Learning Neighborhoods of High Confidence in Constraint-Based Causal Discovery

Sofia Triantafillou[1,2], Ioannis Tsamardinos[1,2], and Anna Roumpelaki[1,2]

[1] Institute of Computer Science,
Foundation for Research and Technology - Hellas (FORTH),
N. Plastira 100 Vassilika Vouton, GR-700 13 Heraklion, Crete, Greece
[2] Computer Science Department, University of Crete, Heraklion, Greece

Abstract. Constraint-based causal discovery algorithms use conditional independence tests to identify the skeleton and invariant orientations of a causal network. Two major disadvantages of constraint-based methods are that (a) they are sensitive to error propagation and (b) the results of the conditional independence tests are binarized by being compared to a hard threshold; thus, the resulting networks are not easily evaluated in terms of reliability. We present PROPeR, a method for estimating posterior probabilities of pairwise relations (adjacencies and non-adjacencies) of a network skeleton as a function of the corresponding p-values. This novel approach has no significant computational overhead and can scale up to the same number of variables as the constraint-based algorithm of choice. We also present BiND, an algorithm that identifies neighborhoods of high structural confidence on causal networks learnt with constraint-based algorithms. The algorithm uses PROPeR to estimate the confidence of all pairwise relations. Maximal neighborhoods of the skeleton with minimum confidence above a user-defined threshold are then identified using the Bron-Kerbosch algorithm for identifying maximal cliques. In our empirical evaluation, we demonstrate that (a) the posterior probability estimates for pairwise relations are reasonable and comparable with estimates obtained using more expensive Bayesian methods and (b) BiND identifies sub-networks with higher structural precision and recall than the output of the constraint-based algorithm.

Keywords: Posterior probabilities, causal networks, constraint-based causal discovery.

1 Introduction

Constraint-based algorithms are a popular choice for learning causal models; they are fast, scalable, and usually guarantee soundness and completeness in the sample limit. However, for smaller sample sizes, identification of false constraints poses a challenge: An erroneous identification of a conditional independence can propagate through the network and lead to erroneous edge identifications or conflicting orientations even in seemingly unrelated parts of the network. Particularly for networks with many variables and small sample sizes, error propagation can result in unreliable networks.

L.C. van der Gaag and A.J. Feelders (Eds.): PGM 2014, LNAI 8754, pp. 487–502, 2014.

Constraint-based algorithms query the data for conditional independencies and then use the results to constrain the search space of possible causal models. Failure to identify which parts of the output of a constraint-based algorithm are reliable is partly due to the nature of conditional independence tests: The test returns a p-value, which stands for the probability of getting a test statistic at least as extreme as the the one actually observed in the data, given that the null hypothesis (conditional independence) is true. If this probability is lower than a chosen significance threshold (typically 5-10%), the null hypothesis is rejected, and the alternative hypothesis is implicitly accepted. While lower p-values indicate higher confidence conditional dependencies, the p-value can not be interpreted as the probability of a conditional independence, and it therefore cannot be used to compare a conditional dependence to a conditional independence in terms of belief. Thus, the decisions made by the constraint-based algorithm (accept or reject a conditional independence) cannot be evaluated in terms of confidence.

We propose Posterior RatiO PRobability (PROPeR), a method for identifying posterior probabilities for all (non) adjacencies of a causal network learnt with a constraint-based algorithm. We use the term **pairwise relations** to denote adjacencies and non-adjacencies in a causal graph (ignoring orientations).

For each pair of variables, a constraint-based algorithm tries a number of conditional tests of independence. We use the maximum p-value obtained for every pair of variables as a representative of the corresponding pairwise relation. Posterior probabilities are then estimated as a function of these representative p-values. The method has no significant computational overhead, and can therefore scale up to the same number of variables as the algorithm of choice. Moreover, it does not depend on any additional assumptions (e.g. acyclicity, causal sufficiency, parametric assumptions) and can therefore be used with any constraint-based algorithm equipped with an appropriate test of conditional independence.

Notice that PROPeR *is not used to improve the algorithm per se, but to produce confidence estimates for pairwise relations learnt from the algorithm.* Identifying which parts of the learnt network are reliable is of great importance for practitioners who use causal discovery methods, and are often interested in high-confidence pairwise connections among variables or in avoiding a specific type of error (e.g. false positive or false negative edges). It can also be useful for selecting subsequent experiments for a system under study, by pointing out relationships that are uncertain.

We use the estimates obtained by PROPeR, to identify neighborhoods of high structural confidence in causal networks. The proposed method, called BiND (β-NeighborhooDs), takes as input a causal graph $\mathcal{G}$ along with representative p-values for every pairwise relation in $\mathcal{G}$ and a desired threshold of confidence β. The algorithm outputs all neighborhoods in $\mathcal{G}$ for which all pairwise relations have confidence estimates above β. Internally, BiND uses PROPeR to obtain probability estimates for each pairwise relation, creates a graph $\mathcal{H}_\beta$ where edges correspond to pairwise relations with confidence above β, and then uses the Bron-Kerbosch algorithm to identify all maximal cliques in graph $\mathcal{H}_\beta$.

In our empirical evaluation, we use simulated data to test the calibration of PROPeR' s probability estimates, and compare against two Bayesian methods that can be used to obtain similar estimates [1][2]. Results indicate that PROPeR produces reasonable probability estimates, while being significantly faster than other approaches. The behavior of BiND was also examined using simulated data sets. Results indicate that BiND identifies neighborhoods that include a smaller proportion of false positive and false negative edges, compared to the original induced network.

2 Background

We use V to denote random variables (interchangeably nodes of a causal graph), and bold upper-case letters to denote sets of variables. We use the notation $X \perp\!\!\!\perp Y|\mathbf{Z}$ to denote the independence of variables X and Y given the set of variables $\mathbf{Z}$. We use $\mathcal{G}{=}(\mathbf{V},\mathcal{E})$ to denote a graph over variables $\mathbf{V}$ with edges $\mathcal{E}$. For the scope of this work, we only deal with undirected edges, thus, members of $\mathcal{E}$ are unordered tuples of $\mathbf{V}$.

Bayesian networks consist of a Directed Acyclic Graph (DAG) $\mathcal{G}$ over a set of variables $\mathbf{V}$ and a joint probability distribution $\mathcal{P}$ over the same variables. A directed edge in $\mathcal{G}$ denotes a direct causal relation (in the context of measured variables). The DAG $\mathcal{G}$ and the distribution $\mathcal{P}$ are connected by the Causal Markov condition (CMC): Every variable is independent of its non-descendants given its parents. The graph in conjunction with the CMC entails a set of conditional independencies that hold in $\mathcal{P}$. The faithfulness condition (FC) states that all the conditional independencies that hold in $\mathcal{P}$ stem from $\mathcal{G}$ and the CMC, instead of being accidental parametric properties of the distribution.

Under CMC and FC, the conditional (in)dependencies that hold in $\mathcal{P}$ can be identified from the graph $\mathcal{G}$ according to a graphical criterion, namely *d*-separation. For graphs and distributions that are faithful to each other, we say that $\mathcal{P}$ satisfies the global Markov property with respect to $\mathcal{G}$: $X \perp\!\!\!\perp Y \ |\mathbf{Z}$ in $\mathcal{P}$ if and only if X and Y are *d*-separated given $\mathbf{Z}$ in $\mathcal{G}$. Constraint-based methods for learning Bayesian Networks use independence relations present in the data to constrain the search space of possible underlying causal graphs. The following theorem is the cornerstone of constraint-based causal learning:

Theorem 1. *[3] If $\langle \mathcal{G}, \mathcal{P} \rangle$ is a Bayesian network over $\mathbf{V}$ and $\mathcal{G}$ is faithful to $\mathcal{P}$, then the following holds: For every pair of variables $X, Y \in \mathbf{V}$: X and Y are not adjacent in $\mathcal{G} \leftrightarrow \exists \mathbf{Z} \subseteq V \setminus \{X, Y\}$ s.t. $X \perp\!\!\!\perp Y|\mathbf{Z}$.*

The theorem states that every missing edge in $\mathcal{G}$ corresponds to a conditional independence in $\mathcal{P}$. This is also known as the *pairwise Markov property.* Essentially, the theorem matches the skeleton of the causal graph to a kernel of conditional independencies (one for every missing edge). Thus, to identify the network skeleton, constraint-based algorithms use a search strategy to iterate over all pairs of variables in $\mathbf{V}$. For each such pair (X, Y), the algorithm tries to identify a set of variables $\mathbf{Z}$ that renders X and Y independent. If no such set exists, X and Y are adjacent in the resulting causal graph $\mathcal{G}$, otherwise the edge between them is removed and $\mathbf{Z}$ is reported as the separating set of X and Y. The set of all conditional independencies that hold in a probability distribution is called the **independence model** $\mathcal{J}$ of the distribution $\mathcal{P}$. Under CMC and FC, the minimal set of independencies identified by a (sound and complete) constraint-based algorithm are sufficient to entail *all* conditional independencies in $\mathcal{J}$.

Apart from CMC and FC, Bayesian networks rely on the assumption of causal sufficiency: Pairs of variables in a Bayesian network cannot be confounded, i.e. they cannot be effects of the same unmeasured common cause. This assumption is very restrictive and likely to be violated in many applications. Maximal Ancestral Graphs (MAGs) are extensions of Bayesian networks that can handle possible hidden confounders. In faithful MAGs, the graph $\mathcal{G}$ and the distribution $\mathcal{P}$ are connected through a graphical criterion similar to d-separation, called m-separation. MAGs also satisfy the pairwise Markov property: a missing edge in $\mathcal{G}$ corresponds to a conditional independence in $\mathcal{P}$. Edges and orientations in MAGs, however, have slightly different causal semantics than in Bayesian networks.

Methods presented in this work do not depend on the assumption of causal sufficiency, and can therefore work for both DAGs and MAGs. They do require, however, that the causal graph and the distribution satisfy the pairwise Markov property: every missing edge must correspond to a conditional independence. This holds for DAGs and MAGs, but is not true for all graphical models.

Typically, for a joint probability distribution $\mathcal{P}$ over a set of variables $\mathbf{V}$, there exists a class (instead of a single) of causal graphs (DAGs or MAGs) that entail all and only the conditional independencies that hold in $\mathcal{P}$. Causal graphs that belong to the same class, and cannot be distinguished based on conditional independencies alone, are called Markov Equivalent. For both DAGs and MAGs, Markov Equivalent graphs share the same skeleton, and vary in some of the orientations.

For the scope of this work, we only attempt to quantify our belief to the adjacency or non-adjacency of each pair of variables, regardless of orientations. Thus, we only need to take into account the output of *the skeleton identification step of a constraint-based algorithm.* In the remainder of this paper, we use $\mathcal{G}$ $=(\mathbf{V}, \mathcal{E})$ to denote the output such an algorithm, thus, the skeleton of a BN or a MAG *without orientations.*

3 Posterior Probabilities for Pairwise Relations

In this section, we present the PROPeR algorithm for estimating posterior probabilities of pairwise relations in causal networks. PROPeR takes as input the causal skeleton $\mathcal{G}$ returned by a constraint-based algorithm and a set of representative p-values and outputs a posterior probability estimate for every adjacency and non-adjacency in $\mathcal{G}$. We use $P(X—Y)$ and $P(\neg X—Y)$ to denote the posterior probability of the adjacency and non-adjacency of X and Y in $\mathcal{G}$, respectively.

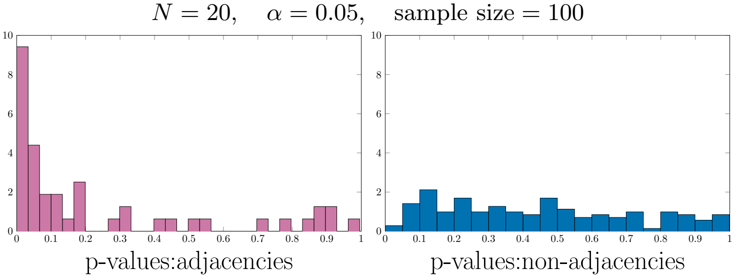


Fig. 1. Representative p-values for adjacencies and non-adjacencies. Normalized histograms of 190 representative p-values identified by the PC skeleton algorithm for a random network of 20 variables. p-values corresponding to adjacencies in the data-generating network (left) follow a distribution with decreasing density. p-values corresponding to non-adjacencies in the data-generating network (right) follow a uniform distribution in the interval $[\alpha, 1]$. A smaller number of predictions fall in the $[0, \alpha]$ interval. This bias is introduced due to constraint-base search strategy: while the representative p-value is below the threshold, the algorithm performs more tests. Naturally, in real scenarios, we do not know which p-values come from which distribution.

According to the pairwise Markov condition, a non-adjacency in a causal graph $\mathcal{G}$ over variables $\mathbf{V}$ corresponds to a conditional independence given a subset of $\mathbf{V}$. In contrast, an adjacency in $\mathcal{G}$ corresponds to the lack of such a subset: If X and Y are adjacent in $\mathcal{G}$, there exists no subset $\mathbf{Z}$ of observed variables such that $X \perp\!\!\!\perp Y|\mathbf{Z}$. Thus, edge X—Y will be present in $\mathcal{P}$ if the data support the null hypothesis

$$H_0 : \exists \mathbf{Z} \subset \mathbf{V} : X \perp\!\!\!\perp Y|\mathbf{Z} \textit{ less} \text{ than the alternative } H_1 : \forall \mathbf{Z} \subset \mathbf{V} : X \not\perp\!\!\!\perp Y|\mathbf{Z} \quad (1)$$

For a network with N variables, this complex set of hypotheses involves $|2^{N-2}|$ conditional independencies. To simplify Equation 1, we use a surrogate conditioning set. For each pair of variables, during the skeleton search, a constraint-based algorithm performs a number of tests, each for a different conditioning set. To avoid performing all possible tests, most algorithms avoid conditioning sets that are theoretically not likely to be d-separating the variables, and also use a threshold on the cardinality of attempted conditioning sets. Let p_{XY} be the maximum p-value of any attempted test of conditional independence between X and Y, and let $\mathbf{Z}_{XY}$ be the corresponding conditioning set. p_{XY} is used in constraint-based algorithms to determine whether X and Y are adjacent. If p_{XY} is lower than the threshold α, the edge is present in $\mathcal{G}$. Otherwise, the edge is absent in $\mathcal{G}$. We approximate Equation 1 with the following set of hypotheses:

$$H_0 : Ind(X, Y|\mathbf{Z}_{XY}) \text{ against the alternative } H_1 : \neg Ind(X, Y|\mathbf{Z}_{XY}), \qquad (2)$$

Under H_0, the p-values follow a uniform distribution. Under H_1, the p-values follow a distribution with decreasing density. Sellke et al. [4] propose using Beta alternatives to model the distribution of the p-values under the null and the alternative hypotheses, respectively: $Beta(1,1)$ is the uniform distribution and describes the distribution of the p-values under the null hypothesis. $Beta(\xi,1), \quad 0 < \xi < 1$ is a distribution defined in $(0,1)$ with density decreasing in p. It is therefore suitable to model the distribution of p-values under the alternative hypothesis. Figure 1 shows an example of the distributions of representative p-values under H_0 and H_1, identified using the PC skeleton on data simulated from a known network. Equation 2 can be re-formulated on the basis of the representative p-value:

$$H_0 : p_{XY} \sim Beta(1,1) \text{ against } H_1 : p_{XY} \sim Beta(\xi,1) \text{ for some } \xi \in (0,1). \quad (3)$$

We can now estimate whether adjacency is more probable than non-adjacency for a given representative p-value p, *by estimating which of the Beta alternatives it is most likely to follow.* We use $\mathbf{V}^2 = \{(X,Y), X,Y \in \mathbf{V}, X \neq Y\}$ to denote the set of unordered pairs of $\mathbf{V}$, i.e. the set of pairwise relations in a causal skeleton $\mathcal{G}$. Let $\mathbf{p} = \{p_{XY} : (X,Y) \in \mathbf{V}^2\}$ be the set of the representative p-values for each pairwise relation. We assume that this population of p-values follows a mixture of $Beta(\xi,1)$ and $Beta(1,1)$ distributions. If π_0 is the proportion of p-values following $Beta(1,1)$, then the corresponding probability density function is:

$$f(p|\xi,\pi_0) = \pi_0 + (1-\pi_0)\xi p^{\xi-1}$$

For given estimates $\hat{\pi}_0$ and $\hat{\xi}$, the posterior odds of H_0 against H_1 for variables X, Y is

$$\begin{aligned}\mathrm{PO}(p_{XY}) &= \frac{P(p_{XY}|H_0)P(H_0)}{P(p_{XY}|H_1)P(H_1)} = \\ \frac{P(p_{XY}|p_{XY} \sim Beta(1,1))P(p_{XY} \sim Beta(1,1))}{P(p_{XY}|p_{XY} \sim Beta(\hat{\xi},1))P(p_{XY} \sim Beta(\hat{\xi},1))} &= \frac{\hat{\pi}_0}{\hat{\xi} p_{XY}^{\hat{\xi}-1}(1-\hat{\pi}_0)}.\end{aligned} \quad (4)$$

Obviously, if $\mathrm{PO}(p_{XY}) > 1$, non-adjacency is more probable than adjacency for the pair of variables X, Y. Notice that for some $\hat{\xi}$ and $\hat{\pi}_0$, it is possible that $\mathrm{PO}(p_{XY}) > 1$, while X and Y are adjacent in $\mathcal{G}$.

Based on the ratios in Equation 4, we can obtain the probability estimates:

$$P(X—Y) = \frac{1}{1+\mathrm{PO}(p_{XY})}, \quad P(\neg X—Y) = \frac{\mathrm{PO}(p_{XY})}{1+\mathrm{PO}(p_{XY})} \quad (5)$$

To estimate the probabilities in Equation 5, we need to obtain estimates for $\hat{\pi}_0$ and $\hat{\xi}$. To estimate π_0, we use the method described in [5]. The authors propose fitting a natural cubic spline to the distribution of the p-values to estimate the proportion of p-values that come from the null hypothesis.

The method requires that the p-values are i.i.d., an assumption that is clearly violated for the sample of p-values obtained during a skeleton identification algorithm: Typically, the tests of independence attempted by constraint-based network learning algorithms depend on the results of previously attempted tests.

Algorithm 1. PROPeR

```
input  : causal network $\mathcal{G}$ over $\mathbf{V}$, representative p-values $\{p_{XY}\}$
output: Probability estimates $P(X—Y), P(\neg X—Y)$
Estimate $\hat{\pi_0}$ from $\{p_{XY}\}$ using the method described in [5];
Find $\hat{\xi}$ that minimizes $-\sum_{(X,Y)\in\mathbf{V}^2} log(\hat{\pi_0} + (1-\hat{\pi_0})\xi p_{XY}^{\xi-1})$;
foreach $(X,Y) \in \mathbf{V}^2$ with representative p-value $p_{XY}$ do
    $\mathrm{PO}(p_{XY}) \leftarrow \frac{\hat{\pi_0}}{\hat{\xi} p_{XY}^{\hat{\xi}-1}(1-\hat{\pi_0})}$;
    $P(X—Y) \leftarrow \frac{1}{\mathrm{PO}(p_{XY})+1}$,   $P(\neg X—Y) \leftarrow \frac{\mathrm{PO}(p_{XY})}{\mathrm{PO}(p_{XY})+1}$;
end
```

Moreover, each p_{XY} is the maximum among many attempted tests. Finally, the p-values coming from the null hypothesis are not uniform, since independence is only accepted if $p > \alpha$. Thus, the obtained estimate $\hat{\pi_0}$ may be biased. Nevertheless, we believe that the estimates produced using this method are reasonable approximations. An example of the distribution of representative p-values coming from H_0 and H_1 is illustrated in Figure 1.

For a given $\hat{\pi_0}$, the likelihood for a set of representative p-values $\{p_{XY}\}$ is

$$L(\xi) = \prod_{(X,Y)\in\mathbf{V}^2} (\hat{\pi_0} + (1-\hat{\pi_0})\xi p_{XY}^{\xi-1}).$$

The respective negative log likelihood is

$$-LL(\xi) = -\sum_{(X,Y)\in\mathbf{V}^2} log(\hat{\pi_0} + (1-\hat{\pi_0})\xi p_{XY}^{\xi-1}). \tag{6}$$

Equation 6 can easily be optimized for ξ. Algorithm 1 describes how to obtain probability estimates for all pairwise relations given their representative p-values.

4 Identifying Neighborhoods of High Structural Confidence

Algorithm 2 takes as input a causal skeleton $\mathcal{G}$, confidence estimates on $\mathcal{G}$'s pairwise relations and a confidence threshold β and outputs the set of all β-neighborhoods in $\mathcal{G}$. In the previous section we presented a method for obtaining posterior probability estimates for all pairwise relations in a causal skeleton. In this section, we will use these estimates to identify neighborhoods of high structural confidence on the same skeleton. We define a neighborhood of structural confidence β as follows:

Definition 1 (β-neighborhood). *Let $\mathcal{G} = (\mathbf{V}, \mathcal{E})$ be a causal skeleton, and $\{P_{XY},\ (X,Y) \in \mathbf{V}^2\}$ the set of probability estimates:*

$$P_{XY} = \begin{cases} P(X—Y), \text{ if } (X,Y) \text{ adjacent in } \mathcal{G} \\ P(\neg X—Y), \text{ if } (X,Y) \text{ not adjacent in } \mathcal{G} \end{cases}$$

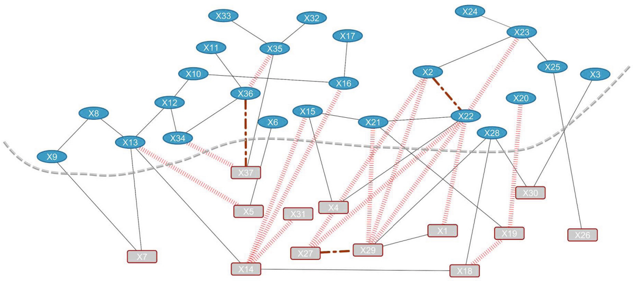


Fig. 2. An example maximum 0.8-neighborhood identified using Algorithm 2. We used the DAG of the Alarm network coupled with random parameters to simulate 100 samples. PC-skeleton was used to obtain the network skeleton $\mathcal{G}$, consisting of 34 edges: 31 true positive edges (solid lines in the figure) and 3 false positive edges ($- \cdot -$ lines). 15 edges were not identified by the algorithm, even though they are present in the data-generating graph (false negative edges, depicted as |||| lines). Algorithm 2 was used to identify the maximum 0.8-neighborhoods of $\mathcal{G}$. One of the maximum 0.8-neighborhoods, consisting of 24 variables that share 17 adjacencies, is noted: elliptical blue nodes denote variables in the neighborhood, while the remaining variables are shown as rectangular grey nodes (the neighborhood is also separated from the rest of the network with a dashed grey line). The proportion of false inferences within the clique is far lower than the overall proportion of false inferences: The clique includes only two false negative edges and only one false positive. Most of the false inferences are pairwise relations between members and non-members of the neighborhood.

*A subgraph $\mathcal{G}' = (\mathbf{V}', \mathcal{E}')$ of G is a β-**neighborhood** iff: $\forall X, Y \in \mathbf{V}' : P_{XY} > \beta$ The size of a β-neighborhood $\mathcal{G}' = (\mathbf{V}', \mathcal{E}')$ is $|\mathbf{V}'|$.*

Thus, a neighborhood of confidence β is a subgraph of the causal network in which the posterior probability of every pairwise relation is above a given threshold β. For a causal skeleton and a set of confidence estimates on all pairwise relations, finding a β - neighborhood can be reformulated as a graph theoretical problem: Let $\mathcal{H} = (\mathbf{V}, \mathcal{E}_\beta)$ be an undirected graph with edges defined as follows:

$$(X, Y) \in \mathcal{E}_\beta \text{ if } P_{XY} \geq \beta, \quad (X, Y) \notin \mathcal{E}_\beta \text{ if } P_{XY} < \beta \tag{7}$$

Variables X and Y are adjacent in $\mathcal{H}_\beta$ only if the probability of their respective pairwise relation in $\mathcal{G}$ is above the confidence threshold β. Finding β-neighborhoods in $\mathcal{G}$ is equivalent to identifying cliques in $\mathcal{H}_\beta$.

Naturally, a causal skeleton can have many β-neighborhoods. Moreover, if a subgraph $\mathcal{G}' = (\mathbf{V}', \mathcal{E}')$ of $\mathcal{G}$ is a β-neighborhood, then every subgraph of $\mathcal{G}'$ is a β-neighborhood. More interesting inferences may be made by identifying all **maximal** β-neighborhoods on a graph:

Definition 2. *Let $\mathcal{G}$=($\mathbf{V}$, $\mathcal{E}$) be a causal skeleton and $\mathcal{G}' = (\mathbf{V}', \mathcal{E}')$ be a β-neighborhood. $\mathcal{G}$ is a **maximal** β-neighborhood if $\nexists \mathbf{V}'' \supset \mathbf{V}'$ such that the subgraph $\mathcal{G}'' = (\mathbf{V}'', \mathcal{E}'')$ is a β-neighborhood.*

Algorithm 2. BiND

```
input  : causal network $\mathcal{G}$ over $\mathbf{V}$, pairwise confidence estimates $P_{XY}$,
         confidence threshold $\beta$
output: $\beta$-neighborhoods $\{\mathcal{G}'\}$
$\mathcal{H}_\beta \leftarrow$ empty graph;
foreach $(X, Y), X, Y \in \mathbf{V}$ do
    if $P_{XY} \geq \beta$ then add $(X, Y)$ to $\mathcal{H}_\beta$
end
$\{\mathbf{V}'\} \leftarrow$ Bron-Kerbosch($\mathcal{H}_\beta$);
$\{\mathcal{G}'\} \leftarrow$ subgraphs of $\mathcal{G}$ over $\{\mathbf{V}'\}$;
```

Thus, a maximal β-neighborhood is a β-neighborhood that is not part of a larger neighborhood. Identifying all maximal β-neighborhoods in $\mathcal{G}$ can be solved by finding all maximal cliques in the corresponding $\mathcal{H}_\beta$. Identifying maximal cliques is NP-hard [6], but algorithms that run in exponential time or identify approximate solutions are available. We use the Bron-Kerbosch algorithm [7].

Maximal cliques can often be very small; for example, if no larger cliques exist, all adjacencies and all non-adjacencies with $P_{XY} > \beta$ are (trivial) maximal cliques of size 2. Another interesting problem that could be solved using Algorithm 2 is to identify the **maximum** β-neighborhoods of a causal skeleton, i.e. the maximal β-neighborhoods with the maximum possible number of variables. This is equivalent to identifying all maximum cliques in $\mathcal{H}_\beta$, and can be easily obtained from the output of Algorithm 2. Figure 2 shows an example maximum clique, identified using Algorithm 2 on simulated data. The neighborhood includes 24 out of 37 variables. While the neighborhood includes more than half of the total variables and edges of $\mathcal{G}$, the number of false positive and false negative edges within the neighborhood is much lower than the corresponding number in the entire skeleton.

5 Related Work

Friedman et al. [8] propose a method for estimating probabilities on features of Bayesian networks. They use bootstrap to resample the data and learn a Bayesian network from each sampled data set. The probability of a structural feature is then estimated as the proportion of appearances of the feature in the resulting networks. Friedman and Koller [9] present a Bayesian method for estimating probabilities of features using MCMC samples over variable orderings. The methods are evaluated in terms of the classification performance (i.e. how accurately they accept or reject a feature), but not in terms of the calibration of predicted probability estimates.

Koivisto and Sood [10] and Koivisto [11] present algorithms for identifying exact posterior probabilities of edges in Bayesian networks. The methods use a dynamic programming strategy and constrain the search space of candidate causal models by bounding the number of possible parents per variable. The algorithms require a special type of non-uniform prior that does not respect Markov equivalence. Thus, resulting probabilities may be biased. Subsequent methods try to fix this problem by using MCMC simulations to compute network priors [2] or exploiting special types of nodes [12]. All methods in this category scale up to about 25 variables, since the minimum time and space requirement of these algorithms is $\mathcal{O}(n2^n)$.

Claasen and Heskes [1] propose a method for estimating Bayesian probabilities of a feature as a normalized sum of the posterior probabilities of all networks that entail this feature. The method requires exhaustive search of the space of possible networks, and is therefore not applicable for networks with more than 5-6 variables. The authors propose using this method as a standalone test of conditional independence, and also use it to decide on features inside a constraint-based algorithm. Pena, Kocka and Nielsen [13] estimate the confidence of a feature as the fraction of models containing the feature out of the different locally optimal models.

6 Experimental Evaluation

We performed a series of experiments to characterize the behavior of the proposed algorithms.

6.1 Calibration of Estimated Probabilities

We initially used simulated data to examine if the returned probability estimates are calibrated. We generated random DAGs with 10 and 20 variables, where each variable had 0 to 5 parents (randomly selected). The networks were then coupled with random parameters to create linear gaussian networks (continuous data) or discrete Bayesian networks (binary data). For continuous variables, a minimum correlation coefficient of 0.2 was imposed on the parameters to avoid weak interactions. We then simulated networks of various sample sizes, to test the method's behavior in different settings.

We used the PC skeleton identification step [3] with significance threshold $\alpha = 0.05$ and maximum conditioning set size 3 (explained below), modified to additionally return the maximum p-value encountered for each pair of variables. The set of maximum p-values was then used as input in Algorithm 1 to produce probability estimates for all pairwise relations. We compared our method against two alternative approaches:

1. **BCCD-P**: A method based on the BCCD algorithm presented in [1]. As mentioned above, the method estimates the posterior probability of a feature as a normalized sum of the posterior probabilities of DAGs that entail this feature. The algorithm scores all possible DAGs, and the authors use it to estimate probabilities for networks of at most 5 variables. To estimate the probabilities of pairwise relations, we scored the DAGs over variables X, Y and $\mathbf{Z}_{XY}$, where $\mathbf{Z}_{XY}$ is the conditioning set maximizing the p-value of

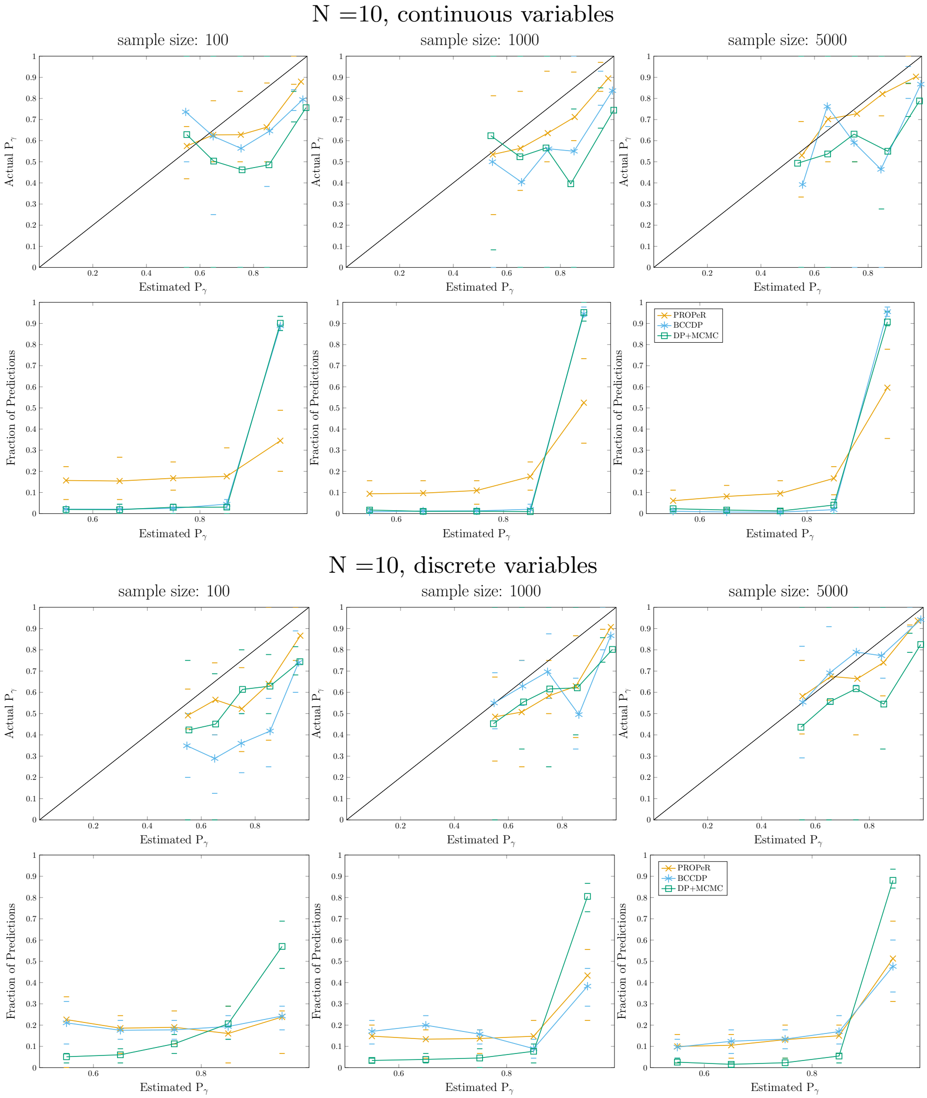


Fig. 3. Probability calibration plots for PROPeR, BCCD-P and MCMC+DP for networks of 10 variables. Bars indicate the quartiles. All methods tend to overestimate probabilities. Bayesian scoring methods are often very confident: For continuous variables, most of the probability estimates predicted by BCCD-P or MCMC+DP lie in the interval [0.9, 1], while MCMC+DP exhibits similar behavior for discrete variables also.

the tests $X \perp\!\!\!\perp Y | \mathbf{Z}$ performed by PC. This means that the cardinality of $\mathbf{Z}_{XY}$ cannot exceed 3. For a fair comparison, we used 3 as the maximum conditioning set of PC in all experiments. The probability of an adjacency

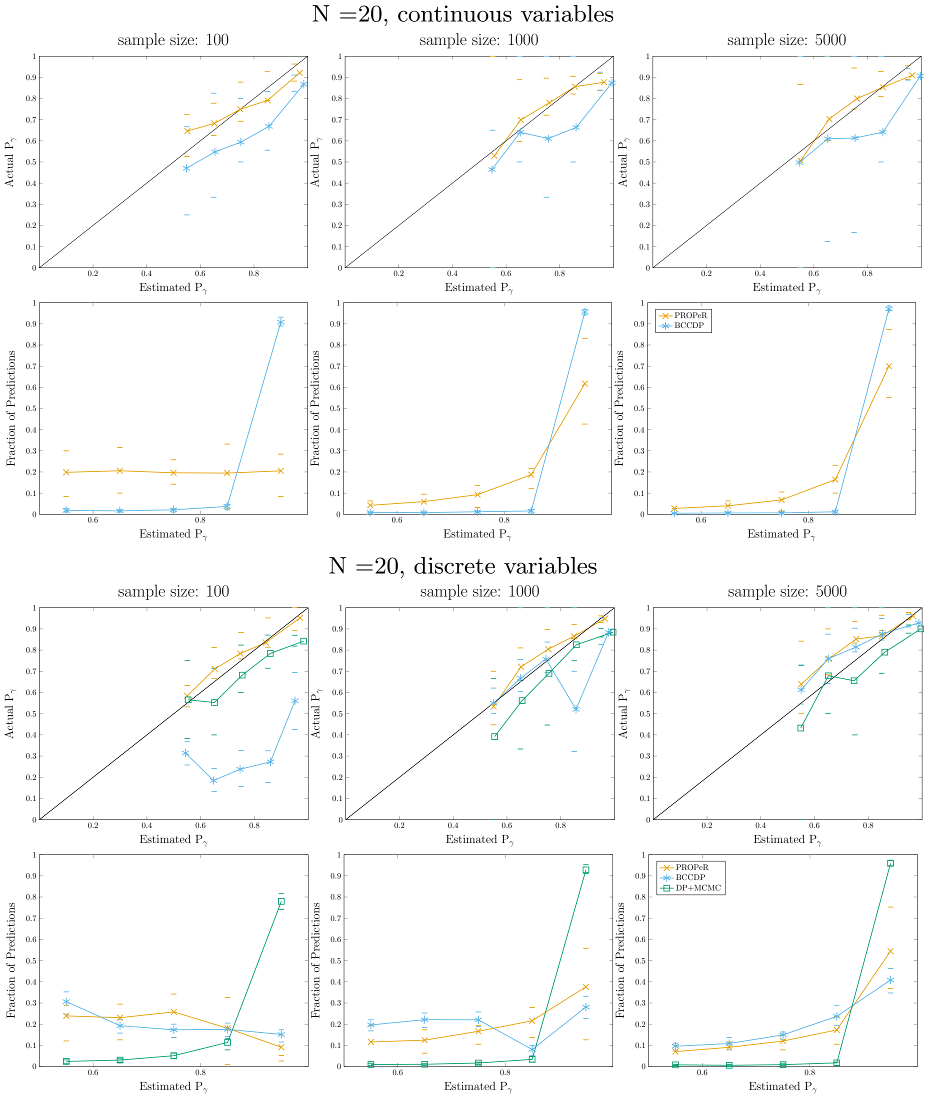


Fig. 4. Probability calibration plots for PROPeR, BCCD-P and MCMC+DP for networks of 20 variables. Bars indicate the quartiles. Similar to the results in Figure 3, Bayesian scoring methods tend to overestimate probabilities. MCMC+DP produced memory errors and failed to complete in all iterations for the BGE score, and is therefore not inlcuded in the corresponding plot.

was estimated as: $P(X—Y) = \sum_{\mathcal{G}\vdash X—Y} P(\mathbf{D}|\mathcal{G})P(\mathcal{G})$. Consistent priors described in [1] were pre-calculated and cached. To speed up the algorithm, we only scored one DAG per Markov equivalence class. For both approaches,

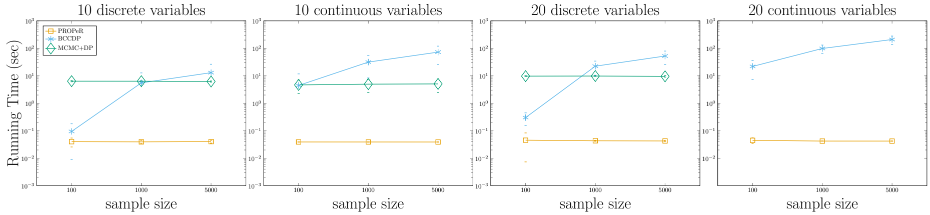


Fig. 5. Running times for PROPeR, BCCD-P and MCMC+DP

we used the BDe metric for discrete data and the BGe metric for gaussian data. Both metrics are score-equivalent.

2. **DP+ MCMC**: The method presented in [2] for identifying exact probabilities for edges in Bayesian networks. The method uses a combination of the DP algorithm [11] and MCMC sampling to correct the bias from the modular priors. We used the implementation provided by the authors in the BDAGL package. Maximum parents was set to 5, and the default parameters suggested by the authors in the package documentation were used. The method estimates probability estimates for directed edges, so we used $P(X\text{—}Y) = P(X\longrightarrow Y) + P(Y\longrightarrow X)$, $P(\neg X\text{—}Y) = 1 - P(X\text{—}Y)$.

To produce the probability calibration plots, the resulting predicted probabilities in [0.5, 1] were binned in 5 intervals. For every pair of variables, $P(X\text{—}Y)$ =1-$P(\neg X\text{—}Y)$. Thus, to consider each estimate once, we only need to consider half of the interval [0, 1]. If N pairwise relations have probability estimates $\{\hat{P}_i\}_{i=1}^{N}$ that lie in interval $[\gamma, \gamma + 0.1]$, we expect that $\bar{\hat{P}}_i \times N$ of the corresponding relations will be true. The actual probability P_γ for each interval is the fraction of relations with probability estimates in the given interval that are actually true in the data-generating graph. Figures 3 and 4 illustrate the mean estimated versus the mean actual probability for each bin, as well as the fraction of predictions in each bin for networks with 10 and 20 variables. Running times for all methods are shown in Figure 5.

Overall, results indicate that:

- PROPeR produces reasonable probability estimates, particularly in comparison to the more expensive BCCD-P and MCMC+DP approaches.
- MCMC+DP tends to identify very high (resp. very low) probabilities for the pairwise relations, even for small sample sizes for both metrics (BGE and BDE). BCCD has similar behavior for the BGE score, but not for the BDE score. This could explain the large deviations (and the seemingly unpredictable behavior) observed for these algorithms in the first four bins ([0.5 0.9]), since the means are computed over very few data points.
- As far as running times are concerned, both BCCD-P and MCMC+DP algorithms have (theoretically) exponential complexity with respect to the number of variables. BCCD-P also increases exponentially with sample size, but this is probably due to an increase in maximum conditioning set sizes

reported by PC skeleton for larger sample sizes. i.e., BCCD-P iterates networks with many variables (4-5) for most pairwise relations. This also explains the poor performance of BCCD-P for the BDE metric and sample size 100: estimates are obtained by scoring smaller networks. The employed implementation of MCMC+DP failed to complete any iterations for N=20 and continuous variables.

We must point out that the calibration of the probability estimates *is not necessarily related to the predictive power of the respective approaches*, which depends more on the relative ranking of probabilities among pairwise relations, rather than the actual estimates. For example, MCMC+DP has been shown to produce rankings of edges with very high AUC [2]. For the purposes of this work, however, obtaining estimates that are calibrated is important for identifying neighborhoods of a user-defined confidence. For example, using MCMC+DP estimates in Algorithm 2 would result in an almost fully connected $\mathcal{H}_{0.9}$, since most of the pairwise relations have probability estimates above this threshold.

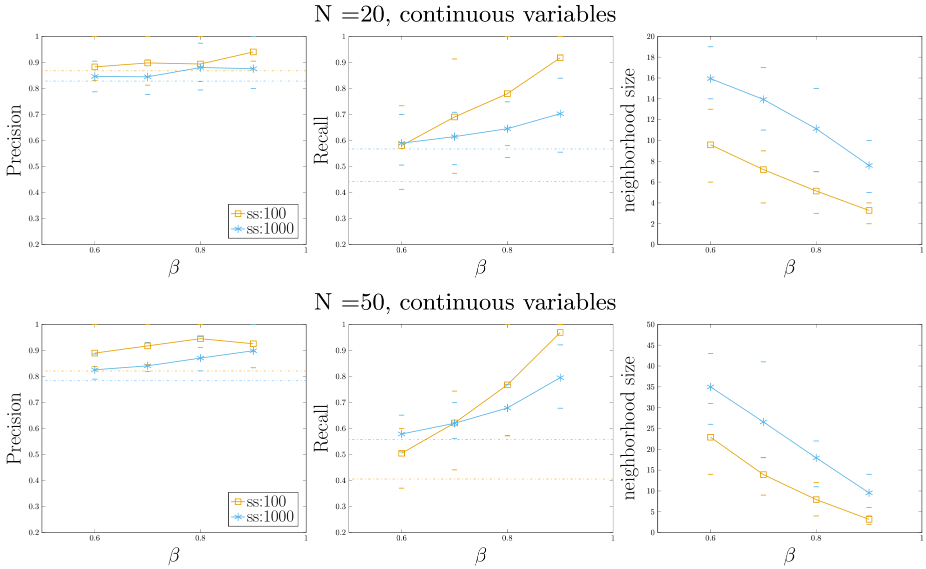
6.2 Evaluation of Neighborhoods Identified with BiND

To demonstrate the value of BiND, we simulated data of 100 and 1000 samples from random networks with 20 and 50 variables, as described above. For the causal skeletons identified with the PC skeleton algorithm and the posterior probability estimates produced by PROPeR, all maximal β-neighborhoods for β=0.6, 0.7 and 0.9 were identified using Algorithm 2.

We examined the structural precision ($\frac{\#\text{ edges in } \mathcal{G}' \text{ and the ground truth}}{\#\text{ edges in } \mathcal{G}'}$) and recall ($\frac{\#\text{ edges in } \mathcal{G}' \text{ and the ground truth}}{\#\text{ edges in the ground truth}}$) of the resulting neighborhoods, compared to the baseline precision and recall for $\mathcal{G}$. As mentioned above,the maximal cliques can be very small and uninformative, particularly for high confidence thresholds. We are more interested in identifying large parts of the networks that we are confident about, and therefore focused in the maximum β-neighborhoods. Figure 6 illustrates the precision, recall and size of maximum β-neighborhoods for networks of 20 and 50 variables, for both discrete and continuous data. The algorithm took 85.56 seconds on average to identify the maximum 0.6-neighborhoods for 50 variables and 1000 samples (the most expensive case). Detailed time results are omitted due to space limitations.

Results indicate the following:

- The method identifies subgraphs with lower ratios of false inferences compared to the entire skeleton.
- For high confidence thresholds and small sample sizes, the algorithm cannot identify large neighborhoods.
- The algorithm is particularly useful in small sample sizes, where the overall recall is very low.


N =20, continuous variables

N =50, continuous variables

Fig. 6. Precision, recall, number and size of maximum cliques identified using BiND in networks of 20 continuous variables. Bars indicate quartiles. Dashed horizontal lines show the mean baseline precision and recall (mean precision and recall of the output of PC skeleton. BiND identifies neighborhoods of higher structural precision and recall than the corresponding baseline, particularly for small sample sizes.

7 Discussion

Equipping constraint-based causal discovery algorithms with a method that can provide some measure of confidence on their output improves their usability. Bayesian scoring and bootstrapping methods can be employed for this purpose, but are computationally expensive and do not scale up to the number of variables constraint-based algorithms can handle.

We have presented PROPeR, an algorithm for estimating posterior probabilities of adjacencies and non-adjacencies in networks learnt using constraint-based methods. The algorithm has no significant computational overhead and is scalable to practically any input size: increasing the number of variables processed by the constraint-based algorithm merely increases the sample size of p-values on which PROPeR fits a probability density function. PROPeR is shown to produce calibrated probability estimates, while being significantly faster than other state of the art algorithms. We have also presented BiND, an algorithm that identifies the maximal (or maximum) neighborhoods of high confidence on a causal network. In simulated scenarios, the algorithm is able to identify neighborhoods that are indeed more reliable.

PROPeR and BiND can easily accompany any constraint-based algorithm on any type of data, provided an appropriate test of conditional independence is available. Estimating posterior probabilities based on p-values can be of use in several causal discovery tasks, including conflict resolution, improving orientations, and experiment selection.

Acknowledgements. ST and IT were funded by the STATegra EU FP7 project, No 306000. IT was partially funded by the EPILOGEAS GSRT ARISTEIA II project, No 3446.

References

1. Claassen, T., Heskes, T.: A Bayesian approach to constraint based causal inference. In: Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence, pp. 2992–2996 (2012)
2. Eaton, D., Murphy, K.P.: Exact bayesian structure learning from uncertain interventions. In: Proceedings of the 11th International Conference on Artificial Intelligence and Statistics, pp. 107–114 (2007)
3. Spirtes, P., Glymour, C., Scheines, R.: Causation, prediction, and search, vol. 81. MIT Press (2000)
4. Sellke, T., Bayarri, M., Berger, J.: Calibration of ρ values for testing precise null hypotheses. The American Statistician 55(1), 62–71 (2001)
5. Storey, J., Tibshirani, R.: Statistical significance for genomewide studies. PNAS 100(16), 9440 (2003)
6. Karp, R.: Reducibility Among Combinatorial Problems. In: Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)
7. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Communications of the ACM 16(9), 575–577 (1973)
8. Friedman, N., Goldszmidt, M., Wyner, A.: On the application of the bootstrap for computing confidence measures on features of induced Bayesian networks. In: Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics, pp. 196–205 (1999)
9. Friedman, N., Koller, D.: Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. Machine Learning 50(1-2), 95–125 (2003)
10. Koivisto, M., Sood, K.: Exact Bayesian structure discovery in Bayesian networks. JMLR 5, 549–573 (2004)
11. Koivisto, M.: Advances in exact Bayesian structure discovery in Bayesian networks. In: Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence, pp. 241–248 (2006)
12. Tian, J., He, R.: Computing posterior probabilities of structural features in Bayesian networks. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, pp. 538–547 (2009)
13. Pena, J., Kocka, T., Nielsen, J.: Featuring multiple local optima to assist the user in the interpretation of induced Bayesian Network models. In: Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 1683–1690 (2004)