Journal of Machine Learning Research ? (????) ?? Submitted 09/06; Published ??

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

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

Peter Bühlmann BUEHLMANN@STAT.MATH.ETHZ.CH
Seminar für Statistik
ETH Zürich
8092 Zürich, Switzerland

Editor: ???

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 for sparse problems with many nodes, i.e. 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 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 and compare it with other methods.

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). 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.

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, i.e. 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 allows to estimate those aspects of a sparse DAG even if p is in the hundreds or thousands. For the high-dimensional setting with $p > n$, sparsity of the underlying DAG is crucial for statistical consistency and computational feasibility. The PC-algorithm seems to be the only method for high-dimensional settings which is computationally feasible and, due to the new results in this paper, provably correct in an asymptotic sense.

We argue empirically that the PC-algorithm is rather insensitive to the choice of its single tuning parameter, a significance level for testing, and we compare the PC-algorithm with other methods, at least for low- or mid-dimensional problems.

2. Finding the Equivalence Class of a DAG

2.1 Definitions and Preliminaries

A graph $G = (V, E)$ consists of a set of nodes or vertices $V = \{1, \ldots, p\}$ and a set of edges $E \subseteq V \times V$, i.e. 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 $i \to j$. If both $(i, j) \in E$ and $(j, i) \in E$, the edge is called undirected; we then use the notation $i - j$. An directed acyclic graph (DAG) is a graph G where all edges are directed and not containing any cycle.

If there is a directed edge $i \to j$, 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 $\mathbf{s} \subseteq V$,

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

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 Spirtes et al. (2000, Th. 3.2).

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 $i \to j$ and $k \to j$, 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), and we can only identify an equivalence class of DAGs, even when having infinitely many observations. Using a result from Verma and Pearl (1991), 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 DAG 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 PDAG and (2) for every undirected edge $i-j$ there exists a DAG with $i \to j$ and a DAG with $i \leftarrow j$ in the equivalence class. While several PDAGs might represent the same equivalence class, it was shown in Chickering (2002b) that two CPDAGs are identical if and only if they represent the same equivalence class. Therefore, we prefer CPDAGs over PDAGs for representing equivalence classes.

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,

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

(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. I.e., 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 is quite sensible to use the undirected skeleton as a target rather than the CPDAG.

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 or numerical simulations 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)), i.e. all partial correlations in the case of multivariate normal distributions as first suggested by Verma and J.Pearl. 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

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.

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 $q = \max_{1 \le j \le p} |adj(G, j)|$. 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 7.

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, i.e. the conditional independence relations 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.

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.
$\ell = -1; \quad C = \tilde{C}$
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 for each ordered pair of adjacent nodes i,j: $|adj(C, i) \setminus \{j\}| < \ell$.

Proposition 2 *Assume that the distribution P of the random vector $\mathbf{X}$ is multivariate normal. For $i \neq j \in \{1, \ldots, p\}$, $\mathbf{k} \subseteq \{1, \ldots, p\} \setminus \{i, j\}$, 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, cf. 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, i.e. the significance level for testing partial correlations. The algorithm seems to be rather insensitive to the choice of α, see Section 4.

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 PDAG belonging to the equivalence class of the underlying DAG. (Spirtes et al. (2000) provide an alternative, which we think is harder to implement.).

Algorithm 2 Extending the skeleton to a PDAG

INPUT: Skeleton G_{skel}, separation sets S
OUTPUT: PDAG 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.

The output of Algorithm 2 is a PDAG. To transform it into a CPDAG, we first transform it to a DAG (see Dor and Tarsi, 1992) and then to a CPDAG (see Chickering, 2002b). Comparing with the task of finding the skeleton in Algorithm 1, the computational complexity of these steps is negligible.

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

$$O(p^{\hat{m}_{reach,n}}) \text{ which is with high probability bounded by } O(p^{q_n}) \tag{4}$$

as a function of dimensionality p; here, q_n 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_n is small, say $q_n \leq 3$, even if p is large. For non-worst cases, however, we can still do the computations for much larger values of q_n and fairly dense graphs, e.g. some nodes j have neighborhoods of size up to $|adj(G,j)| = 30$.

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 $0 \le a < \infty$.

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

(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:

$$\inf\{|\rho_{i,j|\mathbf{k}}|;\ i, j, \mathbf{k} \text{ with } \rho_{i,j|\mathbf{k}} \neq 0\} \ge c_n,\ c_n^{-1} = O(n^d), \text{ for some } 0 < d < b/2,$$

$$\sup_{n;i,j,\mathbf{k}} |\rho_{i,j|\mathbf{k}}| \le M < 1,$$

where $0 < b \le 1$ 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, i.e. 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), e.g. fixed $q_n = q < \infty$ the partial correlations can decay as $n^{-1/2+\varepsilon}$ for any $0 < \varepsilon \le 1/2$. 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, the Lasso has been proposed as a computationally efficient algorithm for estimating high-dimensional undirected 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*

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

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

A proof is given in the Section 7. 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, the second part will never fail (see Pearl, 2000). Furthermore, the extension of a PDAG to a DAG and from a DAG to a CPDAG were shown to be correct in Dor and Tarsi (1992) and Chickering (2002b), respectively. Therefore, we easily obtain:

Theorem 2 *Assume (A1)-(A4). Denote by* $\hat{G}_{CPDAG}(\alpha_n)$ *the estimate from the entire PC-algorithm 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*

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

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

A proof, consisting of one short argument, is given in the Section 7. As for Theorem 2, we can choose $\alpha_n = 2(1 - \Phi(n^{1/2}c_n/2))$.

4. Numerical Examples

We analyze the PC-algorithm and alternative methods for finding the skeleton and the CPDAG using various simulated data sets. Again, we will first deal with the skeleton and then with the CPDAG. The numerical results have been obtained using the `R`-package `pcalg` and the `Bayes Net Toolbox` of Kevin Murphy in `MATLAB`.

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

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

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

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

where all $\epsilon^{(1)}, \ldots, \epsilon^{(p)}$ are independent.

4.2 Skeleton

4.2.1 Comparison with Alternative Methods

In this section, we will compare the PC-algorithm with two alternative methods, Greedy Equivalent Search (GES, see Chickering, 2002a) and Maximum Weight Spanning Trees (MWST, see Heckerman et al., 1995) which both try to find DAGs that maximize the BIC criterion. We report here the accuracy for finding the true skeleton: The rate of correctly identified edges, True Positive Rate (TPR), the rate of estimated edges which are false, False Positive Rate (FPR), and as a measure of reliability the ratio of correctly found edges and the total number of all found edges, the True Discovery Rate (TDR).

We found, that the BIC based methods find DAGs with high TPR but also rather high FPR. If only a small amount of observations is available (as is often the case in a very high-dimensional setting), we cannot hope to recover the complete underlying model. Therefore, instead of large TPR, we would rather prefer a high TDR. A measure for high reliability is the True Discovery Rate (TDR), which is the ratio of correctly found edges and the total number of all edges found.

Method	ave$[TPR]$	ave$[FPR]$	ave$[TDR]$
PC	0.57 (0.06)	0.02 (0.01)	0.91 (0.05)
GES	0.85 (0.05)	0.13 (0.04)	0.71 (0.07)
MWST	0.66 (0.07)	0.06 (0.01)	0.78 (0.06)

Table 1: $p = 10$ nodes, sample size $n = 50$, sparseness $s = 0.1$, 50 replicates. Standard errors are given in parentheses. The PC-algorithm achieves a substantially higher True Discovery Rate than GES or MWST.

As can be seen in Table 1, the PC-algorithm achieves in our simulations by far higher True Discovery Rates than GES or MWST: of all found edges, 91% were correct. Thus, although a smaller total of edges was found, the estimated edges were correct more frequently. We think, that this is a substantial advantage for real world applications.

4.2.2 Different Parameter Settings

As introduced in Section 2.2.2, the PC-algorithm has only one tuning parameter α. In this section, we analyze the dependence of the algorithm on this parameter for different settings.

α	ave$[TPR]$	ave$[FPR]$	ave$[TDR]$	ave$[\hat{m}_{reach}]$
0.001	0.065 (0.002)	0.0057 (0.0005)	0.80 (0.02)	2.56 (0.07)
0.01	0.089 (0.003)	0.0082 (0.0007)	0.78 (0.02)	2.92 (0.06)
0.05	0.116 (0.003)	0.0133 (0.0009)	0.75 (0.02)	3.26 (0.06)
0.1	0.128 (0.003)	0.0161 (0.0010)	0.73 (0.02)	3.46 (0.08)
0.3	0.151 (0.005)	0.0238 (0.0011)	0.68 (0.02)	4.28 (0.08)

Table 2: $p = 30$, $n = 20$, $s = 0.1$, 50 replicates; s.e. in parentheses.

Tables 2 to 7 show the average over 50 replicates of TPR, FPR, TDR and $\hat{m}_{reach}$ for the DAG model in Section 4.1 with $p = 30$ nodes and varying sample size n and sparseness s.

In the wide range of αs, no choice can be identified as being the best or worst. Especially in the case of very few observations we see that small α leads to the discovery of very few

α	ave$[TPR]$	ave$[FPR]$	ave$[TDR]$	ave$[\hat{m}_{reach}]$
0.001	0.069 (0.002)	0.0056 (0.0005)	0.80 (0.02)	2.30 (0.07)
0.01	0.092 (0.002)	0.0097 (0.0007)	0.77 (0.02)	2.92 (0.06)
0.05	0.116 (0.003)	0.0141 (0.0008)	0.73 (0.01)	3.28 (0.07)
0.1	0.131 (0.003)	0.0165 (0.0008)	0.73 (0.01)	3.50 (0.08)
0.3	0.159 (0.004)	0.0233 (0.0010)	0.70 (0.01)	4.34 (0.07)

Table 3: $p = 30$, $n = 20$, $s = 0.4$, 50 replicates; s.e. in parentheses.

α	ave$[TPR]$	ave$[FPR]$	ave$[TDR]$	ave$[\hat{m}_{reach}]$
0.001	0.153 (0.004)	0.015 (0.001)	0.77 (0.01)	4.02 (0.07)
0.01	0.175 (0.005)	0.017 (0.001)	0.77 (0.01)	4.38 (0.09)
0.05	0.193 (0.005)	0.020 (0.001)	0.76 (0.01)	4.82 (0.08)
0.1	0.200 (0.005)	0.021 (0.001)	0.76 (0.01)	5.00 (0.09)
0.3	0.221 (0.006)	0.025 (0.001)	0.74 (0.01)	5.66 (0.09)

Table 4: $p = 30$, $n = 100$, $s = 0.1$, 50 replicates; s.e. in parentheses.

α	ave$[TPR]$	ave$[FPR]$	ave$[TDR]$	ave$[\hat{m}_{reach}]$
0.001	0.155 (0.004)	0.015 (0.001)	0.78 (0.01)	4.12 (0.08)
0.01	0.174 (0.004)	0.016 (0.001)	0.78 (0.01)	4.54 (0.08)
0.05	0.188 (0.005)	0.020 (0.001)	0.76 (0.01)	4.78 (0.09)
0.1	0.196 (0.005)	0.021 (0.001)	0.76 (0.01)	4.92 (0.09)
0.3	0.217 (0.006)	0.028 (0.001)	0.71 (0.01)	5.58 (0.10)

Table 5: $p = 30$, $n = 100$, $s = 0.4$, 50 replicates; s.e. in parentheses.

α	ave$[TPR]$	ave$[FPR]$	ave$[TDR]$	ave$[\hat{m}_{reach}]$
0.001	0.250 (0.007)	0.033 (0.001)	0.71 (0.01)	6.5 (0.1)
0.01	0.258 (0.007)	0.036 (0.001)	0.70 (0.01)	6.7 (0.1)
0.05	0.264 (0.007)	0.038 (0.001)	0.69 (0.01)	7.0 (0.1)
0.1	0.268 (0.007)	0.041 (0.001)	0.68 (0.01)	7.3 (0.1)
0.3	0.283 (0.007)	0.047 (0.001)	0.67 (0.01)	7.6 (0.1)

Table 6: $p = 30$, $n = 5000$, $s = 0.1$, 50 replicates; s.e. in parentheses.

α	ave$[TPR]$	ave$[FPR]$	ave$[TDR]$	ave$[\hat{m}_{reach}]$
0.001	0.260 (0.007)	0.031 (0.001)	0.73 (0.01)	6.40 (0.09)
0.01	0.268 (0.007)	0.035 (0.001)	0.72 (0.01)	6.80 (0.09)
0.05	0.277 (0.006)	0.036 (0.001)	0.72 (0.01)	7.04 (0.09)
0.1	0.281 (0.007)	0.038 (0.001)	0.71 (0.01)	7.22 (0.10)
0.3	0.294 (0.006)	0.045 (0.001)	0.68 (0.01)	7.70 (0.11)

Table 7: $p = 30$, $n = 5000$, $s = 0.4$, 50 replicates; s.e. in parentheses.

edges with high reliability (high TDR), whereas higher values of α lead to the discovery of more edges but with less reliability. Therefore, α can be used for fine tuning in finding a good compromise between amount of edges found and their reliability.

Note, however, that especially for larger sample sizes, the rates vary only little, sometimes only by a few percent. Comparing this with the large change in α (over two orders of magnitude), we feel that the PC-algorithm is rather insensitive to the choice of its single tuning parameter.

4.3 Equivalence Class

In this section, we will compare the abilities of the PC-algorithm and GES to find CPDAGs. To this end, we simulated 100 different DAGs consisting of $p = 8$ nodes and a sparseness of $s = 0.2$. Then, a sample of size n was drawn from each DAG and both the PC-algorithm and GES were used to recover the underlying CPDAG. Then, the percentage of correctly found entire CPDAGs out of 100 was recorded. This is a rather stringent criterion, since only entirely correctly identified CPDAGs are positively scored. In Figure 1 the percentages together with point-wise 95% confidence intervals are shown for several values of n. The performance of both algorithms improves as n increases and one can see that GES is a bit better for finding the true CPDAG.

Drawing a conclusion, we see that GES and the PC-algorithm complement each other very well: GES performs fine at the task of identifying the CPDAG in a low-dimensional setting with an abundance of data. On the other hand, if the dimension is medium or high (or the amount of available samples is small), there is not much hope of finding the true underlying CPDAG. The next best goal would be to identify parts of the underlying skeleton. Although the PC-algorithm finds fewer edges in the skeleton, the edges found are correct more frequently. Therefore, the PC-algorithm seems to be more suitable for this task.

4.4 Computational Costs

In the implementation we used for GES (`Bayes Net Toolbox` of Kevin Murphy in `MATLAB`), it has been computationally feasible to consider problems where the dimensionality p is about ten or less (see Table 1 and Figure 2). The PC-algorithm had in these problems a computational cost which was by a factor of up to several thousands smaller than that for GES. For larger dimensions in the range of dozens or hundreds, GES would not output an estimate within reasonable amount of time (e.g. within one day for a single data set), whereas the PC-algorithm needs a few seconds of CPU time only.

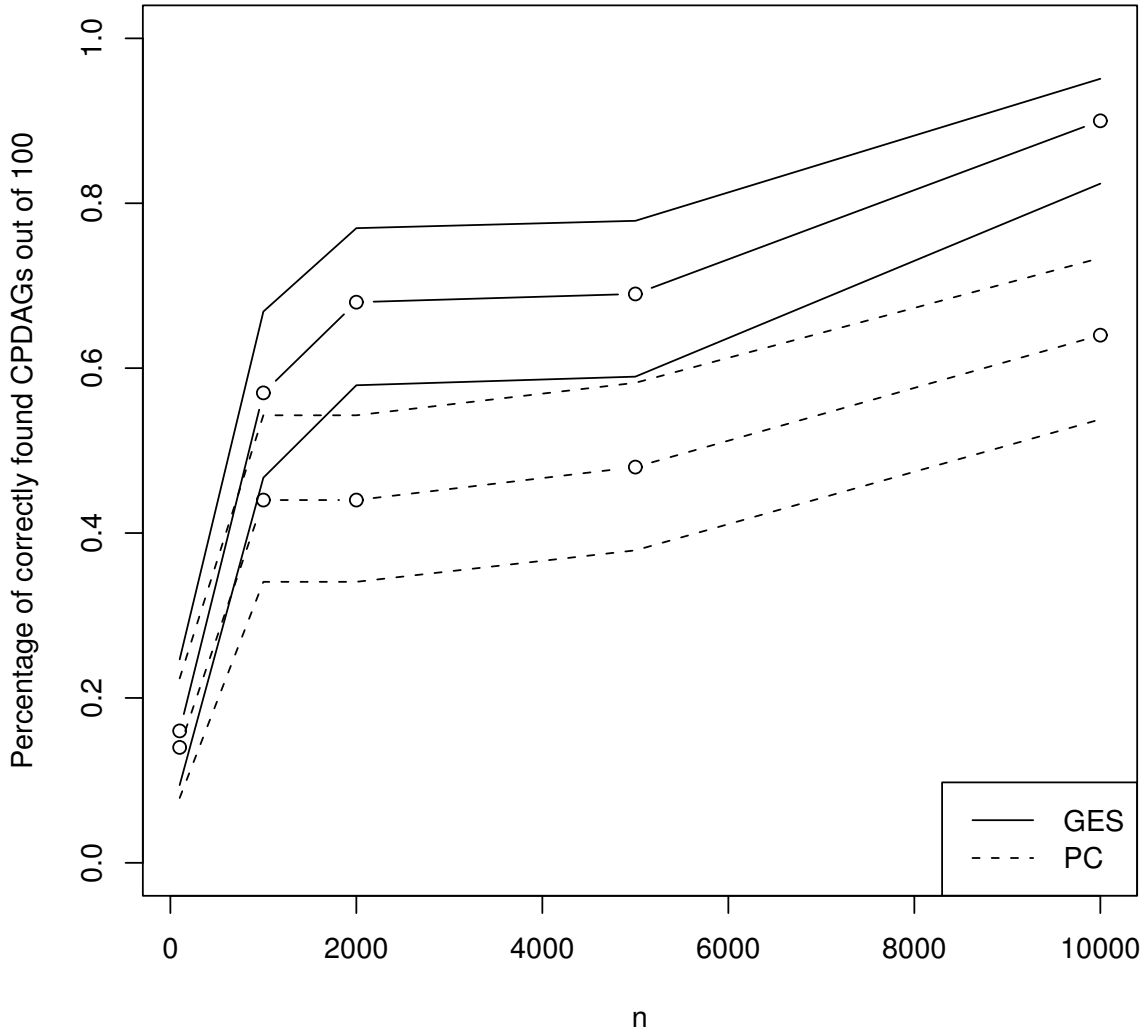


Figure 1: The percentage of correctly found CPDAGs for different sample sizes using the PC-algorithm and GES. The lines with circles show the mean values over 100 replications; the corresponding lines without circles describe point-wise 95% confidence intervals. While the performance increases for both algorithms, GES finds the true CPDAG more frequently ($p = 8$ nodes, sparseness $s = 0.2$)

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`.

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. 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).

```
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
```

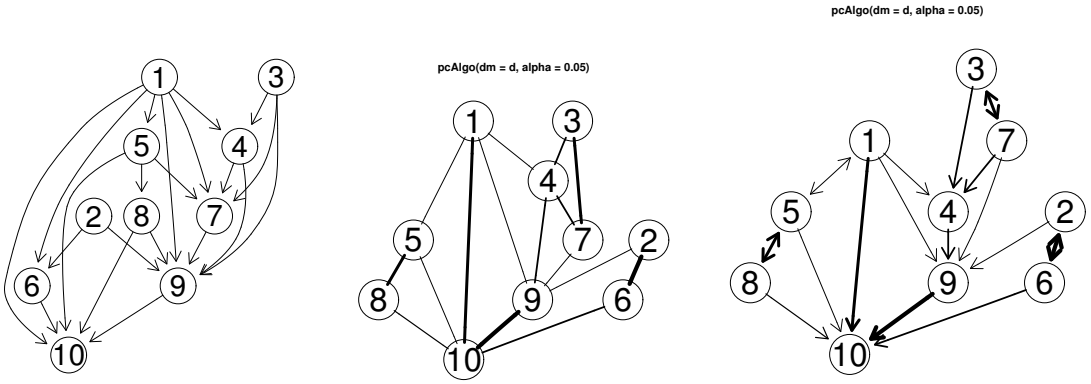
Then we estimate the underlying skeleton by using the function `pcAlgo` and extend the skeleton to the CPDAG by using the function `udag2cpdag`.

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

The results can be easily plotted using the following commands:

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

The original DAG is shown in Figure 2(a). The estimated skeleton and the estimated CPDAG are shown in Figure 2(b) and Figure 2(c), respectively. Note the differing line width, which indicates the reliability (z-values as in (3)) of the involved statistical tests (thick lines are reliable).



(a) True DAG (b) Estimated Skeleton (c) Estimated CPDAG

Figure 2: These plots were 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 $\alpha = 0.05$. 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.

6. Conclusions

We show that the PC-algorithm is asymptotically consistent for the equivalence class of the DAG (i.e. 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 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 (4)) as a function of dimensionality.

The PC-algorithm compares well with alternative approaches like MWST and GES for low- or mid-dimensional problems (where the alternative methods are computationally feasible). We found that the PC-algorithm has higher reliability (higher true discovery rate) for the skeleton than MWST and GES while the latter has slightly better rates for discovering the entire CPDAG. 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.

For high-dimensional settings, MWST and GES (with the implementations we used) become extremely slow while the PC-algorithm is still computationally feasible with a factor

of thousands faster than MWST or GES. Software for the PC-algorithm is available as explained in Section 5

7. Proofs

7.1 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 $m_{reach} \leq q-2$. □

7.2 Analysis of the PC-Algorithm

7.2.1 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 $f_n(r,\rho)$ 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

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

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 (7)-(9) together with the fact that $\frac{\Gamma(n)}{\Gamma(n+\frac{1}{2})} \leq const.$ with respect to n, we have

$$\mathbb{P}[\hat{\rho} > \rho + \gamma]$$

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

where $0 < C_1 < \infty$ depends on M only, but not on ρ or γ. By invoking (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:

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

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

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

Since

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

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 (11), we get

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

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

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

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

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

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

7.2.2 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.

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 **k then**

Delete edge i, j

Denote this new graph by C.

Save **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: $|adj(C,i) \setminus \{j\}| < \ell$.

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 (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 $m \geq m_{reach,n}$. The following Lemma extends this result to the sample PC(m)-algorithm.

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*

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

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

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

where

$$\begin{aligned} &\text{type I error } E_{i,j|\mathbf{k}}^{I}: \ \sqrt{n - |k| - 3}|Z_{i,j|\mathbf{k}}| > \Phi^{-1}(1 - \alpha/2) \text{ and } z_{i,j|\mathbf{k}} = 0, \\ &\text{type II error } E_{i,j|\mathbf{k}}^{II}: \ \sqrt{n - |k| - 3}|Z_{i,j|\mathbf{k}}| \leq \Phi^{-1}(1 - \alpha/2) \text{ and } z_{i,j|\mathbf{k}} \neq 0. \end{aligned}$$

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

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

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,

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

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

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

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

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

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 $\hat{m}_{reach,n}$, i.e. 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 $\text{PC}(m_n)$-algorithm with stopping level in the range of $m_{reach} \le m_n = O(n^{1-b})$, 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 $\text{PC}(m)$-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.

7.2.3 Proof of Theorem 2

The proof of Theorem 1 also covers the issue of choosing the correct separation sets S, i.e., the probability of having the correct set S goes to one as $n \to \infty$. Hence, the proof of Theorem 2 is completed.

References

T.W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2nd edition 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. Dor and M. Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. Technical report r-185, UCLA Computer Science Department, 1992.

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.

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.

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

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

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 edition, 2000.

T. Verma and J.Pearl. Equivalence and synthesis of causal models. In P. Bonissone, M. Henrion, L.N. Kanal, and J.F. Lemmer, editors, *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence*, volume 6 of *Uncertainty in Artificial Intelligence*, pages 255–68. Amserdam:Elsevier.

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, 1991.

P. Zhao and B. Yu. On model selection consistency of lasso. *To appear in Journal of Machine Learning Research*, 2006.