Journal of Machine Learning Research 8 (2007) 613-636 Submitted 9/06; Revised 1/07; Published 3/07

Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm

Markus Kalisch KALISCH@STAT.MATH.ETHZ.CH
Peter Bühlmann BUHLMANN@STAT.MATH.ETHZ.CH
Seminar für Statistik
ETH Zurich
8092 Zürich, Switzerland

Editor: David Maxwell Chickering

Abstract

We consider the PC-algorithm (Spirtes et al., 2000) for estimating the skeleton and equivalence class of a very high-dimensional directed acyclic graph (DAG) with corresponding Gaussian distribution. The PC-algorithm is computationally feasible and often very fast for sparse problems with many nodes (variables), and it has the attractive property to automatically achieve high computational efficiency as a function of sparseness of the true underlying DAG. We prove uniform consistency of the algorithm for very high-dimensional, sparse DAGs where the number of nodes is allowed to quickly grow with sample size n, as fast as $O(n^a)$ for any $0 < a < \infty$. The sparseness assumption is rather minimal requiring only that the neighborhoods in the DAG are of lower order than sample size n. We also demonstrate the PC-algorithm for simulated data.

Keywords: asymptotic consistency, DAG, graphical model, PC-algorithm, skeleton

1. Introduction

Graphical models are a popular probabilistic tool to analyze and visualize conditional independence relationships between random variables (see Edwards, 2000; Lauritzen, 1996; Neapolitan, 2004). Major building blocks of the models are nodes, which represent random variables and edges, which encode conditional dependence relations of the enclosing vertices. The structure of conditional independence among the random variables can be explored using the Markov properties.

Of particular current interest are directed acyclic graphs (DAGs), containing directed rather than undirected edges, which restrict in a sense the conditional dependence relations. These graphs can be interpreted by applying the directed Markov property (see Lauritzen, 1996). When ignoring the directions of a DAG, we get the skeleton of a DAG. In general, it is different from the conditional independence graph (CIG), see Section 2.1. (Thus, estimation methods for directed graphs cannot be easily borrowed from approaches for undirected CIGs.) As we will see in Section 2.1, the skeleton can be interpreted easily and thus yields interesting insights into the dependence structure of the data.

Estimation of a DAG from data is difficult and computationally non-trivial due to the enormous size of the space of DAGs: the number of possible DAGs is super-exponential in the number of nodes (see Robinson, 1973). Nevertheless, there are quite successful search-and-score methods for problems where the number of nodes is small or moderate. For example, the search space may be restricted to trees as in MWST (Maximum Weight Spanning Trees; see Chow and Liu, 1968; Heckerman et al., 1995), or a greedy search is employed. The greedy DAG search can be improved by exploiting probabilistic equivalence relations, and the search space can be reduced from individual DAGs to equivalence classes, as proposed in GES (Greedy Equivalent Search, see Chickering, 2002a). Although this method seems quite promising when having few or a moderate number of nodes, it is limited by the fact that the space of equivalence classes is conjectured to grow super-exponentially in the nodes as well (Gillispie and Perlman, 2001). Bayesian approaches for DAGs, which are computationally very intensive, include Spiegelhalter et al. (1993) and Heckerman et al. (1995).

An interesting alternative to greedy or structurally restricted approaches is the PC-algorithm (after its authors, Peter and Clark) from Spirtes et al. (2000). It starts from a complete, undirected graph and deletes recursively edges based on conditional independence decisions. This yields an undirected graph which can then be partially directed and further extended to represent the underlying DAG (see later). The PC-algorithm runs in the worst case in exponential time (as a function of the number of nodes), but if the true underlying DAG is sparse, which is often a reasonable assumption, this reduces to a polynomial runtime.

In the past, interesting hybrid methods have been developed. Very recently, Tsamardinos et al. (2006) proposed a computationally very competitive algorithm. We also refer to their paper for a quite exhaustive numerical comparison study among a wide range of algorithms.

We focus in this paper on estimating the equivalence class and the skeleton of DAGs (corresponding to multivariate Gaussian distributions) in the high-dimensional context, that is, the number of nodes p may be much larger than sample size n. We prove that the PC-algorithm consistently estimates the equivalence class and the skeleton of an underlying sparse DAG, as sample size $n \to \infty$, even if $p = p_n = O(n^a)$ $(0 \leq a < \infty)$ is allowed to grow very quickly as a function of n.

Our implementation of the PC-algorithm is surprisingly fast, as illustrated in Section 4.5, and it allows us to estimate a sparse DAG even if p is in the thousands. For the high-dimensional setting with $p \gg n$, sparsity of the underlying DAG is crucial for statistical consistency and computational feasibility. Our analysis seems to be the first establishing a provable correct algorithm (in an asymptotic sense) for high-dimensional DAGs which is computationally feasible.

The question of consistency of a class of methods including the PC algorithm has been treated in Spirtes et al. (2000) and Robins et al. (2003) in the context of causal inference. They show that, assuming only faithfulness (explained in Section 2), uniform consistency cannot be achieved, but pointwise consistency can. In this paper, we extend this in two ways: We provide a set of assumptions which renders the PC-algorithm to be uniformly consistent. More importantly, we show that consistency holds even as the number of nodes and neighbors increases and the size of the smallest non-zero partial correlations decrease as a function of the sample size. Stricter assumptions than the faithfulness condition that render uniform consistency possible have been also proposed in Zhang and Spirtes (2003). A rather general discussion on how many samples are needed to learn the correct structure of a Bayesian Network can be found in Zuk et al. (2006).

The problem of finding the equivalence class of a DAG has a substantial overlap with the problem of feature selection: If the equivalence class is found, the Markov Blanket of any variable (node) can be read of easily. Given a set of nodes V and suppose that M is the Markov Blanket of node X, then X is conditionally independent of $V \backslash M$ given M. Thus, M contains all and only the relevant features for X. In recent years, many other approaches to feature selection have been developed for high dimensions. See for example Goldenberg and Moore (2004) for an approach dealing with very high dimensions or Ng (1998) for a rather general approach dealing with bounds for generalization errors.

2. Finding the Equivalence Class of a DAG

In this section, we first present the main concepts. Then, we give a detailed description of the PC-algorithm.

2.1 Definitions and Preliminaries

A graph $G = (V,E)$ consists of a set of nodes or vertices $V = \{1,\dots,p\}$ and a set of edges $E \subseteq V \times V$, that is, the edge set is a subset of ordered pairs of distinct nodes. In our setting, the set of nodes corresponds to the components of a random vector $\mathbf{X} \in \mathbb{R}^p$. An edge $(i, j) \in E$ is called directed if $(i, j) \in E$ but $(j,i) \notin E$; we then use the notation $i \rightarrow j$. If both $(i, j) \in E$ and $(j,i) \in E$, the edge is called undirected; we then use the notation $i - j$. A directed acyclic graph (DAG) is a graph G where all edges are directed and not containing any cycle.

If there is a directed edge $i \rightarrow j$, node i is said to be a parent of node j. The set of parents of node j is denoted by $pa(j)$. The adjacency set of a node j in graph G, denoted by $adj(G, j)$, are all nodes i which are directly connected to j by an edge (directed or undirected). The elements of $adj(G, j)$ are also called neighbors of or adjacent to j.

A probability distribution P on $\mathbb{R}^p$ is said to be faithful with respect to a graph G if conditional independencies of the distribution can be inferred from so-called d-separation in the graph G and vice-versa. More precisely: consider a random vector $\mathbf{X} \sim P$. Faithfulness of P with respect to G means: for any $i, j \in V$ with $i \neq j$ and any set $\mathbf{s} \subseteq V$,

$$\begin{aligned} & \mathbf{X}^{(i)} \text{ and } \mathbf{X}^{(j)} \text{ are conditionally independent given } \{\mathbf{X}^{(r)};\ r \in \mathbf{s}\} \\ \Leftrightarrow\quad & \text{node } i \text{ and node } j \text{ are d-separated by the set } \mathbf{s}. \end{aligned}$$

The notion of d-separation can be defined via moral graphs; details are described in Lauritzen (1996, Prop. 3.25). We remark here that faithfulness is ruling out some classes of probability distributions. An example of a non-faithful distribution is given in Spirtes et al. (2000, Chapter 3.5.2). On the other hand, non-faithful distributions of the multivariate normal family (which we will limit ourselves to) form a Lebesgue null-set in the space of distributions associated with a DAG G, see Meek (1995a).

The skeleton of a DAG G is the undirected graph obtained from G by substituting undirected edges for directed edges. A v-structure in a DAG G is an ordered triple of nodes (i, j, k) such that G contains the directed edges $i \rightarrow j$ and $k \rightarrow j$, and i and k are not adjacent in G.

It is well known that for a probability distribution P which is generated from a DAG G, there is a whole equivalence class of DAGs with corresponding distribution P (see Chickering, 2002a, Section 2.2). Even when having infinitely many observations, we cannot distinguish among the different DAGs of an equivalence class. Using a result from Verma and Pearl (1990), we can characterize equivalent classes more precisely: Two DAGs are equivalent if and only if they have the same skeleton and the same v-structures.

A common tool for visualizing equivalence classes of DAGs are *completed partially directed acyclic graphs (CPDAG)*. A partially directed acyclic graph (PDAG) is a graph where some edges are directed and some are undirected and one cannot trace a cycle by following the direction of directed edges and any direction for undirected edges. Equivalence among PDAGs or of PDAGs and DAGs can be decided as for DAGs by inspecting the skeletons and v-structures. A PDAG is *completed*, if (1) every directed edge exists also in every DAG belonging to the equivalence class of the DAG and (2) for every undirected edge $i - j$ there exists a DAG with $i \rightarrow j$ and a DAG with $i \leftarrow j$ in the equivalence class.

CPDAGs encode all independence informations contained in the corresponding equivalence class. It was shown in Chickering (2002b) that two CPDAGs are identical if and only if they represent the same equivalence class, that is, they represent a equivalence class uniquely.

Although the main goal is to identify the CPDAG, the skeleton itself already contains interesting information. In particular, if P is faithful with respect to a DAG G,

$$\begin{aligned} & \text{there is an edge between nodes } i \text{ and } j \text{ in the skeleton of DAG } G \\ \Leftrightarrow\ & \text{for all } \mathbf{s} \subseteq V \setminus \{i,j\},\ \mathbf{X}^{(i)} \text{ and } \mathbf{X}^{(j)} \text{ are conditionally dependent} \\ & \qquad\qquad\qquad\qquad \text{given } \{\mathbf{X}^{(r)};\ r \in \mathbf{s}\}, \end{aligned} \qquad (1)$$

(Spirtes et al., 2000, Th. 3.4). This implies that if P is faithful with respect to a DAG G, the skeleton of the DAG G is a subset (or equal) to the conditional independence graph (CIG) corresponding to P. (The reason is that an edge in a CIG requires only conditional dependence given the set $V \setminus \{i,j\}$). More importantly, every edge in the skeleton indicates some strong dependence which cannot be explained away by accounting for other variables. We think, that this property is of value for exploratory analysis.

As we will see later in more detail, estimating the CPDAG consists of two main parts (which will naturally structure our analysis): (1) Estimation of the skeleton and (2) partial orientation of edges. All statistical inference is done in the first part, while the second is just application of deterministic rules on the results of the first part. Therefore, we will put much more emphasis on the analysis of the first part. If the first part is done correctly, the second will never fail. If, however, there occur errors in the first part, the second part will be more sensitive to it, since it depends on the inferential results of part (1) in greater detail. Therefore, when dealing with a high-dimensional setting (large p, small n), the CPDAG is harder to recover than the skeleton. Moreover, the interpretation of the CPDAG depends much more on the global correctness of the graph. The interpretation of the skeleton, on the other hand, depends only on a local region and is thus more reliable.

We conclude that, if the true underlying probability mechanisms are generated from a DAG, finding the CPDAG is the main goal. The skeleton itself oftentimes already provides interesting insights, and in a high-dimensional setting it might be interesting to use the undirected skeleton as an alternative target to the CPDAG when finding a useful approximation of the CPDAG seems hopeless.

As mentioned before, we will in the following describe two main steps. First, we will discuss the part of the PC-algorithm that leads to the skeleton. Afterwards we will complete the algorithm by discussing the extensions for finding the CPDAG. We will use the same format when discussing theoretical properties of the PC-algorithm.

2.2 The PC-algorithm for Finding the Skeleton

A naive strategy for finding the skeleton would be to check conditional independencies given all subsets $\mathbf{s} \subseteq V \setminus \{i,j\}$ (see Formula 1), that is, all partial correlations in the case of multivariate normal distributions as first suggested by Verma and J.Pearl (1991). This would become computationally infeasible and statistically ill-posed for p larger than sample size. A much better approach is used by the PC-algorithm which is able to exploit sparseness of the graph. More precisely, we apply the part of the PC-algorithm that identifies the undirected edges of the DAG.

2.2.1 POPULATION VERSION FOR THE SKELETON

In the population version of the PC-algorithm, we assume that perfect knowledge about all necessary conditional independence relations is available. We refer here to the PC-algorithm what others call the first part of the PC-algorithm; the other part is described in Algorithm 2 in Section 2.3.

Algorithm 1 The PC$_{pop}$-algorithm

```
INPUT: Vertex Set $V$, Conditional Independence Information
OUTPUT: Estimated skeleton $C$, separation sets $S$ (only needed when directing the skeleton
  afterwards)
Form the complete undirected graph $\tilde{C}$ on the vertex set V.
$\ell = -1$;    $C = \tilde{C}$
repeat
  $\ell = \ell + 1$
  repeat
    Select a (new) ordered pair of nodes $i, j$ that are adjacent in $C$ such that $|adj(C,i) \setminus \{j\}| \geq \ell$
    repeat
      Choose (new) $\mathbf{k} \subseteq adj(C,i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$.
      if $i$ and $j$ are conditionally independent given $\mathbf{k}$ then
        Delete edge $i, j$
        Denote this new graph by $C$
        Save $\mathbf{k}$ in $S(i, j)$ and $S(j, i)$
      end if
    until edge $i, j$ is deleted or all $\mathbf{k} \subseteq adj(C,i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$ have been chosen
  until all ordered pairs of adjacent variables $i$ and $j$ such that $|adj(C,i) \setminus \{j\}| \geq \ell$ and $\mathbf{k} \subseteq$
    $adj(C,i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$ have been tested for conditional independence
until for each ordered pair of adjacent nodes $i, j$: $|adj(C,i) \setminus \{j\}| < \ell$.
```

The (first part of the) PC-algorithm is given in Algorithm 1. The maximal value of ℓ in Algorithm 1 is denoted by

$$m_{reach} = \text{ maximal reached value of } \ell. \tag{2}$$

The value of m_{reach} depends on the underlying distribution.

A proof that this algorithm produces the correct skeleton can be easily deduced from Theorem 5.1 in Spirtes et al. (2000). We summarize the result as follows.

Proposition 1 *Consider a DAG G and assume that the distribution P is faithful to G. Denote the maximal number of neighbors by* $q = \max_{1 \leq j \leq p} |adj(G, j)|$. *Then, the PC$_{pop}$-algorithm constructs the true skeleton of the DAG. Moreover, for the reached level:* $m_{reach} \in \{q-1, q\}$.

A proof is given in Section A.

2.2.2 SAMPLE VERSION FOR THE SKELETON

For finite samples, we need to estimate conditional independencies. We limit ourselves to the Gaussian case, where all nodes correspond to random variables with a multivariate normal distribution. Furthermore, we assume faithful models, which means that the conditional independence relations correspond to d-separations (and so can be read off the graph) and vice versa; see Section 2.1.

In the Gaussian case, conditional independencies can be inferred from partial correlations.

Proposition 2 *Assume that the distribution P of the random vector $\mathbf{X}$ is multivariate normal. For $i \neq j \in \{1,\dots,p\},\ \mathbf{k} \subseteq \{1,\dots,p\} \setminus \{i,j\}$, denote by $\rho_{i,j|\mathbf{k}}$ the partial correlation between $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ given $\{\mathbf{X}^{(r)};\ r \in \mathbf{k}\}$. Then, $\rho_{i,j|\mathbf{k}} = 0$ if and only if $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ are conditionally independent given $\{\mathbf{X}^{(r)};\ r \in \mathbf{k}\}$.*

Proof: The claim is an elementary property of the multivariate normal distribution, see Lauritzen (1996, Prop. 5.2.). □

We can thus estimate partial correlations to obtain estimates of conditional independencies. The sample partial correlation $\hat{\rho}_{i,j|\mathbf{k}}$ can be calculated via regression, inversion of parts of the covariance matrix or recursively by using the following identity: for some $h \in \mathbf{k}$,

$$\rho_{i,j|\mathbf{k}} = \frac{\rho_{i,j|\mathbf{k}\setminus h} - \rho_{i,h|\mathbf{k}\setminus h}\rho_{j,h|\mathbf{k}\setminus h}}{\sqrt{(1-\rho^2_{i,h|\mathbf{k}\setminus h})(1-\rho^2_{j,h|\mathbf{k}\setminus h})}}.$$

In the following, we will concentrate on the recursive approach. For testing whether a partial correlation is zero or not, we apply Fisher's z-transform

$$Z(i,j|\mathbf{k}) = \frac{1}{2}\log\left(\frac{1+\hat{\rho}_{i,j|\mathbf{k}}}{1-\hat{\rho}_{i,j|\mathbf{k}}}\right). \tag{3}$$

Classical decision theory yields then the following rule when using the significance level α. Reject the null-hypothesis $H_0(i,j|\mathbf{k}):\ \rho_{i,j|\mathbf{k}} = 0$ against the two-sided alternative $H_A(i,j|\mathbf{k}):\ \rho_{i,j|\mathbf{k}} \neq 0$ if $\sqrt{n-|\mathbf{k}|-3}|Z(i,j|\mathbf{k})| > \Phi^{-1}(1-\alpha/2)$, where $\Phi(\cdot)$ denotes the cdf of $\mathcal{N}(0,1)$.

The sample version of the PC-algorithm is almost identical to the population version in Section 2.2.1.

The PC-algorithm

> Run the PC$_{pop}$(m)-algorithm as described in Section 2.2.1 but replace in line 11 of Algorithm 1 the if-statement by
> **if** $\sqrt{n-|\mathbf{k}|-3}|Z(i,j|\mathbf{k})| \leq \Phi^{-1}(1-\alpha/2)$ **then**.

The algorithm yields a data-dependent value $\hat{m}_{reach,n}$ which is the sample version of (2).

The only tuning parameter of the PC-algorithm is α, which is the significance level for testing partial correlations. See Section 4 for further discussion.

As we will see below in Section 3, the algorithm is asymptotically consistent even if p is much larger than n but the DAG is sparse.

2.3 Extending the Skeleton to the Equivalence Class

While finding the skeleton as in Algorithm 1, we recorded the separation sets that made edges drop out in the variable denoted by S. This was not necessary for finding the skeleton itself, but will be essential for extending the skeleton to the equivalence class. In Algorithm 2 we describe the work of Pearl (2000, p.50f) to extend the skeleton to a CPDAG belonging to the equivalence class of the underlying DAG.

Algorithm 2 Extending the skeleton to a CPDAG

```
INPUT: Skeleton $G_{skel}$, separation sets $S$
OUTPUT: CPDAG $G$
for all pairs of nonadjacent variables $i, j$ with common neighbour $k$ do
    if $k \notin S(i, j)$ then
        Replace $i - k - j$ in $G_{skel}$ by $i \rightarrow k \leftarrow j$
    end if
end for
In the resulting PDAG, try to orient as many undirected edges as possible by repeated application
of the following three rules:
R1 Orient $j - k$ into $j \rightarrow k$ whenever there is an arrow $i \rightarrow j$ such that $i$ and $k$ are nonadjacent.
R2 Orient $i - j$ into $i \rightarrow j$ whenever there is a chain $i \rightarrow k \rightarrow j$.
R3 Orient $i - j$ into $i \rightarrow j$ whenever there are two chains $i - k \rightarrow j$ and $i - l \rightarrow j$ such that $k$ and
$l$ are nonadjacent.
R4 Orient $i - j$ into $i \rightarrow j$ whenever there are two chains $i - k \rightarrow l$ and $k \rightarrow l \rightarrow j$ such that $k$ and
$l$ are nonadjacent.
```

The output of Algorithm 2 is a CPDAG, which was first proved by Meek (1995b).

3. Consistency for High-Dimensional Data

As in Section 2, we will first deal with the problem of finding the skeleton. Consecutively, we will extend the result to finding the CPDAG.

3.1 Finding the Skeleton

We will show that the PC-algorithm from Section 2.2.2 is asymptotically consistent for the skeleton of a DAG, even if p is much larger than n but the DAG is sparse. We assume that the data are realizations of i.i.d. random vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n$ with $\mathbf{X}_i \in \mathbb{R}^p$ from a DAG G with corresponding distribution P. To capture high-dimensional behavior, we will let the dimension grow as a function of sample size: thus, $p = p_n$ and also the DAG $G = G_n$ and the distribution $P = P_n$. Our assumptions are as follows.

(A1) The distribution P_n is multivariate Gaussian and faithful to the DAG G_n for all n.

(A2) The dimension $p_n = O(n^a)$ for some $0 \leq a < \infty$.

(A3) The maximal number of neighbors in the DAG G_n is denoted by $q_n = \max_{1 \leq j \leq p_n} |adj(G, j)|$, with $q_n = O(n^{1-b})$ for some $0 < b \leq 1$.

(A4) The partial correlations between $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ given $\{\mathbf{X}^{(r)}; r \in \mathbf{k}\}$ for some set $\mathbf{k} \subseteq \{1, \ldots, p_n\} \setminus \{i, j\}$ are denoted by $\rho_{n;i,j|\mathbf{k}}$. Their absolute values are bounded from below and above:

$$\begin{aligned}&\inf\{|\rho_{i,j|\mathbf{k}}|;\ i,j,\mathbf{k} \text{ with } \rho_{i,j|\mathbf{k}} \neq 0\} \geq c_n,\ c_n^{-1} = O(n^d),\\ &\qquad\qquad\qquad\qquad\qquad\qquad\text{for some } 0 < d < b/2,\\ &\sup_{n;i,j,\mathbf{k}} |\rho_{i,j|\mathbf{k}}| \leq M < 1,\end{aligned}$$

where $0 < b \leq 1$ is as in (A3).

Assumption (A1) is an often used assumption in graphical modeling, although it does restrict the class of possible probability distributions (see also third paragraph of Section 2.1); (A2) allows for an arbitrary polynomial growth of dimension as a function of sample size, that is, high-dimensionality; (A3) is a sparseness assumption and (A4) is a regularity condition. Assumptions (A3) and (A4) are rather minimal: note that with $b = 1$ in (A3), for example fixed $q_n = q < \infty$, the partial correlations can decay as $n^{-1/2+\varepsilon}$ for any $0 < \varepsilon \leq 1/2$. If the dimension p is fixed (with fixed DAG G and fixed distribution P), (A2) and (A3) hold and (A1) and the second part of (A4) remain as the only conditions. Recently, for undirected graphs the Lasso has been proposed as a computationally efficient algorithm for estimating high-dimensional conditional independence graphs where the growth in dimensionality is as in (A2) (see Meinshausen and Bühlmann, 2006). However, the Lasso approach can be inconsistent, even with fixed dimension p, as discussed in detail in Zhao and Yu (2006).

Theorem 1 *Assume (A1)-(A4). Denote by $\hat{G}_{skel,n}(\alpha_n)$ the estimate from the (first part of the) PC-algorithm in Section 2.2.2 and by $G_{skel,n}$ the true skeleton from the DAG G_n. Then, there exists $\alpha_n \to 0$ $(n \to \infty)$, see below, such that*

$$\begin{aligned}&\mathbb{P}[\hat{G}_{skel,n}(\alpha_n) = G_{skel,n}]\\ =\ & 1 - O(\exp(-Cn^{1-2d})) \to 1\ (n \to \infty) \textit{ for some } 0 < C < \infty,\end{aligned}$$

where $d > 0$ is as in (A4).

A proof is given in the Section A. A choice for the value of the significance level is $\alpha_n = 2(1 - \Phi(n^{1/2}c_n/2))$ which depends on the unknown lower bound of partial correlations in (A4).

3.2 Extending the Skeleton to the Equivalence Class

As mentioned before, all inference is done while finding the skeleton. If this part is completed perfectly, that is, if there was no error while testing conditional independencies (it is not enough to assume that the skeleton was estimated correctly), the second part will never fail (see Meek, 1995b). Therefore, we easily obtain:

Theorem 2 *Assume (A1)-(A4). Denote by $\hat{G}_{CPDAG}(\alpha_n)$ the estimate from the entire PC-algorithm in Section 2.2.2 and 2.3 and by G_{CPDAG} the true CPDAG from the DAG G. Then, there exists $\alpha_n \to 0$ $(n \to \infty)$, see below, such that*

$$\begin{aligned}&\mathbb{P}[\hat{G}_{CPDAG}(\alpha_n) = G_{CPDAG}]\\ =\ & 1 - O(\exp(-Cn^{1-2d})) \to 1\ (n \to \infty) \textit{ for some } 0 < C < \infty,\end{aligned}$$

where $d > 0$ is as in (A4).

A proof, consisting of one short argument, is given in the Section A. As for Theorem 1, we can choose $\alpha_n = 2(1-\Phi(n^{1/2}c_n/2))$.

By inspecting the proofs of Theorem 1 and Theorem 2, one can derive explicit error bounds for the error probabilities. Roughly speaking, this bounding function is the product of a linearly increasing and an exponentially decreasing term (in n). The bound is loose but for completeness, we present it in the Appendix.

4. Numerical Examples

We analyze the PC-algorithm for finding the skeleton and the CPDAG using various simulated data sets. The numerical results have been obtained using the `R`-package `pcalg`. For an extensive numerical comparison study of different algorithms, we refer to Tsamardinos et al. (2006).

4.1 Simulating Data

In this section, we analyze the PC-algorithm for the skeleton using simulated data. In order to simulate data, we first construct an adjacency matrix A as follows:

1. Fix an ordering of the variables.

2. Fill the adjacency matrix A with zeros.

3. Replace every matrix entry in the lower triangle (below the diagonal) by independent realizations of Bernoulli(s) random variables with success probability s where $0 < s < 1$. We will call s the sparseness of the model.

4. Replace each entry with a 1 in the adjacency matrix by independent realizations of a Uniform($[0.1,1]$) random variable.

This then yields a matrix A whose entries are zero or in the range $[0.1,1]$. The corresponding DAG draws a directed edge from node i to node j if $i < j$ and $A_{ji} \neq 0$. The DAGs (and skeletons thereof) that are created in this way have the following property: $\mathbb{E}[N_i] = s(p-1)$, where N_i is the number of neighbors of a node i.

Thus, a low sparseness parameter s implies few neighbors and vice-versa. The matrix A will be used to generate the data as follows. The value of the random variable $X^{(1)}$, corresponding to the first node, is given by

$$\begin{aligned}
&\varepsilon^{(1)} \sim N(0,1),\\
&X^{(1)} = \varepsilon^{(1)},
\end{aligned}$$

and the values of the next random variables (corresponding to the next nodes) can be computed recursively as

$$\begin{aligned}
&\varepsilon^{(i)} \sim N(0,1),\\
&X^{(i)} = \sum_{k=1}^{i-1} A_{ik} X^{(k)} + \varepsilon^{(i)} \; (i = 2,\dots,p),
\end{aligned}$$

where all $\varepsilon^{(1)},\dots,\varepsilon^{(p)}$ are independent.

4.2 Choice of Significance Level

In Section 3 we provided a value of the significance level $\alpha_n = 2(1-\Phi(n^{1/2}c_n/2))$. Unfortunately, this value is not constructive, since it depends on the unknown lower bound of partial correlations in (A4). To get a feeling for good values of the significance level in the domain of realistic parameter settings, we fitted a wide range of parameter settings and compared the quality of fit for different significance levels.

Assessing the quality of fit is not quite straightforward, since one has to examine simultaneously both the true positive rate (TPR) and false positive rate (FPR) for a meaningful comparison. We follow an approach suggested by Tsamardinos et al. (2006) and use the Structural Hamming Distance (SHD). Roughly speaking, this counts the number of edge insertions, deletions and flips in order to transfer the estimated CPDAG into the correct CPDAG. Thus, a large SHD indicates a poor fit, while a small SHD indicates a good fit.

We fitted 40 replicates to all combinations of

- $\alpha \in \{0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$,
- $p \in \{7, 15, 40, 70, 100\}$,
- $n \in \{30, 100, 300, 1000, 3000, 10000, 30000\}$,
- $E[N] \in \{2, 5\}$,

(where $E[N]$ is the average neighborhood size) and evaluated the SHD. Each value of α was used 40 times on each of the 70 possible parameter settings, and we then computed the average SHD over the 70 parameter settings.

The result is shown in Figure 1. One can see that the average SHD achieves a minimum in the region around $\alpha = 0.005$ and $\alpha = 0.01$. For higher or lower significance levels, the average SHD increases; the increase for bigger significance levels is much more pronounced. We analyzed the results of the simulation (see Figure 1) using pairwise Wilcoxon-Tests and Bonferroni correction. It turns out that $\alpha = 0.005$ and $\alpha = 0.01$ yield significantly lower average SHD than the other values of α. In contrast, there is no significant difference between $\alpha = 0.005$ and $\alpha = 0.001$ (without Bonferroni correction). Of course, if n was of different order of magnitude, a reasonable α should be a function of n with $\alpha = \alpha_n \to 0$ $(n \to \infty)$.

4.3 Performance for Different Parameter Settings

In this section, we give an overview over the performance in terms of the true positive rate (TPR) and false positive rate (FPR) for the skeleton and the SHD for the CPDAG. In order to keep the overview at a manageable size, we restrict the significance level to $\alpha = 0.01$. This was one of the two settings minimizing the average SHD as described in the previous section. The remaining parameters will be chosen as follows:

- $p \in \{7, 40, 100\}$,
- $n \in \{30, 100, 300, 1000, 3000, 10000, 30000\}$,
- $E[N] \in \{2, 5\}$.

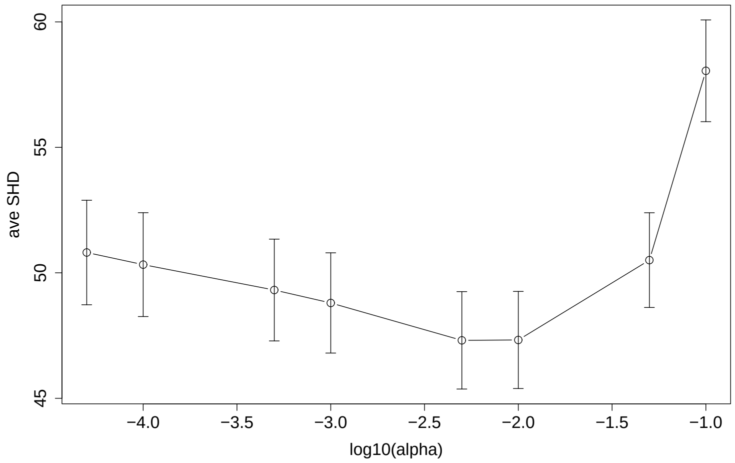


Figure 1: Average Structural Hamming Distance (ave SHD) with 95% confidence intervals. For each value of α, the average SHD was averaged over 70 parameter settings using 40 replicates each. One can see that the average SHD is minimized for significance levels between $\alpha = 0.005$ and $\alpha = 0.01$.

The overview is given in Figure 2. As expected, the fit for a dense graph (triangles; $E[N] = 5$) is worse than the fit for a sparse graph (circles; $E[N] = 2$). While the TPR and the SHD show a clear tendency with increasing sample size, the behavior of FPR is not so clear. The latter seems surprising at first sight but is due to the fact that we used the same $\alpha = 0.01$ for all n.

4.4 Properties in High-Dimensional Setting

In this section, we study the behaviour of the error rates in a high-dimensional setting. The number of variables increases exponentially, the number of samples increases linearly and the expected neighborhood size increases sub-linearly. By inspecting the theory, we would expect the error rates to stay constant or even decrease. Table 1 shows the parameter setting of a small numerical study addressing this question. Note that p increases exponentially, n increases linearly and the expected neighborhood size $E[N] = 0.2\sqrt{n}$ increases sub-linearly. We used $\alpha = 0.05$ and the results are based on 20 simulation runs.

Figure 3 shows boxplots of the TPR and the FPR over 20 replicates of this study. One can easily see that the TPR increases and the FPR decreases with sample size, although the number of $p = p_n$ grows fast and $E[N]$ grows slowly with n. This confirms our theory very clearly.

We should note, that while the number of neighbors to a given variable may be growing almost as fast as n, so that the number of neighbors is increasing with sample size, the percentage of true among all possible edges is going down with n. So in one sense, the sparsity in terms of percentage of true edges of the DAGs is decreasing, and in another sense the sparsity in terms of the neighborhood size is increasing with n.

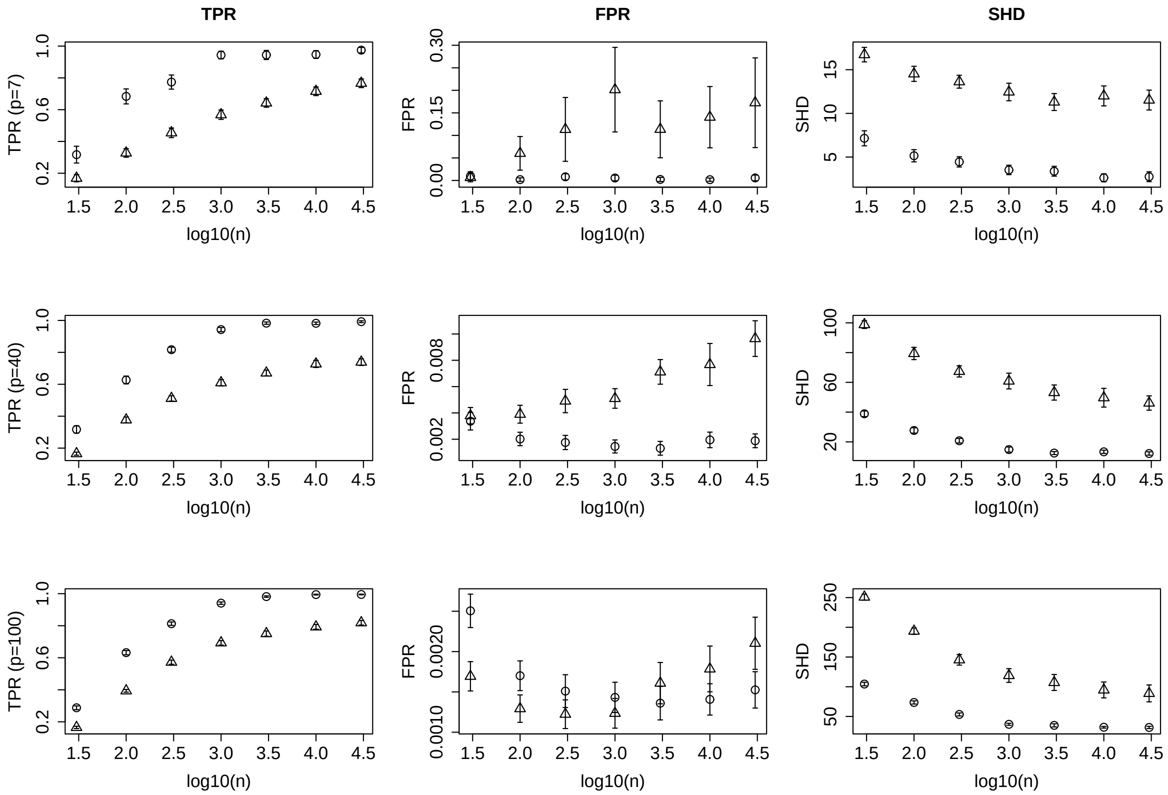


Figure 2: Performance of the PC-algorithm for different parameter settings, showing the mean of TPR, FPR and SHD together with 95% confidence intervals. The triangles represent parameter settings where $E[N] = 5$, while the circles represent parameter settings where $E[N] = 2$.

p	n	$E[N]$	TPR	FPR
9	50	1.4	0.61 (0.03)	0.023 (0.005)
27	100	2.0	0.70 (0.02)	0.011 (0.001)
81	150	2.4	0.753 (0.007)	0.0065 (0.0003)
243	200	2.8	0.774 (0.004)	0.0040 (0.0001)
729	250	3.2	0.794 (0.004)	0.0022 (0.00004)
2187	300	3.5	0.805 (0.002)	0.0012 (0.00002)

Table 1: The number of variables p increases exponentially, the sample size n increases linearly and the expected neighborhood size $E[N]$ increases sub-linearly. As supported by theory, the TPR increases and the FPR decreases in this setting. The results are based on using $\alpha = 0.05$, 20 simulation runs, and standard deviations are given in brackets.

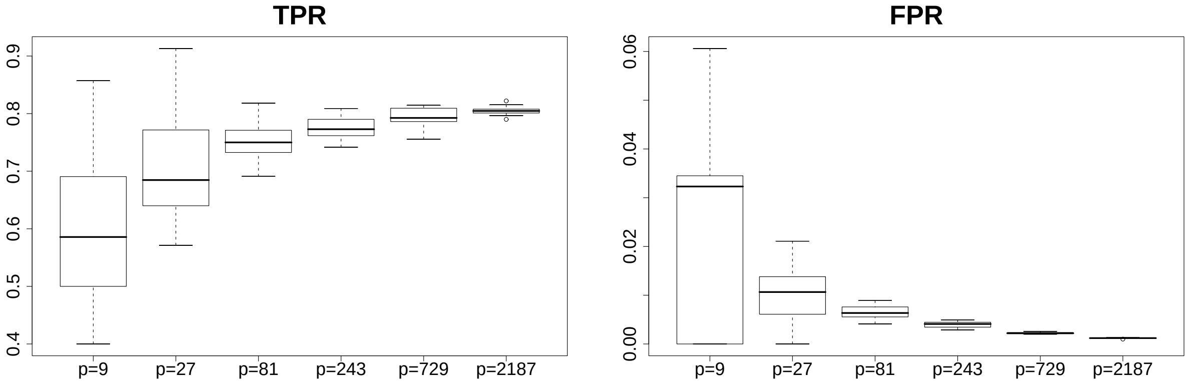


Figure 3: While the number of variables p increases exponentially, the sample size n increases linearly and the expected neighborhood size $E[N]$ increases sub-linearly, the TPR increases and the FPR decreases. See Table 1 for a more detailed specification of the parameters.

4.5 Computational Complexity

Our theoretical framework in Section 3 allows for large values of p. The computational complexity of the PC-algorithm is difficult to evaluate exactly, but the worst case is bounded by

$$O(p^{\hat{m}_{reach}}) \text{ which is with high probability bounded by } O(p^q) \qquad (4)$$

as a function of dimensionality p; here, q is the maximal size of the neighborhoods as described in assumption (A3) in Section 3. We note that the bound may be very loose for many distributions. Thus, for the worst case where the complexity bound is achieved, the algorithm is computationally feasible if q is small, say $q \leq 3$, even if p is large. For non-worst cases, however, we can still do the computations for much larger values of q and fairly dense graphs, for example some nodes j having neighborhoods of size up to $|adj(G, j)| = 30$.

We provide a small example of the processor time for estimating a CPDAG by using the PC-algorithm. The runtime analysis was done on an AMD Athlon 64 X2 Dual Core Processor 5000+ with 2.6 GHz and 4 GB RAM running on Linux and using R 2.4.1. The number of variables varied between $p = 10$ and $p = 1000$ while the number of samples was fixed at $n = 1000$. The sparseness was either $E[N] = 2$ or $E[N] = 8$. For each parameter setting, 10 replicates were used. In each case, the significance level used in the PC-algorithm was $\alpha = 0.01$. The average processor time together with its standard deviation for estimating both the skeleton and the CPDAG is given in Table 2. Graphs of $p = 1000$ nodes and 8 neighbors on average could be estimated in about 25 minutes, while networks with up to $p = 100$ nodes could be estimated in about a second. The additional time spent for finding the CPDAG from the skeleton is comparable for both neighborhood sizes and varies between a couple to almost 100 percent of the time needed to estimate the skeleton. The percentage tends to decrease with increasing number of variables.

Figure 4 gives a graphical impression of the results of this example. The sparse graphs (solid line with circles) were estimated faster than the dense graphs. While the line for the dense graph is very straight, the line for the sparse graphs has a positive curvature. Note, that this is a log-log plot; therefore, the slope of the lines indicates the exponent of polynomial growth. In this case, both curves follow very roughly a line with slope two indicating quadratic growth. The positive curvature of the solid line would indicate exponential growth; theory tells us, that this is not possible. One possible explanation for the positive curvature is the fact, that with increasing p, the maximal neighborhood size (which was not controlled in the simulation) is likely to increase. This would gradually increase the exponent in the polynomial growth of the upper bound in Formula (4), thus yielding a positive curvature.

p	$E[N]$	$\hat{G}_{skel}$	$\hat{G}_{CPDAG}$
10	2	0.037 (0.004)	0.072 (0.005)
10	8	0.093 (0.005)	0.124 (0.006)
30	2	0.15 (0.02)	0.23 (0.02)
30	8	0.84 (0.05)	0.93 (0.05)
50	2	0.33 (0.01)	0.48 (0.02)
50	8	2.2 (0.06)	2.4 (0.06)
100	2	1.03 (0.05)	1.49 (0.05)
100	8	8.9 (0.3)	9.4 (0.27)
300	2	8.3 (0.1)	13.8 (0.13)
300	8	89 (3)	95 (3)
1000	2	116 (0.5)	262 (0.8)
1000	8	1300 (60)	1445 (59)

Table 2: The average processor time (Athl. 64, 2.6 GHz, 4 GB) for estimating the skeleton ($\hat{G}_{skel}$) or the CPDAG ($\hat{G}_{CPDAG}$) for different DAGs in seconds, with standard errors in brackets. We used $\alpha = 0.01$ and sample size $n = 1000$.

5. R-Package `pcalg`

The R-package `pcalg` can be used to estimate from data the underlying skeleton or equivalence class of a DAG. To use this package, the statistics software `R` needs to be installed. Both `R` and the R-package `pcalg` are available free of charge at `http://www.r-project.org`. For low-dimensional problems (but not for p in the hundreds or thousands), there are a number of other implementations of the PC-algorithm that are also worth mentioning:

- Hugin at `http://www.hugin.com` .
- Murphy's Bayes Network toolbox at `http://bnt.sourceforge.net` .
- Tetrad IV at `http://www.phil.cmu.edu/projects/tetrad` .

In the following, we show an example of how to generate a random DAG, draw samples and infer from data the skeleton and the equivalence class of the original DAG using the R-package `pcalg`. The line width of the edges in the resulting skeleton and CPDAG can be adjusted to correspond to the reliability of the estimated dependencies. (The line width is proportional to the smallest value of $\sqrt{n-|\mathbf{k}|-3}\, Z(i,j,\mathbf{k})$ causing an edge, see also 3. Therefore, thick lines are reliable).

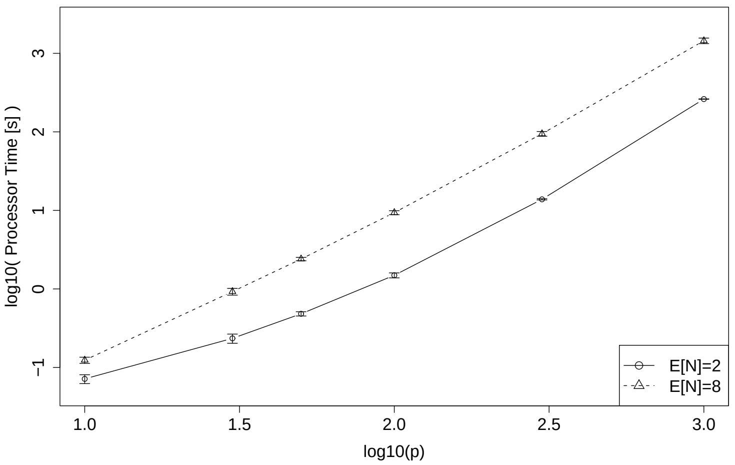


Figure 4: Average processor time over 10 runs together with 95% confidence intervals. Triangles correspond to dense ($E[N] = 8$), circles to sparse ($E[N] = 2$) underlying DAGs. We used $\alpha = 0.01$ and sample size $n = 1000$.

```
library(pcalg)
## define parameters
p <- 10 # number of random variables
n <- 10000 # number of samples
s <- 0.4 # sparsness of the graph
```

For simulating data as described in Section 4.1:

```
## generate random data
set.seed(42)
g <- randomDAG(p,s) # generate a random DAG
d <- rmvDAG(n,g) # generate random samples
```

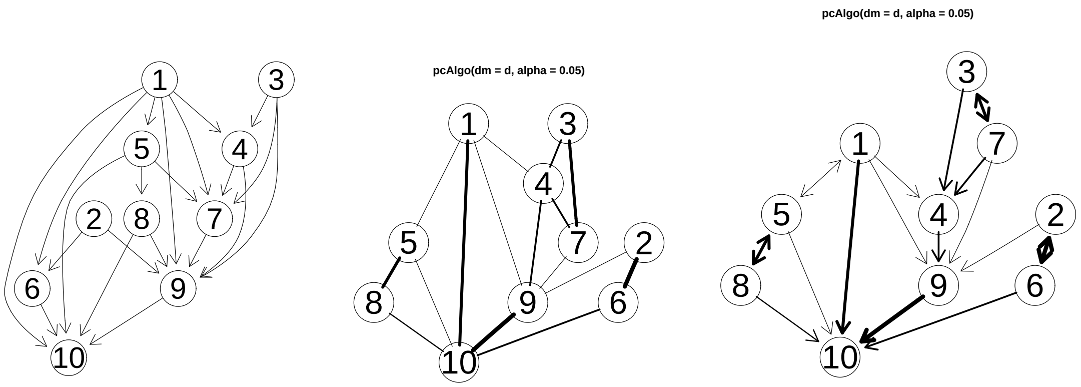
Then we estimate the underlying skeleton by using the function `pcAlgo` and extend the skeleton to the CPDAG by using the function `udag2cpdag`.

```
gSkel <-
  pcAlgo(d,alpha=0.05) # estimate of the skeleton
gCPDAG <-
  udag2cpdag(gSkel)
```

The CPDAG can also be estimated directly using

```
gCPDAG <-
  pcAlgo(d,alpha=0.05, directed=TRUE) # estimate of the CPDAG
```

The results can be easily plotted using the following commands:



(a) True DAG

(b) Estimated Skeleton

(c) Estimated CPDAG

Figure 5: Plots generated using the R-package pcalg as described in Section 5. (a) The true DAG. (b) The estimated skeleton using the R-function pcAlgo with $\alpha = 0.05$ and $n = 10000$. Line width encodes the reliability (z-values) of the dependence estimates (thick lines are reliable). (c) The estimated CPDAG using the R-function udag2cpdag. Double-headed arrows indicate undirected edges.

```
plot(g)
plot(gSkel,zvalue.lwd=TRUE)
plot(gCPDAG,zvalue.lwd=TRUE)
```

The original DAG is shown in Figure 5(a). The estimated skeleton and the estimated CPDAG are shown in Figure 5(b) and Figure 5(c), respectively. Note the differing line width, which indicates the reliability (z-values as in Formula 3) of the involved statistical tests (thick lines are reliable).

6. Conclusions

We show that the PC-algorithm is asymptotically consistent for the equivalence class of the DAG (represented by the CPDAG) and its skeleton with corresponding very high-dimensional, sparse Gaussian distribution. Moreover, the PC-algorithm is computationally feasible for such high-dimensional, sparse problems. Putting these two facts together, the PC-algorithm is established as a method (so far the only one) which is computationally feasible and provably correct, in the sense of uniform consistency, for high-dimensional DAGs. Sparsity, in terms of the maximal size of the neighborhoods of the true underlying DAG, is crucial for statistical consistency (assumption (A3) and Theorems 1 and 2) and for computational feasibility with at most a polynomial complexity (see Formula 4) as a function of dimensionality.

We emphasize that the skeleton of a DAG oftentimes provides interesting insights, and in a high-dimensional setting it is quite sensible to use the undirected skeleton as a simpler but more realistic target rather than the entire CPDAG. Software for the PC-algorithm is available as explained in Section 5.

Acknowledgments

We would like to thank Martin Mächler for his help with developing the R-package pcalg. We also thank three anonymous reviewers and the editor for their constructive comments. Markus Kalisch was supported by the Swiss National Science Foundation (grant no. 200021-105276 and 200020-113270/1).

Appendix A. Proofs

In the following, we give the proofs of all our theorems.

A.1 Proof of Proposition 1

Consider $\mathbf{X}$ with distribution P. Since P is faithful to the DAG G, conditional independence of $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ given $\{\mathbf{X}^{(r)};\ r \in \mathbf{k}\}$ ($\mathbf{k} \subseteq V \setminus \{i, j\}$) is equivalent to d-separation of nodes i and j given the set $\mathbf{k}$ (see Spirtes et al., 2000, Th. 3.3). Thus, the population PC$_{pop}$-algorithm as formulated in Section 2.2.1 coincides with the one from Spirtes et al. (2000) which is using the concept of d-separation, and the first claim about correctness of the skeleton follows from Spirtes et al. (2000, Th. 5.1., Ch. 13).

The second claim about the value of m_{reach} can be proved as follows. First, due to the definition of the PC$_{pop}$-algorithm and the fact that it constructs the correct skeleton, $m_{reach} \leq q$. We now argue that $m_{reach} \geq q-1$. Suppose the contrary. Then, $m_{reach} \leq q-2$: we could then continue with a further iteration in the algorithm since $m_{reach}+1 \leq q-1$ and there is at least one node j with neighborhood-size $|adj(G, j)| = q$: that is, the reached stopping level would be at least $q-1$ which is a contradiction to $m_{reach} \leq q-2$. □

A.2 Analysis of the PC-Algorithm

When analyzing the consistency of the PC-algorithm, the convergence properties of partial correlations turn out to be crucial. Therefore, we first concentrate on partial correlations and then shift our focus to the analysis of the PC-algorithm.

A.2.1 ANALYSIS OF PARTIAL CORRELATIONS

We first establish uniform consistency of estimated partial correlations. Denote by $\hat{\rho}_{i,j}$ and $\rho_{i,j}$ the sample and population correlation between $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$. Likewise, $\hat{\rho}_{i,j|\mathbf{k}}$ and $\rho_{i,j|\mathbf{k}}$ denote the sample and population partial correlation between $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ given $\{\mathbf{X}^{(r)}; r \in \mathbf{k}\}$, where $\mathbf{k} \subseteq \{1, \ldots, p_n\} \setminus \{i, j\}$.

Many partial correlations (and non-partial correlations) are tested for being zero during the run of the PC(m_n)-algorithm. For a fixed ordered pair of nodes i, j, the conditioning sets are elements of

$$K_{i,j}^{m_n} = \{\mathbf{k} \subseteq \{1, \ldots, p_n\} \setminus \{i, j\} : |\mathbf{k}| \leq m_n\}$$

whose cardinality is bounded by

$$|K_{i,j}^{m_n}| \leq B p_n^{m_n} \text{ for some } 0 < B < \infty. \tag{5}$$

Lemma 1 *Assume (A1) (without requiring faithfulness) and* $\sup_{n,i\neq j}|\rho_{n;i,j}|\leq M<1$ *(compare with (A4)). Then, for any* $0<\gamma\leq 2$*,*

$$\sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}}\mathbb{P}[|\hat{\rho}_{n;i,j}-\rho_{n;i,j}|>\gamma]\leq C_1(n-2)\exp\left((n-4)\log(\frac{4-\gamma^2}{4+\gamma^2})\right),$$

for some constant $0<C_1<\infty$ *depending on M only.*

Proof: We make substantial use of Hotelling (1953)'s work. Denote by $f_n(\hat{\rho},\rho)$ the probability density function of the sample correlation $\hat{\rho}=\hat{\rho}_{n+1;i,j}$ based on $n+1$ observations and by $\rho=\rho_{n+1;i,j}$ the population correlation. (It is notationally easier to work with sample size $n+1$; and we just use the abbreviated notations with $\hat{\rho}$ and ρ). For $0<\gamma\leq 2$,

$$\mathbb{P}[|\hat{\rho}-\rho|>\gamma]=\mathbb{P}[\hat{\rho}<\rho-\gamma]+\mathbb{P}[\hat{\rho}>\rho+\gamma].$$

It can be shown, that $f_n(r,\rho)=f_n(-r,-\rho)$, see Hotelling (1953, p.201). This symmetry implies,

$$\mathbb{P}_\rho[\hat{\rho}<\rho-\gamma]=\mathbb{P}_{\tilde{\rho}}[\hat{\rho}>\tilde{\rho}+\gamma]\text{ with }\tilde{\rho}=-\rho. \tag{6}$$

Thus, it suffices to show that $\mathbb{P}[\hat{\rho}>\rho+\gamma]=\mathbb{P}_\rho[\hat{\rho}>\rho+\gamma]$ decays exponentially in n, uniformly for all ρ.

It has been shown (Hotelling, 1953, p.201, Formula (29)), that for $-1<\rho<1$,

$$\mathbb{P}[\hat{\rho}>\rho+\gamma]\leq\frac{(n-1)\Gamma(n)}{\sqrt{2\pi}\Gamma(n+\frac{1}{2})}M_0(\rho+\gamma)(1+\frac{2}{1-|\rho|}) \tag{7}$$

with

$$\begin{aligned}M_0(\rho+\gamma)&=\int_{\rho+\gamma}^1(1-\rho^2)^{\frac{n}{2}}(1-x^2)^{\frac{n-3}{2}}(1-\rho x)^{-n+\frac{1}{2}}dx\\
&=\int_{\rho+\gamma}^1(1-\rho^2)^{\frac{\tilde{n}+3}{2}}(1-x^2)^{\frac{\tilde{n}}{2}}(1-\rho x)^{-\tilde{n}-\frac{5}{2}}dx\ \ (\text{using }\tilde{n}=n-3)\\
&\leq\frac{(1-\rho^2)^{\frac{3}{2}}}{(1-|\rho|)^{\frac{5}{2}}}\int_{\rho+\gamma}^1(\frac{\sqrt{1-\rho^2}\sqrt{1-x^2}}{1-\rho x})^{\tilde{n}}dx\\
&\leq\frac{(1-\rho^2)^{\frac{3}{2}}}{(1-|\rho|)^{\frac{5}{2}}}2\max_{\rho+\gamma\leq x\leq 1}(\frac{\sqrt{1-\rho^2}\sqrt{1-x^2}}{1-\rho x})^{\tilde{n}}.\end{aligned} \tag{8}$$

We will show now that $g_\rho(x)=\frac{\sqrt{1-\rho^2}\sqrt{1-x^2}}{1-\rho x}<1$ for all $\rho+\gamma\leq x\leq 1$ and $-1<\rho<1$ (in fact, $\rho\leq 1-\gamma$ due to the first restriction). Consider

$$\begin{aligned}\sup_{-1<\rho<1;\rho+\gamma\leq x\leq 1}g_\rho(x)&=\sup_{-1<\rho\leq 1-\gamma}\frac{\sqrt{1-\rho^2}\sqrt{1-(\rho+\gamma)^2}}{1-\rho(\rho+\gamma)}\\
&=\frac{\sqrt{1-\frac{\gamma^2}{4}}\sqrt{1-\frac{\gamma^2}{4}}}{1-(\frac{-\gamma}{2})(\frac{\gamma}{2})}=\frac{4-\gamma^2}{4+\gamma^2}<1\text{ for all }0<\gamma\leq 2.\end{aligned} \tag{9}$$

Therefore, for $-1 < -M \leq \rho \leq M < 1$ (see assumption (A4)) and using Formulas (7)-(9) together with the fact that $\frac{\Gamma(n)}{\Gamma(n+\frac{1}{2})} \leq const.$ with respect to n, we have

$$\begin{aligned}
& \mathbb{P}[\hat{\rho} > \rho + \gamma] \\
\leq \; & \frac{(n-1)\Gamma(n)}{\sqrt{2\pi}\Gamma(n+\frac{1}{2})} \frac{(1-\rho^2)^{\frac{3}{2}}}{(1-|\rho|)^{\frac{5}{2}}} 2(\frac{4-\gamma^2}{4+\gamma^2})^{\tilde{n}}(1+\frac{2}{1-|\rho|}) \\
\leq \; & \frac{(n-1)\Gamma(n)}{\sqrt{2\pi}\Gamma(n+\frac{1}{2})} \frac{1}{(1-M)^{\frac{5}{2}}} 2(\frac{4-\gamma^2}{4+\gamma^2})^{\tilde{n}}(1+\frac{2}{1-M}) \leq \\
\leq \; & C_1(n-1)(\frac{4-\gamma^2}{4+\gamma^2})^{\tilde{n}} = C_1(n-1)\exp((n-3)\log(\frac{4-\gamma^2}{4+\gamma^2})),
\end{aligned}$$

where $0 < C_1 < \infty$ depends on M only, but not on ρ or γ. By invoking Formula (6), the proof is complete (note that the proof assumed sample size $n+1$). □

Lemma 1 can be easily extended to partial correlations, as shown by Fisher (1924), using projections for Gaussian distributions.

Lemma 2 *(Fisher, 1924)*
Assume (A1) (without requiring faithfulness). If the cumulative distribution function of $\hat{\rho}_{n;i,j}$ is denoted by $F(\cdot|n,\rho_{n;i,j})$, then the cdf of the sample partial correlation $\hat{\rho}_{n;i,j|\mathbf{k}}$ with $|\mathbf{k}| = m < n-1$ is $F[\cdot|n-m,\rho_{n;i,j|\mathbf{k}}]$. That is, the effective sample size is reduced by m.

A proof can be found in Fisher (1924); see also Anderson (1984). □

Lemma 1 and 2 yield then the following.

Corollary 1 *Assume (the first part of) (A1) and (the upper bound in) (A4). Then, for any $\gamma > 0$,*

$$\begin{aligned}
& \sup_{i,j,\mathbf{k} \in K_{i,j}^{m_n}} \mathbb{P}[|\hat{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}| > \gamma] \\
\leq \; & C_1(n-2-m_n)\exp\left((n-4-m_n)\log(\frac{4-\gamma^2}{4+\gamma^2})\right),
\end{aligned}$$

for some constant $0 < C_1 < \infty$ depending on M from (A4) only.

The PC-algorithm is testing partial correlations after the z-transform $g(\rho) = 0.5\log((1+\rho)/(1-\rho))$. Denote by $Z_{n;i,j|\mathbf{k}} = g(\hat{\rho}_{n;i,j|\mathbf{k}})$ and by $z_{n;i,j|\mathbf{k}} = g(\rho_{n;i,j|\mathbf{k}})$.

Lemma 3 *Assume the conditions from Corollary 1. Then, for any $\gamma > 0$,*

$$\begin{aligned}
& \sup_{i,j,\mathbf{k} \in K_{i,j}^{m_n}} \mathbb{P}[|Z_{n;i,j|\mathbf{k}} - z_{n;i,j|\mathbf{k}}| > \gamma] \\
\leq \; & O(n-m_n)\left(\exp((n-4-m_n)\log(\frac{4-(\gamma/L)^2}{4+(\gamma/L)^2})) + \exp(-C_2(n-m_n))\right)
\end{aligned}$$

for some constant $0 < C_2 < \infty$ and $L = 1/(1-(1+M)^2/4)$.

Proof: A Taylor expansion of the z-transform $g(\rho) = 0.5\log((1+\rho)/(1-\rho))$ yields:

$$Z_{n;i,j|\mathbf{k}} - z_{n;i,j|\mathbf{k}} = g'(\tilde{\rho}_{n;i,j|\mathbf{k}})(\hat{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}), \qquad (10)$$

where $|\tilde{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}| \le |\hat{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}|$. Moreover, $g'(\rho) = 1/(1-\rho^2)$. By applying Corollary 1 with $\gamma = \kappa = (1-M)/2$ we have

$$\begin{aligned} &\sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[|\tilde{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}| \le \kappa] \\ > \;& 1 - C_1(n-2-m_n)\exp(-C_2(n-m_n)). \end{aligned} \qquad (11)$$

Since

$$\begin{aligned} g'(\tilde{\rho}_{n;i,j|\mathbf{k}}) &= \frac{1}{1-\tilde{\rho}^2_{n;i,j|\mathbf{k}}} = \frac{1}{1-(\rho_{n;i,j|\mathbf{k}} + (\tilde{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}))^2} \\ \le & \frac{1}{1-(M+\kappa)^2} \text{ if } |\tilde{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}| \le \kappa, \end{aligned}$$

where we also invoke (the second part of) assumption (A4) for the last inequality. Therefore, since $\kappa = (1-M)/2$ yielding $1/(1-(M+\kappa)^2) = L$, and using Formula (11), we get

$$\begin{aligned} &\sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[|g'(\tilde{\rho}_{n;i,j|\mathbf{k}})| \le L] \\ \ge \;& 1 - C_1(n-2-m_n)\exp(-C_2(n-m_n)). \end{aligned} \qquad (12)$$

Since $|g'(\rho)| \ge 1$ for all ρ, we obtain with Formula (10):

$$\begin{aligned} &\sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[|Z_{n;i,j|\mathbf{k}} - z_{n;i,j|\mathbf{k}}| > \gamma] \qquad (13) \\ \le \;& \sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[|g'(\tilde{\rho}_{n;i,j|\mathbf{k}})| > L] + \sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[|\hat{\rho}_{n;i,j|\mathbf{k}} - \rho_{n;i,j|\mathbf{k}}| > \gamma/L]. \end{aligned}$$

Formula (13) follows from elementary probability calculations: for two random variables U, V with $|U| \ge 1$ ($|U|$ corresponding to $|g'(\tilde{\rho})|$ and $|V|$ to the difference $|\hat{\rho} - \rho|$),

$$\begin{aligned} \mathbb{P}[|UV| > \gamma] &= \mathbb{P}[|UV| > \gamma, |U| > L] + \mathbb{P}[|UV| > \gamma, 1 \le |U| \le L] \\ &\le \mathbb{P}[|U| > L] + \mathbb{P}[|V| > \gamma/L]. \end{aligned}$$

The statement then follows from Formulas (13), (12) and Corollary 1. □

A.2.2 PROOF OF THEOREM 1

For the analysis of the PC-algorithm, it is useful to consider a more general version as shown in Algorithm 3.

The PC-algorithm in Section 2.2.1 equals the PC$_{pop}$(m_{reach})-algorithm . There is the obvious sample version, the PC(m)-algorithm, and the PC-algorithm in Section 2.2.2 is then same as the PC($\hat{m}_{reach}$)-algorithm, where $\hat{m}_{reach}$ is the sample version of Formula (2).

The population version PC$_{pop}$(m_n)-algorithm when stopped at level $m_n = m_{reach,n}$ constructs the true skeleton according to Proposition 1. Moreover, the PC$_{pop}$(m)-algorithm remains to be correct when using $m \ge m_{reach,n}$. The following Lemma extends this result to the sample PC(m)-algorithm.

Algorithm 3 The PC$_{pop}(m)$-algorithm

```
INPUT: Stopping level $m$, Vertex Set $V$, Conditional Independence Information
OUTPUT: Estimated skeleton $C$, separation sets $S$ (only needed when directing the skeleton afterwards)
Form the complete undirected graph $\tilde{C}$ on the vertex set V.
$\ell = -1; \quad C = \tilde{C}$
repeat
  $\ell = \ell + 1$
  repeat
    Select a (new) ordered pair of nodes $i,j$ that are adjacent in $C$ such that $|adj(C,i) \setminus \{j\}| \geq \ell$
    repeat
      Choose (new) $\mathbf{k} \subseteq adj(C,i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$.
      if $i$ and $j$ are conditionally independent given $\mathbf{k}$ then
        Delete edge $i, j$
        Denote this new graph by $C$.
        Save $\mathbf{k}$ in $S(i,j)$ and $S(j,i)$
      end if
    until edge $i,j$ is deleted or all $\mathbf{k} \subseteq adj(C,i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$ have been chosen
  until all ordered pairs of adjacent variables $i$ and $j$ such that $|adj(C,i) \setminus \{j\}| \geq \ell$ and $\mathbf{k} \subseteq adj(C,i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$ have been tested for conditional independence
until $\ell = m$ or for each ordered pair of adjacent nodes $i,j$: $|adj(C,i) \setminus \{j\}| < \ell$.
```

Lemma 4 *Assume (A1), (A2), (A3) where* $0 < b \leq 1$ *and (A4) where* $0 < d < b/2$. *Denote by* $\hat{G}_{skel,n}(\alpha_n, m_n)$ *the estimate from the PC(m_n)-algorithm in Section 2.2.2 and by* $G_{skel,n}$ *the true skeleton from the DAG* G_n. *Moreover, denote by* $m_{reach,n}$ *the value described in (2). Then, for* $m_n \geq m_{reach,n}$, $m_n = O(n^{1-b})$ $(n \to \infty)$, *there exists* $\alpha_n \to 0$ $(n \to \infty)$ *such that*

$$\begin{aligned} & \mathbb{P}[\hat{G}_{skel,n}(\alpha_n, m_n) = G_{skel,n}] \\ = \ & 1 - O(\exp(-Cn^{1-2d})) \to 1 \ (n \to \infty) \textit{ for some } 0 < C < \infty. \end{aligned}$$

Proof: An error occurs in the sample PC-algorithm if there is a pair of nodes i, j and a conditioning set $\mathbf{k} \in K_{i,j}^{m_n}$ (although the algorithm is typically only going through a random subset of $K_{i,j}^{m_n}$) where an error event $E_{i,j|\mathbf{k}}$ occurs; $E_{i,j,\mathbf{k}}$ denotes that "an error occurred when testing partial correlation for zero at nodes i, j with conditioning set $\mathbf{k}$". Thus,

$$\begin{aligned} & \mathbb{P}[\text{an error occurs in the PC}(m_n)\text{-algorithm}] \\ \leq \ & P[\bigcup_{i,j,\mathbf{k} \in K_{ij}^{m_n}} E_{i,j|\mathbf{k}}] \leq O(p_n^{m_n+2}) \sup_{i,j,\mathbf{k} \in K_{ij}^{m_n}} \mathbb{P}[E_{i,j|\mathbf{k}}], \end{aligned} \tag{14}$$

using that the cardinality of the set $|\{i, j, \mathbf{k} \in K_{ij}^{m_n}\}| = O(p_n^{m_n+2})$, see also Formula 5. Now

$$E_{i,j|\mathbf{k}} = E^{I}_{i,j|\mathbf{k}} \cup E^{II}_{i,j|\mathbf{k}}, \tag{15}$$

where

$$\text{type I error } E^{I}_{i,j|\mathbf{k}}: \ \sqrt{n - |k| - 3}|Z_{i,j|\mathbf{k}}| > \Phi^{-1}(1 - \alpha/2) \text{ and } z_{i,j|\mathbf{k}} = 0,$$

$$\text{type II error } E^{II}_{i,j|\mathbf{k}}: \ \sqrt{n - |k| - 3}|Z_{i,j|\mathbf{k}}| \leq \Phi^{-1}(1 - \alpha/2) \text{ and } z_{i,j|\mathbf{k}} \neq 0.$$

Choose $\alpha = \alpha_n = 2(1-\Phi(n^{1/2}c_n/2))$, where c_n is from (A4). Then,

$$\begin{aligned}\sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[E^I_{i,j|\mathbf{k}}] &= \sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[|Z_{i,j|\mathbf{k}} - z_{i,j|\mathbf{k}}| > (n/(n-|\mathbf{k}|-3))^{1/2}c_n/2] \\ &\leq O(n-m_n)\exp(-C_3(n-m_n)c_n^2), \qquad (16)\end{aligned}$$

for some $0 < C_3 < \infty$ using Lemma 3 and the fact that $\log(\frac{4-\delta^2}{4+\delta^2}) \sim -\delta^2/2$ as $\delta \to 0$. Furthermore, with the choice of $\alpha = \alpha_n$ above,

$$\begin{aligned}\sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[E^{II}_{i,j|\mathbf{k}}] &= \sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[|Z_{i,j|\mathbf{k}}| \leq \sqrt{n/(n-|\mathbf{k}|-3)}c_n/2] \\ \leq \quad \sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} &\mathbb{P}[|Z_{i,j|\mathbf{k}} - z_{i,j|\mathbf{k}}| > c_n(1-\sqrt{n/(n-|\mathbf{k}|-3)}/2)],\end{aligned}$$

because $\inf_{i,j;\mathbf{k}\in K_{i,j}^{m_n}} |z_{i,j|\mathbf{k}}| \geq c_n$ since $|g(\rho)| \geq |\rho|$ for all ρ and using assumption (A4). By invoking Lemma 3 we then obtain:

$$\sup_{i,j,\mathbf{k}\in K_{i,j}^{m_n}} \mathbb{P}[E^{II}_{i,j|\mathbf{k}}] \leq O(n-m_n)\exp(-C_4(n-m_n)c_n^2) \qquad (17)$$

for some $0 < C_4 < \infty$. Now, by Formulas (14)-(17) we get

$$\begin{aligned}&\mathbb{P}[\text{an error occurs in the PC}(m_n)\text{-algorithm}] \\ \leq\;& O(p_n^{m_n+2}(n-m_m)\exp(-C_5(n-m_n)c_n^2)) \\ \leq\;& O(n^{a(m_n+2)+1}\exp(-C_5(n-m_n)n^{-2d})) \\ =\;& O\left(\exp\left(a(m_n+2)\log(n)+\log(n)-C_5(n^{1-2d}-m_n n^{-2d})\right)\right) = o(1),\end{aligned}$$

because n^{1-2d} dominates all other terms in the argument of the exp-function due to the assumption in (A4) that $d < b/2$. This completes the proof. □

Lemma 4 leaves some flexibility for choosing m_n. The PC-algorithm yields a data-dependent reached stopping level $\hat{m}_{reach,n}$, that is, the sample version of (2).

Lemma 5 *Assume (A1)-(A4). Then,*

$$\begin{aligned}&\mathbb{P}[\hat{m}_{reach,n} = m_{reach,n}] = 1 - O(\exp(-Cn^{1-2d})) \to 1\ (n\to\infty) \\ &\textit{for some } 0 < C < \infty,\end{aligned}$$

where $d > 0$ is as in (A4).

Proof: Consider the population algorithm $\text{PC}_{pop}(m)$: the reached stopping level satisfies $m_{reach} \in \{q_n - 1, q_n\}$, see Proposition 1. The sample PC(m_n)-algorithm with stopping level in the range of $m_{reach} \leq m_n = O(n^{1-b})$, coincides with the population version on a set A having probability $P[A] = 1 - O(\exp(-Cn^{1-2d}))$, see the last formula in the proof of Lemma 4. Hence, on the set A, $\hat{m}_{reach,n} = m_{reach} \in \{q_n - 1, q_n\}$. The claim then follows from Lemma 4. □

Lemma 4 and 5 together complete the proof of Theorem 1.

Because there are faithful distributions which require $m_n = m_{reach,n} \in \{q_n - 1, q_n\}$ for consistent estimation with the PC(m)-algorithm, Lemma 5 indicates that the PC-algorithm, stopping at $\hat{m}_{reach,n}$, yields with high probability the smallest $m = m_n$ which is universally consistent for all faithful distributions.

A.2.3 PROOF OF THEOREM 2

As mentioned in Section 2.3, due to the result of Meek (1995b), it is sufficient to estimate the correct skeleton and separation sets. The proof of Theorem 1 also covers the issue of choosing the correct separation sets S, that is, the probability of estimating the correct sets S goes to one as $n \to \infty$. Hence, the proof of Theorem 2 is completed.

Appendix B. Bound for error probability of PC-algorithm

M and c are the upper and lower bounds for partial correlations, as defined in Section 3.1. p is the number of variables, q is the maximal size of neighbors, n is the sample size. The significance level is chosen as suggested in the proofs, that is, $\alpha = 2(1-\Phi(n^{1/2}c/2))$. By closely inspecting the proofs, one can derive the following upper bound for the error probability of the PC-algorithm:

$$\mathbb{P}[\hat{G} \neq G] \leq p^{q+2} C_1 (n-1-q)(\exp(-C_2(n-q)) + \exp((n-4-q) f(L, \frac{c}{2})))$$

where $L = \frac{1}{1-(1+M)^2/4}$, $C_1 = \frac{1+2/(1-M)}{(1-M)^{5/2}}$, $C_2 = -\log(\frac{16-(1-M)^2}{16+(1-M)^2})$ and $f(x,y) = \log(\frac{4-(y/x)^2}{4+(y/x)^2})$.

References

T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2nd edition, 1984.

D.M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002a.

D.M. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002b.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

D. Edwards. *Introduction to Graphical Modelling*. Springer Verlag, 2nd edition edition, 2000.

R.A. Fisher. The distribution of the partial correlation coefficient. *Metron*, 3:329–332, 1924.

S.B. Gillispie and M.D. Perlman. Enumerating Markov equivalence classes of acyclic digraph models. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 171–177, 2001.

A. Goldenberg and A. Moore. Tractable learning of large bayes net structures from sparse data. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, pages 44–51. ACM Press, 2004.

D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

H. Hotelling. New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society Series B*, 15(2):193–232, 1953.

S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

C. Meek. Strong completeness and faithfulness in Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 411–418, 1995a.

C. Meek. Causal inference and causal explanation with background knowledge. In P.Besnard and S.Hanks, editors, *Uncertainty in Artificial Intelligence*, volume 11, pages 403–410, 1995b.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462, 2006.

R.E. Neapolitan. *Learning Bayesian Networks*. Pearson Prenctice Hall, 2004.

A. Y. Ng. On feature selection: Learning with exponentially many irrelevant features as training examples. In *Proc. 15th International Conf. on Machine Learning*, pages 404–412. Morgan Kaufmann, San Francisco, CA, 1998.

J. Pearl. *Causality*. Cambridge University Press, 2000.

J.M. Robins, R. Scheines, P. Sprites, and L. Wasserman. Uniform consistency in causal inference. *Biometrika,* 90:491–515, 2003.

R.W. Robinson. Counting labeled acyclic digraphs. In F. Haray, editor, *New Directions in the Theory of Graphs: Proc. of the Third Ann Arbor Conf. on Graph Theory (1971)*, pages 239–273. Academic Press, NY, 1973.

D.J. Spiegelhalter, A.P. Dawid, S.L. Lauritzen, and R.G. Cowell. Bayesian analysis in expert-systems (with discussion). *Statistical Science*, 8:219–283, 1993.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2nd edition, 2000.

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

T. Verma and J.Pearl. A theory of inferred causation. In J. Allen, R. Fikes, and E. Sandewall, editors, *Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, pages 441–452. Morgan Kaufmann, New York, 1991.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In M. Henrion, M. Shachter, R. Kanal, and J. Lemmer, editors, *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227, 1990.

J. Zhang and P. Spirtes. Strong faithfulness and uniform consistency in causal inference. In *UAI*, pages 632–639, 2003.

P. Zhao and B. Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

O. Zuk, S. Margel, and E. Domany. On the number of samples needed to learn the correct structure of a Bayesian network. In *UAI*, 2006.