. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 673–678 (2003)
15. Tsamardinos, I., Aliferis, C.F.: Towards principled feature selection: Relevancy, filter and wrappers. In: 9th International Workshop on Artificial Intelligence and Statistics (AI&Stats 2003) (2003)
16. Tsamardinos, I., Aliferis, C.F., Stantnikov, A.: Time and sample efficient discovery of Markov blankets and direct causal relations. In: Proceedings of SIGKDD 2003 (2003)

17. Yaramakala, S., Margaritis, D.: Speculative Markov blanket discovery for optimal feature selection. In: Proceedings of IEEE International Conference on Data Mining (ICDM) (2005)
18. Moore, A., Lee, M.S.: Cached sufficient statistics for efficient machine learning with large datasets. Journal of Artificial Intelligence Research 8, 67–91 (1998)
19. Komarek, P., Moore, A.: A dynamic adaptation of AD-trees for efficient machine learning on large data sets. In: Proceedings of ICML (2000)