A novel scalable and correct Markov boundary learning algorithm under faithfulness condition

Sergio Rodrigues de Morais
INSA-Lyon, LIESP, F-69622 Villeurbanne, France

Alex Aussem
University of Lyon 1, LIESP, F-69622 Villeurbanne, France

Abstract

In this paper, we propose a novel constraint-based Markov boundary discovery algorithm, called MBOR, that scales up to hundreds of thousands of variables. Its correctness under faithfulness condition is guaranteed. A thorough empiric evaluation of MBOR's robustness, efficiency and scalability is provided on synthetic databases involving thousands of variables. Our experimental results show a clear benefit in several situations: large Markov boundaries, weak associations and approximate functional dependencies among the variables.

1 Introduction

In this paper, we aim to identify the minimal subset of discrete random variables that is relevant for probabilistic classification in data sets with many variables but few instances (Guyon and Elisseeff, 2003). A principled solution to this problem is to determine the *Markov boundary* of the class variable T, i.e., the minimal subset of $\mathbf{U}$ (the full set), denoted by $\mathbf{MB}_T$ in the sequel, that renders the rest of $\mathbf{U}$ independent of T (Nilsson et al., 2007).

Following (Peña et al., 2007), we present a novel divide-and-conquer method, called MBOR, in order to increase the efficiency of the Markov boundary (MB for short) discovery while still being scalable and correct under the faithfulness condition. The problem with constraint-based algorithms is that the conditional independence tests become unreliable as the size of the conditional set increases. Such errors have usually a cascading effect that causes many errors in the final graph.

To get around this problem, MBOR combines rough and moderately accurate MB learners based on IAMB (Tsamardinos et al., 2003) and keeps the conditional test sizes of the tests as small as possible. A key difference between MBOR and all correct divide-and-conquer algorithms is the OR condition: two variables X and Y are considered as neighbors by MBOR if $Y \in PC_X$ OR $X \in PC_Y$, instead of the more stringent AND condition. Clearly, the OR condition makes it easier for true positive nodes to enter the Markov boundary, hence the name and the practical efficiency of our algorithm. Interestingly, some almost-deterministic relationships are also handled by the OR condition. The only difficulty was to maintain the correctness of the algorithm under the faithfulness condition.

In (Rodrigues de Morais and Aussem, 2008), we compared the ability of MBOR to solve real FSS problems using real data bases from the UCI Machine Learning Repository (e.g., Car Evaluation, Chess, Molecular Biology, SPECT heart, Tic-Tac-Toe, Wine and Waveform). In this study, we assess the scalability and the performance of MBOR through several experiments on synthetic databases with very few instances compared to the number of variables. MBOR is proved by extensive empirical simulations to be an excellent trade-off between running time and quality of reconstruction.

2 Notations and preliminaries

We denote a variable with an upper-case, X, and value of that variable by the same lower-case, x. We denote a set of variables by upper-case bold-face, $\mathbf{Z}$, and we use the corresponding lower-case bold-face, $\mathbf{z}$, to denote an assignment of value to each variable in the set. In this paper, we only deal with discrete random variables. We denote the conditional independence of the variable X and Y given $\mathbf{Z}$, in some distribution P by $X \perp_P Y|\mathbf{Z}$. Similarly, we write $X \perp_{\mathcal{G}} Y|\mathbf{Z}$ if X and Y are d-separated by $\mathbf{Z}$ in the DAG $\mathcal{G}$.

A Markov blanket $\mathbf{M}_T$ of the T is is any set of variables such that T is conditionally independent of all the remaining variables given $\mathbf{M}_T$. A Markov boundary, $\mathbf{MB}_T$, of T is any Markov blanket such that none of its proper subsets is a Markov blanket of T. In general, in a Baysesian network $< \mathcal{G}, P >$, we would want an edge to mean a direct dependency. As we know, the faithfulness entails this:

Definition 1. Suppose we have a joint probability distribution P of the random variables in some set $\mathbf{U}$ and a DAG $\mathcal{G} =< \mathbf{U}, \mathbf{E} >$. We say that $< \mathcal{G}, P >$ satisfies the faithfulness condition if, based on the Markov condition, G entails all and only conditional independencies in P.

Theorem 1. *Suppose $< \mathcal{G}, P >$ satisfies the faithfulness condition. Then for each variable X, the set of parents, children of X, and parents of children of X is the unique Markov boundary.*

A proof can be found for instance in (Neapolitan, 2004). A *spouse* of T is a another parent of a T's child node. We denote by $\mathbf{PC}_T$, the unique set of parents and children of T in $\mathcal{G}$ when $< \mathcal{G}, P >$, satisfies the faithfulness condition. Otherwise, $\mathbf{PC}_X^{\mathbf{U}}$ will denote the unique set of the variables that remains dependent on X conditioned on any set $\mathbf{Z} \in \mathbf{U} \setminus \{X, Y\}$.

3 Some problems with constraint-based methods

Constraint-based (CB for short) procedures systematically check the data for independence relationships to infer the structure. The association between two variables X and Y given a conditioning set $\mathbf{Z}$ is a measure of the strength of the dependence with respect to the data base $\mathcal{D}$. It is usually implemented with a statistical measure of association (e.g. χ^2, G^2). CB methods have the advantage of possessing clear stopping criteria and deterministic search procedures. On the other hand, they are prone to several instabilities: namely if a mistake is made early on in the search, it can lead to incorrect edges which may in turn lead to bad decisions in the future, which can lead to even more incorrect edges. This instability has the potential to cascade, creating many errors in the final graph (Dash and Druzdzel, 2003).

Insufficient data presents a lot of problems when working with statistical inference techniques like the independence test mentioned earlier. This occurs typically when the expected counts in the contingency table are small. The decision of accepting or rejecting the null hypothesis depends implicitly upon the degree of freedom which increases exponentially with the number of variables in the conditional set. So the larger the size of the conditioning test, the less accurate are the estimates of conditional probabilities and hence the less reliable are the independence tests. Another difficulty arises when true- or almost-deterministic relationships (ADR) are observed among the variables. Loosely speaking, a relationship is said to be almost deterministic when the fraction of tuples that violate the deterministic dependency is at most equal to some threshold. True DR are source of unfaithfulness but the existence of ADR among variables doesn't invalidate the faithfulness assumption. Several proposals have been discussed in the literature in order to reduce the cascading effect of early errors that causes many errors to be present in the final graph. The general idea is to keep the size of the conditional sets as small as possible in the curse of the learning process. Another idea is to reduce the degree of freedom of the statistical conditional independence test by some ways. The aim is twofold: to improve the data efficiency and to allow an early detection of ADR. Theses strategies are not discussed here for conciseness, see (Yilmaz et al., 2002; Luo, 2006; Aussem et al., 2007; Rodrigues de Morais et al., 2008) for instance.

4 New method

In this section, we present in detail our learning algorithm called MBOR. We recall that MBOR was designed in order to endow the search procedure with the ability to: 1) handle efficiently data sets with thousands of variables but very few instances, 2) be correct under faithfulness condition, 3) handle implicitly some approximate deterministic relationships (ADR) without detecting them. We discuss next how we tackle each problem.

First of all, MBOR scales up to hundreds of thousands of variables in reasonable time because it searches the Markov boundary of the target without having to construct the whole Bayesian network first. Like PCMB (Peña et al., 2007) and MMMB (Tsamardinos et al., 2006), MBOR takes a divide-and-conquer approach that breaks the problem of identifying $\mathbf{MB}_T$ into two subproblems : first, identifying $\mathbf{PC}_T$ and, second, identifying the parents of the children (the spouses $\mathbf{SP}_T$) of T. According to Peña et al., this divide-and-conquer approach is supposed to be more data efficient than IAMB (Tsamardinos et al., 2003) and its variants, e.g., Fast-IAMB (Yaramakala, 2004) and Interleaved-IAMB (Yaramakala and Margaritis, 2005), because $\mathbf{MB}_T$ can be identified by conditioning on sets much smaller than those used by IAMB. Indeed, IAMB and its variants seek directly the minimal subset of $\mathbf{U}$ (the full set) that renders the rest of $\mathbf{U}$ independent of T, given $\mathbf{MB}_T$. Moreover, MBOR keeps the size of the conditional sets to the minimum possible without sacrificing the performance as discussed next.

The advantage of the divide-and-conquer strategy in terms of data efficiency does not come without some cost. MMMB (Tsamardinos et al., 2006) and PCMB (Peña et al., 2007) apply the "AND condition" to prove correctness under faithfulness condition. In other words, two variables X and Y are considered as neighbors if $Y \in PC_X$ AND $X \in PC_Y$. We believe this condition is far too severe and yields too many false negatives in the output. Instead, MBOR stands for "Markov Boundary search using the OR condition". This "OR condition" is a major difference between MBOR and all the above mentioned correct divide-and-conquer algorithms: two variables X and Y are considered as neighbors with MBOR if $Y \in PC_X$ OR $X \in PC_Y$. Clearly, the OR condition makes it easier for true positive nodes to enter the Markov boundary, hence the name and the practical efficiency of our algorithm. Moreover, the OR condition is a simple way to handle some ADR. For illustration, consider the sub-graph $X \Rightarrow T \rightarrow Y$, since $X \Rightarrow T$ is an ADR, $T \perp Y | X$ so Y will not be considered as a neighbor of T. As Y still sees T in its neighborhood, Y and T will be considered as adjacent by application of the OR condition. The main difficulty was to demonstrate the correctness under the faithfulness condition despite the OR condition.

MBOR (Algorithm 1) works in three steps and it is based on four subroutines called *PCSuperset*, *SPSuperset* and *MBtoPC* (Algorithms 2-4). Before we describe the algorithm step by step, we recall that the general idea underlying MBOR is to use a weak MB learner to create a stronger MB learner. By weak learner, we mean a simple and fast method that may produce many mistakes due to its data inefficiency. In other words, the proposed method aims at producing an accurate MB discovery algorithm by combining fast and moderately inaccurate (but correct) MB learners. The weak MB learner is used in *MBtoPC* (Algorithm 4) to implement a correct Parents and Children learning procedure. It works in two steps. First, the weak MB learner called *CorrectMB* is used at line 1 to output a candidate MB. *CorrectMB* may be implemented by any algorithm of the IAMB family because they don't implement the AND condition. In our implementation, we use Inter-IAMB for its simplicity and performance (Tsamardinos et al., 2003). The key difference between IAMB and Inter-IAMB is that the shrinking phase is interleaved into the growing phase in Inter-IAMB. The second step (lines 3-6) of *MBtoPC* removes the spouses of the target.

In phase I, MBOR calls *PCSuperset* to extract **PCS**, a superset for the parents and children, and then calls *SPSuperset* to extract **SPS**, a superset for the target spouses (parents of children). Filtering reduces as much as possible the number of variables before proceeding to the MB discovery. In *PCSuperset* and *SPSuperset*, the size of the conditioning set $\mathbf{Z}$ in the tests is severely restricted: $card(\mathbf{Z}) \leq 1$ in *PCSuperset* (lines 3 and 10) and $card(\mathbf{Z}) \leq 2$ in *SPSuperset* (lines 5 and 11). As discussed before, conditioning on larger sets of variables would increase the risk of missing variables that are weakly associated to the target. It would also lessen the reliability of the independence tests. So the MB superset, **MBS** (line 3), is computed based on a scalable and highly data-efficient procedure. Moreover, the filtering phase is also a way to handle some ADR. For illustration, consider the sub-graph $Z \Rightarrow Y \rightarrow T \Leftarrow X$, since $X \Rightarrow T$ and $Z \Rightarrow Y$ are ADRs, $T \perp Y|X$ and $Y \perp T|Z$, Y would not be considered as a neighbor of T and vice-versa. The OR-condition in Phase II would not help in this particular case. Fortunately, as Phase I filters out variable Z, Y and T will be considered as adjacent

Phase II finds the parents and children in the restricted set of variables using the OR condition. Therefore, all variables that have T in their vicinity are included in $\mathbf{PC}_T$ (lines 7-8).

Phase III identifies the target's spouses in **MBS** in exactly the same way PCMB does (Peña et al., 2007). Note however that the OR condition is not applied in this last phase because it would not be possible to prove its correctness anymore.

The theorem below establishes MBOR's correctness under faithfulness condition:

Theorem 1. Under the assumptions that the independence tests are reliable and that the database is an independent and identically distributed sample from a probability distribution P faithful to a DAG $\mathcal{G}$, $MBOR(T)$ returns $\mathbf{MB}_T^{\mathbf{U}}$.

The proof may be found in (Rodrigues de Morais and Aussem, 2008). It is omitted here for conciseness. Note that the demonstration is not completely straightforward because a difficulty arises: as **MBS** is a subset of $\mathbf{U}$, a marginal distribution $P^{\mathbf{V}}$ of $\mathbf{V} \subset \mathbf{U}$ may not satisfy the faithfulness condition with any DAG even if $P^{\mathbf{U}}$ does. This is an example of embedded faithfulness (Neapolitan, 2004) and every distribution doesn't admit an embedded faithful representation.

Algorithm 1 *MBOR*

```
Require: $T$: target; $D$: data set ($\mathbf{U}$ is the set of variables)
Ensure: [$\mathbf{PC}$,$\mathbf{SP}$]: Markov boundary of $T$

    Phase I: Find MB superset ($\mathbf{MBS}$)
 1: $[\mathbf{PCS}, \mathbf{dSep}] = PCSuperSet(T, D)$
 2: $\mathbf{SPS} = SPSuperSet(T, D, \mathbf{PCS}, \mathbf{dSep})$
 3: $\mathbf{MBS} = \mathbf{PCS} \cup \mathbf{SPS}$
 4: $\mathcal{D} = \mathcal{D}(\mathbf{MBS} \cup T)$ i.e., remove from data set all variables in $\mathbf{U}/\{\mathbf{MBS} \cup T\}$

    Phase II: Find parents and children of the target
 5: $\mathbf{PC} = MBtoPC(T, \mathcal{D})$
 6: for all $X \in \mathbf{PCS} \setminus \mathbf{PC}$ do
 7:     if $T \in MBtoPC(X, \mathcal{D})$ then
 8:         $\mathbf{PC} = \mathbf{PC} \cup X$
 9:     end if
10: end for

    Phase III: Find spouses of the target
11: $\mathbf{SP} = \emptyset$
12: for all $X \in \mathbf{PC}$ do
13:     for all $Y \in MBtoPC(X, D) \setminus \{\mathbf{PC} \cup T\}$ do
14:         Find minimal $\mathbf{Z} \subset \mathbf{MBS} \backslash \{T \cup Y\}$ such that $T \perp Y|\mathbf{Z}$
15:         if $(T \not\perp Y|\mathbf{Z} \cup X)$ then
16:             $\mathbf{SP} = \mathbf{SP} \cup Y$
17:         end if
18:     end for
19: end for
```

5 Experimental validation

In this section, we assess the scalability and the accuracy of MBOR through several experiments on synthetic databases with very few instances compared to the number of variables. We evaluate first the accuracy, the data-efficiency and running time of MBOR as the number of variables increases. Then, we compare the accuracy of MBOR against InterIAMB and PCMB

Algorithm 2 *PCSuperSet*

```
Require: $T$: target; $D$: data set ($\mathbf{U}$ is the set of variables)
Ensure: PCS: PC superset of $T$; dSep: d-separation set;

    Phase I: Remove $X$ if $T \perp X$
 1: $\mathbf{PCS} = \mathbf{U} \setminus T$
 2: for all $X \in \mathbf{PCS}$ do
 3:    if $(T \perp X)$ then
 4:       $\mathbf{PCS} = \mathbf{PCS} \setminus X$
 5:       $\mathbf{dSep}(X) = \emptyset$
 6:    end if
 7: end for

    Phase II: Remove $X$ if $T \perp X|Y$
 8: for all $X \in \mathbf{PCS}$ do
 9:    for all $Y \in \mathbf{PCS} \setminus X$ do
10:       if $(T \perp X \mid Y)$ then
11:          $\mathbf{PCS} = \mathbf{PCS} \setminus X$
12:          $\mathbf{dSep}(X) = Y$; go to 15
13:       end if
14:    end for
15: end for
```

Algorithm 3 *SPSuperSet*

```
Require: $T$: target; $D$: data set ($\mathbf{U}$ is the set of variables); PCS: PC superset of $T$; dSep: d-separation set;
Ensure: SPS: SP superset of $T$;

 1: $\mathbf{SPS} = \emptyset$
 2: for all $X \in \mathbf{PCS}$ do
 3:    $\mathbf{SPS}_X = \emptyset$
 4:    for all $Y \in \mathbf{U} \setminus \{T \cup \mathbf{PCS}\}$ do
 5:       if $(T \not\perp Y | \mathbf{dSep}(Y) \cup X)$ then
 6:          $\mathbf{SPS}_X = \mathbf{SPS}_X \cup Y$
 7:       end if
 8:    end for
 9:    for all $Y \in \mathbf{SPS}_X$ do
10:       for all $Z \in \mathbf{SPS}_X \setminus Y$ do
11:          if $(T \perp Y | X \cup Z)$ then
12:             $\mathbf{SPS}_X = \mathbf{SPS}_X \setminus Y$; go to 15
13:          end if
14:       end for
15:    end for
16:    $\mathbf{SPS} = \mathbf{SPS} \cup \mathbf{SPS}_X$
17: end for
```

Algorithm 4 *MBtoPC*

```
Require: $T$: target; $D$: data set
Ensure: PC: Parents and children of $T$;

 1: $\mathbf{MB} = CorrectMB(T, D)$
 2: $\mathbf{PC} = \mathbf{MB}$
 3: for all $X \in \mathbf{MB}$ do
 4:    if $\exists \mathbf{Z} \subset (\mathbf{MB} \setminus X)$ such that $T \perp X \mid \mathbf{Z}$ then
 5:       $\mathbf{PC} = \mathbf{PC} \setminus X$
 6:    end if
 7: end for
```

on six well-know BN benchmarks. To evaluate the accuracy, we combine precision (i.e., the number of true positives divided in the output by the number of nodes in the output) and recall (i.e., the number of true positives divided by the true size of the Markov Boundary) as $\sqrt{(1-precision)^2 + (1-recall)^2}$, to measure the Euclidean distance from perfect precision and recall, as proposed in (Peña et al., 2005). To implement the conditional independence test, we calculate the G^2 statistic as in (Spirtes et al., 2000), under the null hypothesis of the conditional independence. The significance level of the test is fixed to 0.05 for all algorithms. It might very well happen that several variables have the same association value with the target in data sets with very few instances. In this particular case, somewhat arbitrary (in)dependence decisions are taken. This can be seen as a source of randomness inherent to all CB procedures. To handle this problem, our implementation breaks ties at random: a random permutation of the variables is carried out before each algorithm is run.

5.1 Scalability

We compare first the accuracy of PCMB and MBOR through experiments on the INSURANCE (27 nodes/52 arcs) benchmark replicated several times (up to 1000 times) to increase the number of variables. We run MBOR and PCMB with the variable 'RiskAversion' as the target. The latter has 10 variables in its MB. Each network is obtained by tiling several copies of the initial INSURANCE network. The tiling is performed in a way that maintains the structural and probabilistic properties of the original network in the tiled network. We focus here on the accuracy and efficiency of the algorithms as a function of the number of variable in the tiled network (up to 27,000 variables). Clearly, the additional variables are all independent on the target. We report the number of conditional independence tests that were conducted (in log-log scale), the distribution of the conditioning test sizes and the Euclidean distance from perfect precision and recall, as a function of the number of variables in the tiled network. The average and standard deviation values are estimated over 50 databases. As may be seen, MBOR requires fewer independence conditional tests than PCMB. The number of tests directly influences the execution time. It grows linearly with the number of nodes for both algorithms. The number of conditional tests of MBOR is 48% that of PCMB. As can be seen from Fig.1 (middle), while PCMB conducts (proportionally) fewer conditional tests which indicates improved test reliability, MBOR yields a significantly shorter distance in all cases Fig.1 (bottom).

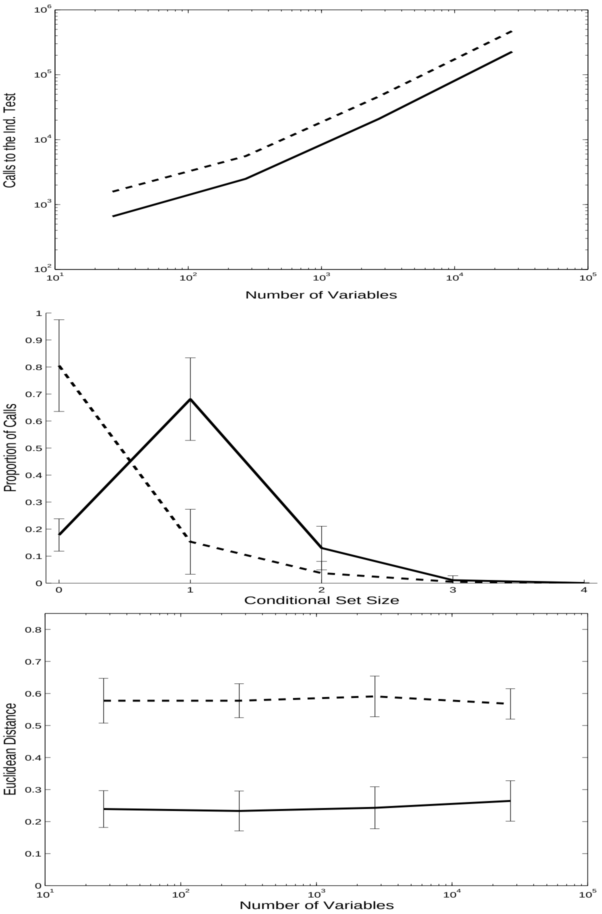

Figure 1: Insurance BN tiled several times. From top to bottom: number of conditional independence tests, distribution of the conditioning test sizes, Euclidean distance, as a function of the number of variables in the tiled BN. MBOR in plain line, PCMB in dotted line.

5.2 Accuracy

We report now the results of our experiments on six common benchmarks : BREAST-CANCER or ASIA (8 nodes/8 arcs), INSURANCE (27/52), INSULINE (35/52), ALARM (37/46), HAILFINDER (56/66) and CARPO (61/74). For each benchmark, we sampled 100 databases containing 100, 500 and 1000 instances respectively. The three algorithms were run, first, with each node in the BN as the target, and second, with the node with the largest MB in the BN as target. Figure 2 summarizes the empirical distribution of the Euclidean distance over 100 databases in the form of triplets of boxplots, one for each algorithm (PCMB, Inter-IAMB and MBOR respectively). Boxplots are convenient ways of graphically depicting the distributions of the Euclidean distances through their five-number summaries (the smallest observation, lower quartile, median, upper quartile, and largest observation). The boxplots also indicate (by the symbol '+') which observations, if any, might be considered outliers. In the left column of Fig.2, the distance is averaged over all nodes in the BN. In the right column, the distance for the node with the largest MB in the BN (i.e., ASIA : 'OR' (MB = 5 variables) ALARM : 'Intubation' & 'HR' (8 variables) INSULINE : 'IPA' & 'GPA' (18 variables) INSURANCE : 'RiskAversion' & 'Accident' (10 variables) HAILFINDER : 'CldShadeOth' (8 variables), CARPO : 'N69' (18 variables).

Several observations can be made from the results in Fig.2. First, it is rather surprising to observe that PCMB performs often worse than interIAMB even if PCMB is meant to conduct more reliable tests by conditioning on fewer variables. Despite the more reliable tests, the AND condition used in PCMB makes it hard for true positives to enter candidate MB. Second, the overall performance of MBOR and InterIAMB, when averaged over all nodes, is very similar (left column). For larger MBs, however, the advantages of MBOR against the other two algorithms are far more noticeable (right column). For instance, MBOR consistently outperforms the other algorithms, especially for databases with 500 and 1000 instances. The larger the MB size, and the greater the gain in performance. As expected, the gain in accuracy is very significant on target variables 'IPA' & 'GPA' in INSULINE and on variable 'N69' in CARPO which contain 18 variables in their MB. The reason is that MBOR reduces drastically the average number of false negatives compared to PCMB and InterIAMB and this benefit comes at very little expense in terms of false positives. Moreover, the gain in accuracy seems to increase with the size of the database.

6 Conclusion

We discussed simple solutions to improve the efficiency of current constraint-based Markov boundary discovery algorithms. We proposed a novel approach called MBOR. Our experimental results on well-known benchmarks show a clear benefit in several situations: densely connected DAGs, weak associations or approximate functional dependencies among the variables.

References

A. Aussem, S. Rodrigues de Morais, and M. Corbex. 2007. Nasopharyngeal carcinoma data analysis with a novel bayesian network skeleton learning. In *11th Conference on Artificial Intelligence in Medicine AIME 07*, pages 326–330.

Denver Dash and Marek J. Druzdzel. 2003. Robust independence testing for constraint-based learning of causal structure. In *UAI*, pages 167–174.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

Wei Luo. 2006. Learning bayesian networks in semideterministic systems. In *Canadian Conference on AI*, pages 230–241.

R. E. Neapolitan. 2004. *Learning Bayesian Networks*. Prentice Hall.

R. Nilsson, J.M. Peña, J. Bjrkegren, and J. Tegnr. 2007. Consistent feature selection for pattern recognition in polynomial time. *Journal of Machine Learning Research*, 8:589–612.

J.M. Peña, J. Bjrkegren, and J. Tegnér. 2005. Scalable, efficient and correct learning of markov boundaries under the faithfulness assumption. In *8th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty (ECSQARU 2005)*, volume 21, pages 136–147. Lecture Notes in Artificial Intelligence 3571.

J.M. Peña, R. Nilsson, J. Bjrkegren, and J. Tegnr. 2007. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232.

S. Rodrigues de Morais and A. Aussem. 2008. A novel scalable and data efficient feature subset selection algorithm. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD*, Antwerp, Belgium.

S. Rodrigues de Morais, A. Aussem, and M. Corbex. 2008. Handling almost-deterministic relationships in constraint-based bayesian network discovery : Application to cancer risk factor identification. In *16th European Symposium on Artificial Neural Networks ESANN'08*, pages 101–106.

Peter Spirtes, Clark Glymour, and Richard Scheines. 2000. *Causation, Prediction, and Search*. The MIT Press, 2 edition.

Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander R. Statnikov. 2003. Algorithms for large scale markov blanket discovery. In *FLAIRS Conference*, pages 376–381.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. 2006. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.

Sandeep Yaramakala and Dimitris Margaritis. 2005. Speculative markov blanket discovery for optimal feature selection. In *ICDM*, pages 809–812.

Sandeep Yaramakala. 2004. Fast markov blanket discovery. In *MS-Thesis, Iowa State University.*

Yusuf Kenan Yilmaz, Ethem Alpaydin, H. Levent Akin, and Taner Bilgiç. 2002. Handling of deterministic relationships in constraint-based causal discovery. In *Probabilistic Graphical Models.*

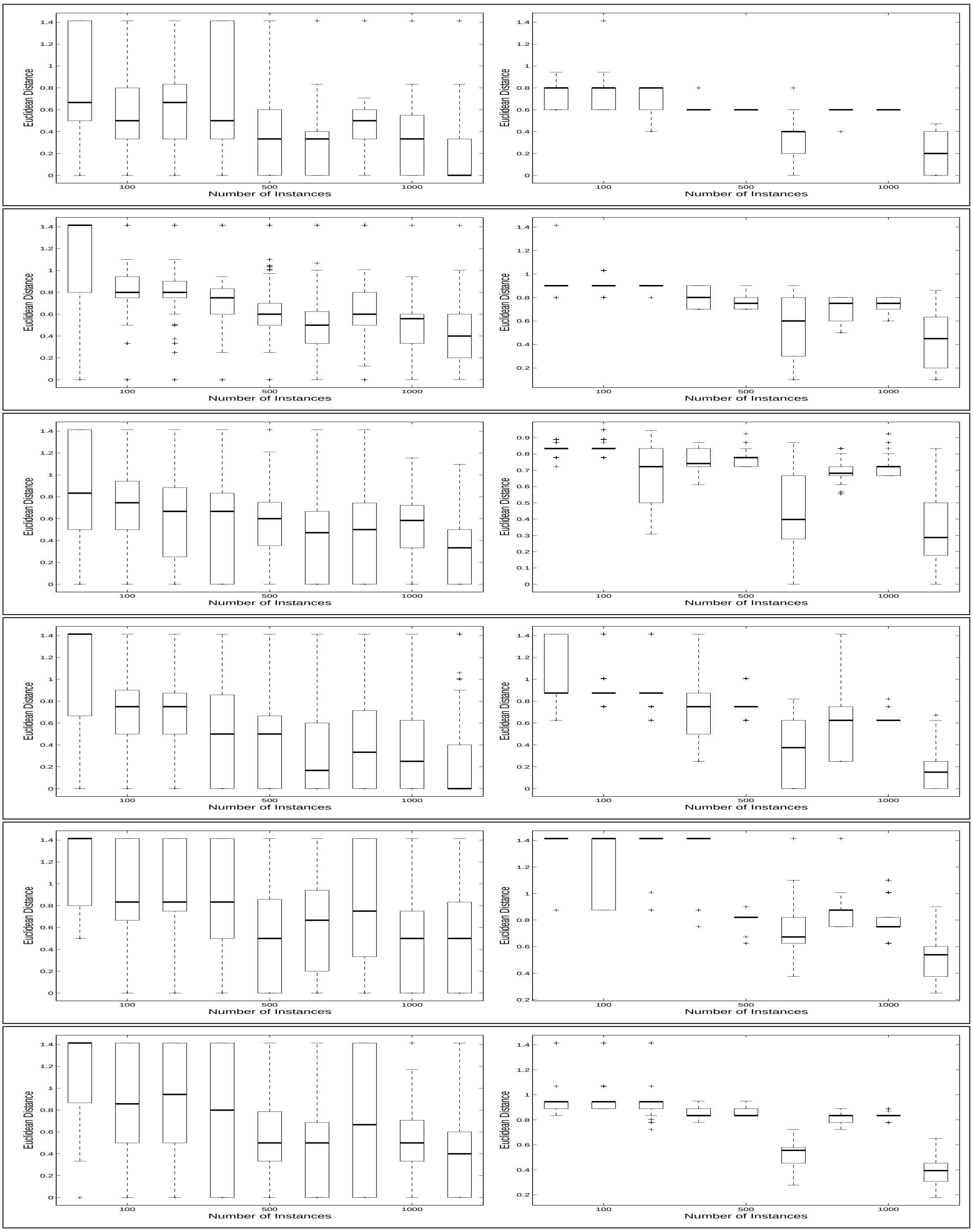

Figure 2: Empirical distribution of the Euclidean distance from perfect precision and recall over 100 databases. Results are shown for 100, 500 and 1000 instances in the form of triplets of boxplots for PCMB (left), InterIAMB (middle) and PCMB (right). From top to bottom: BREAST-CANCER, INSURANCE,, INSULINE, ALARM, HAILFINDER and CARPO. Left column: distance is averaged over all nodes in the BN. Right plot: distance for the node with the largest MB in the BN.