Proceedings of the World Congress on Engineering 2010 Vol I
WCE 2010, June 30 - July 2, 2010, London, U.K.

Feature Selection by Efficient Learning of Markov Blanket

Shunkai Fu and Michel Desmarais

***Abstract*—Markov blanket was proved as the theoretically optimal feature subset to predict the target. IPC-MB was firstly proposed in 2008 to induce the Markov blanket via local search, and it is believed important progress as compared with previously published work, like IAMB, PCMB and PC. However, the proof appearing in its first publication is not complete and sound enough. In this paper, we revisit IPC-MB with discussion as not found in the original paper, especially on the proof of its theoretical correctness. Besides, experimental studies with small to large scale of problems (Bayesian networks) are conducted and the results demonstrate that IPC-MB achieves much higher accuracy than IAMB, and much better time efficiency than PCMB and PC.**

***Index Terms*—Feature selection, Markov Blanket, IPC-MB.**

I. Introduction

The Markov Blanket (MB) of one target T, denoted as $MB(T)$, was realized as theoretically optimal feature subset to predict the value of T by Koller and Sahami in 1996 [1], though the concept Markov Blanket itself, in fact, can be traced back to even earlier time 1988 [2]. One approximate algorithm to induce $MB(T)$was proposed by Koller and Sahami in [1], and it is referred as KS algorithm by the initials of authors. Since then, with problems involved with many features becoming common, several works on feature selection via the induction of $MB(T)$ has appeared, including GS [3], IAMB and its variants [4, 5], MMPC/MB [5], HITON-PC/MB [6], Fast-IAMB [7], PCMB [8] and the latest one IPC-MB [9].

Shunkai Fu is with the Computer Engineering Department of Ecole Polytechnique de Montreal, Canada, and the College of Computer Science and Technology, Donghua University, P.R. China (e-mail: shunkai.fu@ polymtl.ca)

Michel Desmarais is with t the Computer Engineering Department of Ecole Polytechnique de Montreal, Canada (e-mail: Michel.desmarais@polymtl.ca).

Given the faithfulness assumption (see **Definition 1**), $MB(T)$ is known as unique, and it owns two critical properties: (1) Given the full knowledge of $MB(T)$,T is independent with any $X \in \mathbf{U} \setminus MB(T) \setminus \{T\}$, where **U** contains all features in the observations; and (2) $MB(T)$ is composed of T's parents, children and spouses. KS, GS, IAMB and its variants (referred as **GROUP I**) are proposed based on the first property. In contrast, MMPC/MB, HITON-PC/MB, PCMB and IPC-MB (saying **GROUP II**) are built on the second property, i.e. the so-called topology structure. The most advantage of **GROUP II** over **GROUP I** is known as data efficiency, and it greatly influence the actual precision and recall performance in practice between them.

Among the four known algorithms of GROUP II, unfortunately, MMPC/MB and HITON-PC/MB are shown with no guarantee to produce correct outcomes always [8]. PCMB is the first one proved correct and demonstrated with satisfactory data efficiency than those of GROUP I, but it loses to more recently proposed IPC-MB in terms of time as well as data efficiency [9, 10]. As introduced firstly by Fu in 2008 [9], though IPC-MB indeed has the potential to prevail over PCMB and achieve the best trade-off among existing algorithms for inducing Markov blanket but without having to learn the whole Bayesian network first, it is noticed that the proof of IPC-MB is not complete. Therefore, we revisit IPC-MB here by re-explaining how it works, and proving that it guarantees to produce correct results theoretically.

Section 2 contains necessary theory foundation and the overall description of IPC-MB. In Section 3, the specification and proof of each step as involved are covered. Experimental studies are conducted over IAMB, PCMB, IPC-MB and PC in Section 4. We conclude our work and discovery in Section 5.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Theory Foundation and Overall Description

A. Theory Foundation

If the probability distribution over **U** can be faithfully represented by a Bayesian network, $MB(T)$ is unique and composed of T's parents ($Pa(T)$), children ($Ch(T)$) and spouses ($Sp(T)$) [2], i.e. $MB(T) = Pa(T) \cup Ch(T) \cup Sp(T)$.

Definition 1 (Faithfulness Condition) A Bayesian Network G and a joint distribution P are faithful to one another iff. every conditional independence entailed by the graph G and the Markov Condition (referred as Local Markov Property, ***Theorem 3***) is also present in P [2].

Theorem 1 If a Bayesian network G is faithful to a joint probability distribution P, then (1) there is an edge between the pair of nodes X and Y in G iff. X and Y are conditionally dependent given any other set of nodes, and (2) for each triplet of nodes X, Y and Z in G such that X and Y are adjacent to Z but X is not adjacent to Y, $X \rightarrow Z \leftarrow Y$ is a sub-graph of G iff. X and Y are dependent conditioned on every other set of nodes that contains Z [11].

Faithfulness assumption sets up a connection between probability distribution and topology structure, on which **Theorem 1** is built and plays as one important reference for algorithms of **GROUP II**. **A**lgorithms of **GROUP II** induce $MB(T)$ via the recognition of connections as exist between (1) T and $Pa(T)$, (2) T and $Ch(T)$, and (3) the V-structure $T \rightarrow X \leftarrow Y$ where $X \in Ch(T)$and $Y \in Sp(T)$. By **Theorem 1**, recognizing connections of interest then becomes as a series of recognitions that if X and Y are conditionally independent as conditioned on $\mathbf{Z}$ ($\emptyset \subseteq \mathbf{Z} \subseteq \mathbf{U}$). We use $I(X, Y|\mathbf{Z})$ to denote this conditional independence relation, and $I_D(X, Y|\mathbf{Z})$ as some statistical test employed using observations D.

B. Overall Architecture

Although IPC-MB can be grouped into the category of HITON-PC/MB, MMPC/MB and PCMB, it differs from those three in term of search strategy: IPC-MB proceeds by removing non-MB variables iteratively, with true ones left, while the other three directly determine which ones should be included. Because IPC-MB starts to filter out true negatives with empty conditioning set on, and increase the size of conditioning set with one in each iteration (see *RecognizePC* in Fig. 1), negatives are removed by lowest-order tests with priority. By doing so, decisions made via the results of statistical tests are ensured with maximum confidence, improving the overall reliability of the algorithm especially when the sample size is limited in practice. This is meaningful since algorithms based on statistical tests suffer most from the curse dimensionality. In contrast, the others (HITON-PC/MB, MMPC/MB and PCMB) typically check all possible CI tests, involving small to large conditioning set, to decide whether or not to absorb each variable as candidate MB member.

```
IPC − MB(T: Target, D: Dataset, ε: Significance)
{  // Recognize T's parents and children
1.  CanADJ_T = U\{T};      // Candidate adjacency set
2.  CanPC(T) = RecognizePC(T, CanADJ_T, D, ε);
3.  PC(T) = {};
4.  for(∀X ∈ CanPC(T)) do
5.     CanADJ_X = U\{X};
6.     CanPC(X) = RecognizePC(X, CanADJ_X, D, ε)
7.     if(T ∈ CanPC(X))then
8.         PC(T) = PC(T) ∪ {X};
9.         CanSP_T,X = CanPC(X);
10.    end if
11. end for
12. MB(T) = PC(T);
    // Recognize T's spouses
13. for(∀X ∈ PC(T)) do
14.    for (∀Y ∈ CanSP_T,X and Y ∉ MB(T)) do
15.        if(I_D(T, Y|Sepset_T,Y ∪ {X}) > ε) then
16.            MB(T) = MB(T) ∪ {Y};
17.        end if
18.    end for
19. end for
20. return MB(T);
}
```

```
RecognizePC(T, ADJ_T, D, ε)
{
1.  NonPC = ∅;
2.  cutSetSize = 0;
3.  repeat
4.    for( ∀X ∈ ADJ_T) do
5.      for(S ⊆ ADJ_T\{X}) with |S| = cutSetSize) do
6.        if(I_D(T, X|S) ≤ ε) then
7.          NonPC = NonPC ∪ {X};
8.          Sepset_T,X = S;
9.            break;
10.       end if
11.     end for
12.   end for
13.   ADJ_T = ADJ_T\NonPC;
14.   NonPC = ∅;
15.   cutSetSize + +;
16. until (|ADJ_T| ≤ cutSetSize)
17. return ADJ_T;
}
```

Fig. 1. Pseudo code of IPC-MB and RecognizePC.

The whole procedure of IPC-MB (Fig. 1) can be divided into two phases:

- Firstly, it recognizes those directly connected to T, i.e. parents and children mixed, and they are denoted as $PC(T)$(Line 1 – 12); Then,

- It recognizes the spouses of T as exist among $\cup_{X \in PC(T)} CanPC(X)$, denoted as $Sp(T)$ (Line 13 – 20). Note that $X \in PC(T) \cap Sp(T)$ is recognized as $PC(T)$ with priority and would be added into $MB(T)$ in the first step.

III. Specification and Proof of Soundness

A. Learn Candidate Parents/Children

As the name indicates, the discovery of parent/child is critical to IPC-MB. In this section, we introduce the core part of IPC-MB, *RecognizePC(T),* which returns to us to one candidate parent/children set, denoted as $CanPC(T)$. It not only contains T's parents and children, but some false positives possibly. How to remove these false ones to get the exact $PC(T)$ is discussed in next section.

RecognizePC starts by assuming that the target T is dependent with all $X \in ADJ_T$, i.e. there is one "virtual" edge connecting T and each X. Then, it determines whether each arc $T - X$ should be removed. When $I_D(T, X|\mathbf{S})$ fails with some $\mathbf{S} \subseteq ADJ_T, X$ is deleted from ADJ_T, just like $T - X$ being deleted. Since the conditioning set **S** begins with empty set on, any recognizable false positive X is always deleted given the minimum conditioning set, which ensures data efficiency.

Theorem 2 Under the assumptions that the independence tests are correct and that the learning data D is an independent and identically distributed sample from a probability distribution P faithful to a DAG G, given $ADJ_T = \mathbf{U} \backslash \{T\}$, *RecognizePC* enables us to find the superset of $PC(T)$, denoted as $CanPC(T)$, and it has two properties: (1) for each $X \in PC(T)$, $X \in CanPC(T)$*;* and (2) $PC(T) \sqsubseteq CanPC(T)$.

Proof. *We prove the first property by contradiction. With $ADJ_T = \mathbf{U} \backslash \{T\}$, we assumed that there is some $X \in PC(T)$ but not output by RecognizePC. According to* ***Theorem 1****, we have such a fact that if $X \in PC(T)$, it should NOT be independent with T given any conditioning set, i.e. it should pass all $I_D(T, X|\mathbf{S})$ as met in RecognizePC. Therefore, X would not be output by RecognizePC only when X is not connected with T, which is obviously contradictory with the fact that $X \in PC(T)$. Therefore, all $X \in PC(T)$ would be returned by RecognizePC. The second one is illustrated by* ***Lemma 2*** (refer below). ■

Theorem 2 concludes the contribution of *RecognizePC*, i.e. *RecognizePC* outputs a superset of the target $PC(T)$. Before discuss how to filter out those un-expected variables, it is necessary to study what they are and how it happens.

Definition 2 (**Descendant**) Y is a descendant of X, if there exists a directed path from X to Y, but there exists no directed path from Y to X. Descendants of X is denoted as $Des(X)$ in the remaining text.

Definition 3(**Non-Descendant**) Given all variable set **U**, those other than descendants are known as non-descendants of X, denoted as $ND(X)$. $ND(X) = \mathbf{U} \backslash Des(X) \backslash \{T\}$.

Theorem 3 $I(X, ND(X) \backslash Pa(X) | Pa(X))$, i.e. X is independent with $ND(X) \backslash Pa(X)$ given the full knowledge of $Pa(X)$ (**Local Markov Property**) [2].

Lemma 1 Given T and $ADJ_T = \mathbf{U} \backslash \{T\}$, the output of *RecognizePC* will NOT contain $ND(T) \backslash Pa(T)$.

Proof. *(1) $I(T, ND(T) \backslash Pa(T) | Pa(T))$ by* ***Theorem 3****; (2) $Pa(T)$ will always stay in ADJ_T by* ***Theorem 2****; (3) The conditioning set starts with ∅ on, so we are guaranteed to have chance to be conditioned on $Pa(T)$ when $cutSetSize = |Pa(T)|$; (4) We check each $X \in ADJ_T$ in each iteration, and $ND(T) \subseteq ADJ_T$. Therefore, each $X \in ND(T) \backslash Pa(T)$ is able to be recognized and deleted from ADJ_T due to $I_D(T, X|Pa(T)) \leq \varepsilon$.* ■

Lemma 2 Given T and $ADJ_T = \mathbf{U} \backslash \{T\}$, the output of *RecognizePC* may contain descendants of T.

Due to the limit of space, interesting readers can refer [8] for such examples. In the next section, we will discuss how to construct a true parent-children set of T, i.e. $PC(T)$, by filtering those false positives as may output by *RecognizePC(T)*.

B. Learn Parents/Children

As we discussed above, *RecognizePC(T)* may output some false positives, and they can only be descendants of T (combine **Lemma 1** and **Lemma 2**). Foutunately, filtering out these false positives from $CanPC(T)$ is trivial. Given $\forall X \in CanPC(T)$, we check if $T \in CanPC(X)$ (Line 7, *IPC-MB*) via repeatedly calling *RecognizePC(X)* (Line 6, *IPC-MB*) to get $CanPC(X)$. If $T \in CanPC(X)$, X is known as one true parent/child, and it is added into $PC(T)$ (Line 8, *IPC-MB*). Otherwise, it is ignored and won't enter into $PC(T)$.

Lemma 3 With $CanPC(T)$ = *RecognizePC*(T) ready, given $\forall X \in CanPC(T)$ and $CanPC(X)$ = *RecognizePC*(X), (1) if $T \in CanPC(X)$, X is known as a true parent/children, and should be added into $PC(T)$ (Line 7-10, IPC-MB); (2) otherwise, X is known as a false parent/children, and should be ignored.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Proof. *From* ***Theorem 2*** *and* ***Lemma 2****, it is known that* $CanPC(T)$ *may contain two types of attributes: true parents/children of* T *as expected, and descendants of* T *that is not desired. Given* $\forall X \in CanPC(T)$*, we need to prove that: what true is still recognized as true, but what false can be successfully filtered out.*

Given $X \in PC(T)$*, obviously* $T \in PC(X)$ *and* T *definitely would be returned by RecognizePC(X) given* ***Theorem 2****. So, it is safe to add such* X *into* $PC(T)$*.*

Given $X \in Des(T)$*, we know that* $CanPC(X)$ *may contain* $PC(X)$ *plus some of* X*'s descendants possibly (****Lemma 2****). Since* $T \notin PC(X)$*,* T *may be output by ReconigzePC(X) only because it is* X*'s descendant. If this is true, then* T *is its own descendant's descendant, which is impossible because one cycle happens. Hence, given* $X \in CanPC(T)$ *but* $T \notin CanPC(X)$*,* X *is known as false positive and should not be added to* $PC(T)$*.* $\blacksquare$

Theorem 4 Under the assumptions that the independence tests are correct and that the learning data D is an independent and identically distributed sample from a probability distribution P faithful to a DAG G, IPC-MB allows us to find the complete and correct parents and children about T of interest.

Proof. *Given* ***Theorem 2****,* ***Lemma*** *1,* ***Lemma 2****,* ***Lemma*** **3** *and the fact that we call RecognizePC(X) for* each $X \in CanPC(T)$*, the proof is trivial.* $\blacksquare$

Therefore, by the Line 12 of IPC-MB, we have a true $PC(T)$, and it is noticed that the learning is built on a series of *RecognizePC(X)* , which exactly explains why the algorithm is called Iterative Parent-Child based learning of Markov Blanket (IPC-MB). What left is the learning of T's spouses, i.e. $Sp(T)$. How to recognize $Sp(T)$ is discussed in the section below, but it is necessary to predict that it depends on the output of *RecognizePC(X)* as well.

C. Learn Spouse

By the Line 12 of IPC-MB (Fig. 1), we have $MB(T) = PC(T)$ as discussed in last section. In fact, we also have collected all candidate spouses of T during the repeated calls of *RecognizePC(X) given* $X \in CanPC(T)$ (Line 4-11, IPC-MB).

Lemma 4 Given $X \in CanPC(T)$, if $T \in CanPC(X)$, $CanPC(X)$ contains candidate spouses of T if there are.

Proof. ***Theorem 2*** *tells us that* $CanPC(X)$ *contains all parents/children of* X*. Given* $X \in CanPC(T)$*, if* $T \in CanPC(X)$*, then* X *is known as a true parent/child (***Lemma 3***). If* X *is a common child of* T and Y*,* Y *is known as* T*'s spouse, and it should be contained in* $CanPC(X)$*. This applies to all such Y, so all of them should be contained in* $CanPC(X)$*.* $\blacksquare$

All outputs of *RecognizePC*(X) regarding to each $X \in PC(T)$ are cached as $CanSP_{T,X}$ (Line 9, IPC-MB) with index (T,X) for later reference. Obviously, they contain more than what we expect:

- T, since $T \in CanPC(X)$, which would be ignored (Line 14, IPC-MB);
- True parents and/or children of T, which would be ignored as well since they are already in *MB(T)* (Line 14, IPC-MB), requiring no extra effort;
- True spouses of T, i.e. those having X as their child as T. These are what we are interested to distinguished here;
- False positives, neither parents, children nor spouses of T. T's descendants, if there are in $CanPC(T)$, would be ignored to save computing resource (Line 14, IPC-MB) since they are impossibly to be T's spouses .

Lemma 5 Given $CanPC(T)$=*RecognizePC*(T), $Sp(T) \sqsubset \cup_X CanPC(X)$, where $X \in CanPC(T)$ and $T \in CanPC(X)$, i.e. $Sp(T) \sqsubset \cup_{X \in PC(T)} CanPC(X)$.

Proof. *Given* ***Lemma 4****, the proof is trivial.* $\blacksquare$

With **Lemma 5**, it is known that $\cup_{X \in PC(T)} CanPC(X)$ contain all candidate spouses of T, by Line 12 of *IPC-MB*, and they are denoted with shorthand $CanSP(T)$. Similarly to the discovery of $PC(T)$, we depend on the underlying connectivity information to recognize $Sp(T)$ from $CanSP(T)$. For any $X \in Sp(T)$, there are two facts available for reference: (1) it must belong to $PC(Y)$ for some $Y \in PC(T)$, where Y is the common child of X and T; (2) it is independent with T as conditioned on $Sepset_{T,X}$ or $Sepset_{X,T}$ (that is why it is not included in $PC(T)$, but it should be dependent with T as conditioned on $Sepset_{T,X} \cup \{Y\}$ or $Sepset_{X,T} \cup \{Y\}$. The first observation is obvious given the underlying topology, and the second is based on **Theorem 1**.

Lemma 6 In IPC-MB, for each $X \in CanSP(T)$but$X \notin PC(T)$, either $Sepset_{T,X} \neq NIL$ or $Sepset_{X,T} \neq NIL$ ***(Note that*** $\emptyset$ ***means empty set, while*** NIL ***means NULL pointer)****.*

Proof. *If* $X \notin PC(T)$*, obviously it fails some statistical test* $I_D(T, X|\boldsymbol{S})$*in RecognizePC(T) or RecognizePC(X), and* $\boldsymbol{S}$ *must be assigned to* $Sepset_{T,X}$ *or* $Sepset_{X,T}$ *then at Line 8 of ReconigzePC.* $\blacksquare$

Due that either $Sepset_{T,X}$ *or* $Sepset_{X,T}$ may be NULL, it is necessary to check them before the assignment at Line 15 of *IPC-MB*.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Theorem 5 Given $X \in PC(T)$ and $CanPC(X)$ by $RecognizePC(X)$, for each $Y \in CanPC(X) \backslash MB(T) \backslash CanPC(T)$ (excluding recognized and descendants of T if there are), if Y is conditionally dependent with T given $Sepset_{T,X} \cup \{X\}$ or $Sepset_{X,T} \cup \{X\}$ (depending on which one is not NIL), Y is known as a true spouse.

Proof. *Given $X \in PC(T)$ and $Y \in CanPC(X)$, Y can be X's parent, child or descendant of T(**Theorem 2**). Given $Y \in CanPC(X) \backslash MB(T) \backslash CanPC(T)$, it is secure to declare that T is connected with X, denoted as $T - X$, and T is NOT connected with Y,denoted as $Y \nleftrightarrow T$. Besides, due that $Y \notin PC(T)$, there exists $Sepset$ so that $I(T, Y|Sepset)$ (**Lemma 6**). Then, we need prove that if $I(T, Y|Sepset \cup \{X\})$ is NOT true, then Y can only be T's spouse:*

1. *$X \in Pa(T)$ and $Y \in Pa(X)$, i.e. $Y \in ND(T)$, then $Y \to X \to T$ but $Y \nleftrightarrow T$. To blocking the path $Y \to X \to T$, the statement that $X \in Sepset$ must be true. Otherwise, at least we have one non-blocked path, which is contradictory to the fact that $I(T, Y|Sepset)$. Hence, we still have $I(T, Y|Sepset \cup \{X\})$;*
2. *$X \in Pa(T)$ and $Y \in Ch(X)$, i.e.$Y \leftarrow X \to T$ but $Y \nleftrightarrow T$. Same proof as case 1;*
3. *$X \in Pa(T)$ and $Y \in Des(X)$. (1) Since $Y \in CanPC(X)$ and $Y \in Des(X)$, there must exist, at least one, non-blocked path $Y - \cdots - X$. (2) Because $Y \notin PC(T)$, all paths connecting Y and T must be blocked by some $Sepset$. Given one such non-blocked path $Y - \cdots - X$ (due to $Y \in CanPC(X)$), it is extendable to access T via X, i.e. $Y - \cdots - X \to T$. To ensure the d-separation (due to $Y \notin PC(T)$), this path $Y - \cdots - X \to T$ has to be blocked, hence X has to be observed, i.e. $X \in Sepset$. Otherwise, $Y - \cdots - X \to T$ will keep open (since there is no chance to construct a converging pattern here with the existing of $X \to T$), which is contradictory to the fact that $I(T, Y|Sepset)$. Since $X \in Sepset$, we still have $I(T, Y|Sepset \cup \{X\})$;*
4. *$X \in Ch(T)$ and $Y \in Pa(X)$, i.e.$Y \to X \leftarrow T$ but* $Y \nleftrightarrow T$. *It is easy to prove that adding X does make the path $Y \to X \leftarrow T$ non-blocked, i.e. $Sepset \cup \{X\}$ won't d-separates Y and T anymore, and Y is known as a true spouse successfully;*
5. *$X \in Ch(T)$ and $Y \in Ch(X)$, i.e. $Y \leftarrow X \leftarrow T$ but* $Y \nleftrightarrow T$. *Same as case 1;*
6. *$X \in Ch(T)$ and $Y \in Des(X)$. Similar proof as case 3.*

These six cases cover all possible happenings, so the proof itself is complete. It is noticed that only true spouse of T will fail the statistical test at Line 15 (IPC-MB), and be added to $MB(T)$. ∎

Theorem 6 Under the assumptions that the independence tests are correct and that the learning data D is an independent and identically distributed sample from a probability distribution P faithful to a DAG G, all spouses of T are found with IPC-MB.

Proof. *By **Lemma 5**, it is known that $CanSP(T)$ contains all spouses of T. With $\forall Y \in CanPC(X) \backslash MB(T) \backslash CanPC(T)$, it will be correctly recognized if it is true spouse (**Theorem 5**). Since this checking applies to all variables in $CanSP(T)$, we are able to find all spouses of T.* ∎

Therefore, IPC-MB is able to find the true parents and children of *T* first, and further enables us to find the true spouses of *T* based on previous outcome.

IV. Empirical Study

In this section, we compare IPC-MB with two competitive and most cited works, IAMB and PCMB. Besides, we also include PC algorithm into study considering that it is the most known Bayesian network structure learning algorithm. Data sampled from four networks are used for experiments. Table 1 gives a brief introduction of the four networks. Asia and Alarm are known Bayesian network; PolyAlarm is one polytree derived from Alarm; Test152 is one example network, along with PolyAlarm, as included in the BNJ (http://bnj.sourceforge.net, one well-known open source Bayesian network processing software) installation package. Therefore, our experiments cover four algorithms, including local and global search; besides, we study their behavior given tiny, medium, large and polytree Bayesian networks.

Table 1. Summary of the four Bayesian networks used in our experiments.

Name	# of Nodes	# of Arcs	Size of Largest MB
Asia	8	8	5
Alarm	37	46	8
PolyAlarm	37	36	8
Test152	152	200	5

We are interested to measure the accuracy and time efficiency of the algorithms. Regarding the accuracy, we run IAMB, PCMB and IPC-MB with each node of BN as the target variable T, and then report the average precision and recall. **Precision** is the number of true positives in the output divided by the number of nodes in the output. **Recall** is the

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

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

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

to measure the Euclidean **distance** from perfect precision and recall. The significance level for the independence test is 0.05. PC algorithm is ran to induce the whole network, and average precision and distance are measured similarly.

The time efficiency is measured by the number of data passes and CI tests as required. It is easy to understand why we measure the times of CI tests, considering that all these algorithms are built on statistical tests. Regarding the data pass, it is defined as scanning the data file for one time. Because it involves disk operation, and the number of observations may be very large, scanning the whole data may be quite time consuming. This measure normally is ignored in theoretical work, but it may be influential to the actual timing cost in practice where it is possible to cache all instances or related frequency counts in memory. In our implementation, we construct necessary contingency tables only when know they are required, and one data pass is consumed for the collection of related frequencies. For example, in IPC-MB, we need a scan of the data for different cutSetSize in *RecognizePC* since the size of conditioning set, as well as the adjacency set, change. To make the comparison fair, we try our best to cache all expected frequencies within a data pass.

Similarly we report the average number of data passes and CI tests as required by IAMB/PCMB/IPC-MB to induce the corresponding Markov blanket given different node as target, and the number of CI tests as reported about PC is that required by it to induce the whole Bayesian network. This will give us chance to compare the relative efficiency between local and global search. Given each BN, experiments with different sample size are conducted, and 10 groups of samples are prepared given each different sample size.

V. Conclusion

In this paper, we briefly review related algorithms for inducing $MB(T)$ since it is believed as the optimal feature subset for the prediction of T. Then, we introduce how IPC-MB works, alone with proof of its correctness. Experiments are conducted to compare IPC-MB with IAMB, PCMB and PC. From Table 2 and Table 3, we observe that (1) IPC-MB has the best accuracy performance among the four algorithms, given the same amount of instances; (2) IAMB is quite poor on accuracy performance, though it is expected fastest among the four algorithms; (3) PCMB is much slower than IPC-MB and IAMB, and may even be slower than PC, though it declares as local search; (4) By local search, IPC-MB reduces the time complexity greatly as compared with PC which induces the Markov blanket by learning the whole Bayesian network first.

References

[1] M. S. Daphne Koller, "Toward Optimal Feature Selection," in ICML, 1996, pp. 284-292.

[2] J. Pearl, Probabilistic reasoning in expert systems. San Matego: Morgan Kaufmann, 1988.

[3] S. T. Dimitris Margaritis, "Bayesian Network Induction via Local Neighborhoods," in Advances in Neural Information Processing Systems Denver, Colorado, USA, 1999.

[4] C. F. A. Ioannis Tsamardinos, Alexander R. Statnikov, "Time and sample efficient discovery of Markov blankets and direct causal relations," in the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 673-678.

[5] C. F. A. Ioannis Tsamardinos, Alexander R. Statnikov, "Algorithms for Large Scale Markov Blanket Discovery," in the Sixteenth International Florida Artificial Intelligence Research Society Conference, St. Augustine, Florida, USA, 2003, pp. 376-381.

[6] I. T. Constantin F. Aliferis, Alexander R. Statnikov, "HITON: A novel Markov blanket algorithm for optimal variable selection," in American Medical Informatics Association Annual Symposium, 2003, pp. 21-25.

[7] D. M. Sandeep Yaramakala, "Speculative Markov Blanket Discovery for Optimal Feature Selection," in ICDM, 2005, pp. 809-812.

[8] R. N. José M. Peña, Johan Björkegren, Jesper Kalé, "Towards scalable and data efficient learning of Markov boundaries," International Journal of Approximate Reasoning vol. 45, 2007.

[9] M. C. D. Shunkai Fu, "Fast Markov Blanket Discovery Algorithm Via Local Learning within Single Pass," in Canadian Conference on AI, Windsor, Canada, 2008, pp. 96-107.

[10] M. C. D. Shunkai Fu, "Tradeoff Analysis of Different Markov Blanket Local Learning Approaches," in Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference (PAKDD), Osaka, Japan, 2008, pp. 562-571.

[11] [11] C. G. Peter Spirtes, Richard Scheines, Causation, Prediction and Search (2nd Edition): The MIT Press, 2001.

TABLE 2. AVERAGE ACCURACY (INCLUDING PRECISION, RECALL AND DISTANCE) ABOUT IAMB/PCMB/IPC-MB/PC GIVEN SMALL TO LARGE PROBLEMS

BN	Instances	Algorithm	Precision	Recall	Distance
Asia	100	IAMB	.55±.08	.51±.09	.72±.10
		PCMB	.55±.11	.49±.17	.76±.15
		IPC-MB	.55±.11	.47±.17	.77±.16
		PC	.55±.14	.60±.26	.71±.13
	4000	IAMB	.85±.07	.82±.09	.26±.11
		PCMB	.86±.04	.76±.11	.31±.10
		IPC-MB	.87±.02	.76±.07	.30±.08
		PC	.83±.05	.74±.08	.35±.08
Alarm	500	IAMB	.57±.02	.55±.02	.67±.04
		PCMB	.86±.03	.78±.04	.31±.05
		IPC-MB	.85±.02	.77±.04	.32±.04
		PC	.77±.05	.78±.03	.37±.04
	4000	IAMB	.51±.03	.59±.02	.68±.03
		PCMB	.97±.02	.94±.03	.07±.04
		IPC-MB	.99±.01	.95±.01	.06±.03
		PC	.97±.01	.94±.02	.09±.03
PolyAlarm	500	IAMB	.64±.03	.71±.03	.53±.04
		PCMB	.84±.05	.75±.04	.33±.07
		IPC-MB	.85±.05	.74±.04	.33±.07
		PC	.76±.07	.72±.05	.43±.08
	2000	IAMB	.65±.02	.89±.01	.42±.02
		PCMB	.93±.02	.89±.02	.14±.02
		IPC-MB	.93±.01	.90±.03	.13±.04
		PC	.83±.03	.83±.02	.29±.04
Test152	250	IAMB	.54±.03	.74±.00	.59±.01
		PCMB	.89±.02	.71±.01	.37±.02
		IPC-MB	.90±.02	.71±.01	.36±.01
		PC	.72±.03	.71±.01	.49±.02
	2000	IAMB	.44±.01	.93±.01	.58±.01
		PCMB	.93±.01	.96±.02	.11±.02
		IPC-MB	.95±.01	.96±.02	.09±.02
		PC	.78±.02	.96±.01	.25±.02

TABLE 3. AVERAGE TIME EFFICIENCY MEASURED BY THE NUMBER OF CI TESTS AS REQUIRED BY IAMB/PCMB/IPC-MB/PC GIVEN SMALL TO LARGE PROBLEMS

BN	Instances	Algorithm	#Data Passes	#CI Tests
Asia	100	IAMB	5±1	25±3
		PCMB	80±87	2006±3673
		IPC-MB	10±7	188±288
		PC	26±9	213±267
	4000	IAMB	5±0	23±1
		PCMB	50±7	436±84
		IPC-MB	8±1	84±9
		PC	26±4	139±10
Alarm	500	IAMB	5±0	116±2
		PCMB	160±11	4638±374
		IPC-MB	12±1	561±31
		PC	220±16	2736±82
	4000	IAMB	7±0	187±4
		PCMB	218±6	16007±1326
		IPC-MB	14±0	849±48
		PC	211±18	3902±122
PolyAlarm	500	IAMB	4±0	106±3
		PCMB	47±3	584±48
		IPC-MB	7±0	143±8
		PC	117±16	1061±48
	2000	IAMB	5±0	147±2
		PCMB	59±2	837±57
		IPC-MB	9±0	179±6
		PC	158±24	1223±35
Test152	250	IAMB	5±0	750±1
		PCMB	89±4	3757±148
		IPC-MB	11±0	924±28
		PC	608±3	19803±392
	2000	IAMB	8±0	1041±2
		PCMB	148±3	5928±174
		IPC-MB	15±0	1432±46
		PC	684±80	26173±593

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)