Causality
Learning Causal Structures

Maciej Liśkiewicz

University of Lübeck

January, 2023

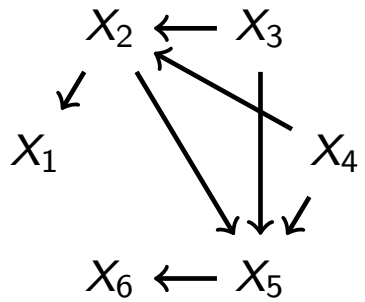
Learning Causal Structures

Agenda

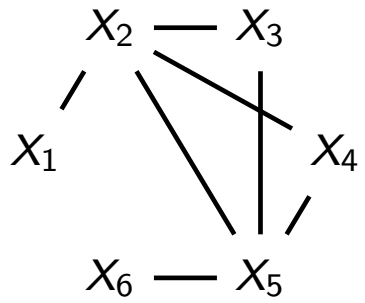
- Causal structural learning: constraint-based approaches
- General framework, Independence Tests
- Markov equivalence classes and CPDAGs
- Faithfulness
- PC Algorithm
- Causal structural learning: score-based methods

Markov equivalence classes and CPDAGs

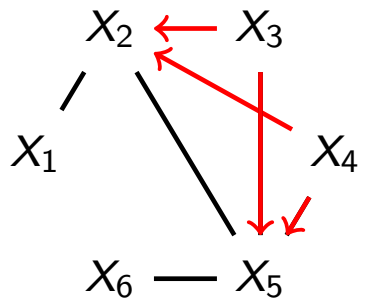
- Recall, for a given DAG G



- The skeleton of G is the undirected graph where every edge in G is substituted by an undirected edge



- An inverted fork $A \to C \leftarrow B$ is called a v-structure if A and B are not adjacent in G



Markov equivalence classes and CPDAGs

- Recall that two DAGs G and G' over $\mathbf{V}$ are *Markov equivalent* if $\mathcal{I}(G) = \mathcal{I}(G')$, where

$$\mathcal{I}(G) = \{(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_G : \text{ for all } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V}\}$$

- **Fact:** Having only access to an oracle which answers d-separation queries of the form

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$$

with yes if and only if $\mathbf{X}$ and $\mathbf{Y}$ are d-separatat in an unknown DAG G, it is impossible to learn uniquely DAG G over $\mathbf{V}$, if the corresponding Markov equivalence class contains more than one DAG

- Due to Verma and Pearl we know that G and G' are Markov equivalent if and only if G and G' have the same skeleton and the same set of v-structures.
- For example

$$G_1 : X \to Z \to Y \qquad G_2 : X \leftarrow Z \leftarrow Y \qquad G_3 : X \leftarrow Z \to Y$$

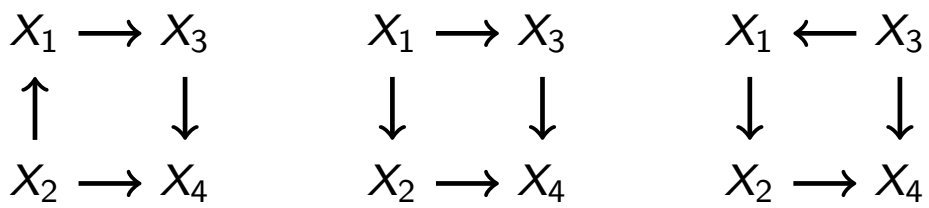
are Markov equivalent but

$$G_1 : X \to Z \to Y \qquad G_4 : X \to Z \leftarrow Y$$

are not

Markov equivalence classes and CPDAGs

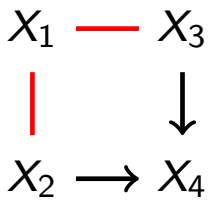
- The set of all DAGs over **V** is partitioned into a set of mutually exclusive and exhaustive *Markov equivalent classes*, which are the set of equivalence classes induced by the Markov equivalence relation
- **Question:** How to represent the classes?
- For example, how to represent uniquely and in a compact way all Markov equivalent DAGs:



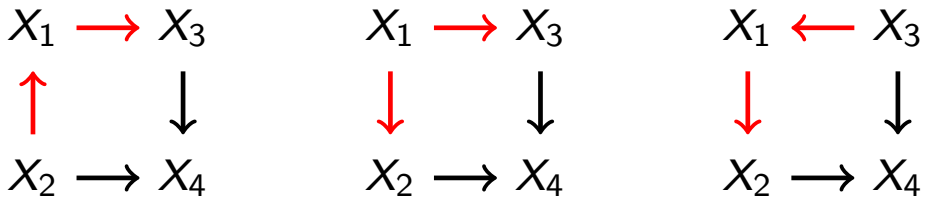
- This is a very important question, since learning a causal structure from data / CI statements we want to find a representation of *all* Markov equivalent DAGs
- We show that a Markov equivalence class can be described uniquely by a *CPDAG* (completed partially directed acyclic graph)
- Thus, the goal of our causal structure learning algorithms is to find a CPDAG

Markov equivalence classes and CPDAGs

- Before giving a formal definition we show an example of a CPDAG:



- This CPDAG encodes three DAGs:



Markov equivalence classes and CPDAGs

Definition (CPDAG)

Given a DAG $G = (\mathbf{V}, \mathbf{E})$, the class of Markov equivalent graphs to G, denoted as $[G]$, is defined as

$$[G] = \{G' \mid G' \text{ is Markov equivalent to } G\}.$$

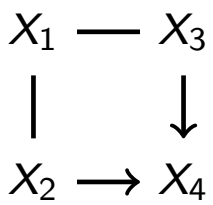
The (mixed) graph representing $[G]$ is called a *CPDAG* and is denoted as $G^* = (\mathbf{V}, \mathbf{E}^*)$, with the set of edges defined as follows:

- $A \to B$ is in $\mathbf{E}^*$ if $A \to B$ belongs to every $G' \in [G]$ and
- $A - B$ is in $\mathbf{E}^*$ if there exist $G', G'' \in [G]$ so that

  $A \to B$ is an edge of G' and
  $A \leftarrow B$ is an edge of G''.

A partially directed graph D is called a CPDAG if $D = G^*$ for some DAG G.

Exercise: It is easy to check, that our example graph is a CPDAG



Faithfulness

- A given distribution P over $\mathbf{V}$ is in general compatible with a variety of structures
- To identify a structure G based on i.i.d. (independent and identically distributed) sample from distribution $P(X_1, \ldots, X_n)$ the following must hold

Definition (Faithfulness)

P over $\mathbf{V}$ is said to be *faithful* to a DAG $G = (\mathbf{V}, \mathbf{E})$ if and only if for all subsets $\mathbf{A}, \mathbf{B}, \mathbf{S}$

$$(\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{S})_P \quad \text{if and only if} \quad (\mathbf{A} \perp\!\!\!\perp \mathbf{B} \mid \mathbf{S})_G$$

- Most distributions are faithful
- More precisely, for DAGs it holds that the non-faithful distributions form a Lebesgue null-set in parameter space associated with a DAG
- **Remark:** In this lecture we consider distributions that are faithful to a DAG

Faithfulness and CPDAGs

- If the distribution P is *Markovian* and *faithful* with respect to the underlying DAG G, we have a one-to-one correspondence between d-separation statements in G and the corresponding CI statements in P
- All graphs outside $[G]$ can therefore be rejected because they impose a set of d-separations that does not equal the set of CIs in P
- Moreover from the Markov condition and faithfulness it follows, that we are not able to distinguish between two DAGs $G', G'' \in [G]$

Lemma (Identifiability of Markov equivalence class)

Assume that P is Markovian and faithful with respect to G.

- Then, for each graph $G' \in [G]$, we find an SCM (S, P) that entails the distribution P.
- Furthermore, there is no graph G'' with $G'' \notin [G]$, such that P is Markovian and faithful with respect to G''.

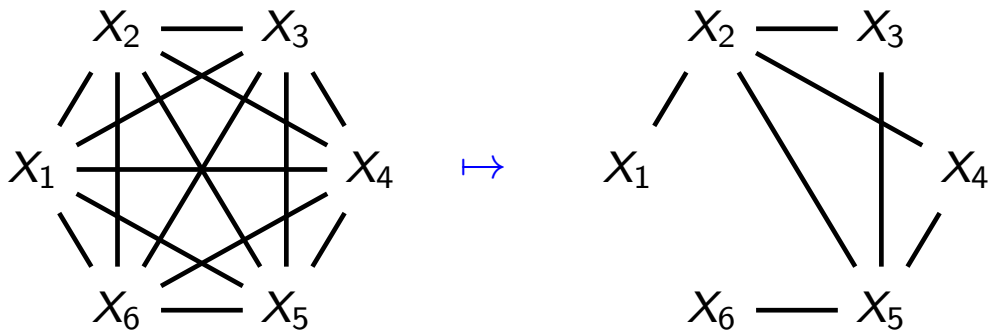
Causal Structural Learning: Constraint-based Methods

- Constraint-based methods (also called conditional independence based methods) assume that the distribution is Markovian and faithful with respect to the underlying graph
- The goal is to estimate the correct Markov equivalence class represented as a CPDAG
- In this lecture we will present PC algorithm (invented by Peter Spirtes and Clark Glymour, 2000)
- Another examples
  - The Inductive Causation (IC) algorithm (Verma and Pearl, 1990)
  - Fast Causal Inference (FCI) (Spirtes et al., 2000)
  - SGS (for the inventors Spirtes, Glymour, and Scheines) algorithm

Causal Structural Learning: PC Algorithm

Finding Skeleton

- Most constraint-based methods, including PC algorithm, first estimate the skeleton starting with a complete undirected graph (and orient as many edges as possible afterward)



- For the skeleton search, the following lemma is useful

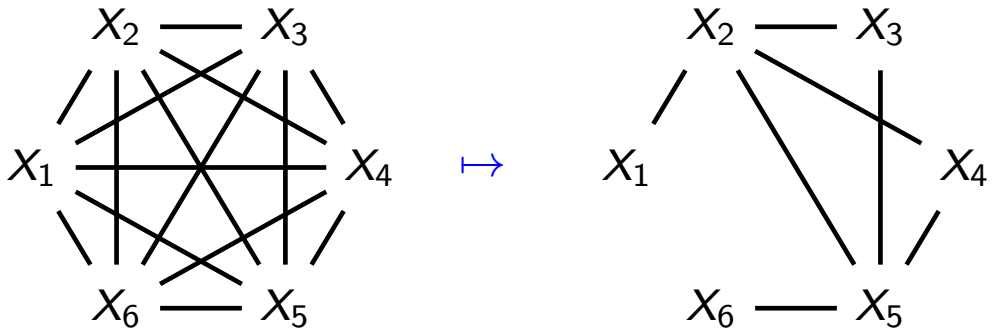
Lemma (Estimation of Skeleton)

The following two statements hold

- Two nodes X, Y in a DAG $G = (\mathbf{V} = \{X_1, \dots X_n\}, \mathbf{E})$ are adjacent if and only if they can not be d-separated by any subset $\mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\}$
- If two nodes X, Y in G are not adjacent, then they are d-separated by either $Pa(X)$ or $Pa(Y)$.

Causal Structural Learning: PC Algorithm

Finding Skeleton



- Thus if two nodes X and Y are adjacent in G then there is no set $\mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\}$ that d-separates X and Y
- If X and Y are *not* adjacent in G then either $Pa(X)$ d-separates X and Y or $Pa(Y)$ d-separates X and Y
- Due to the faithfulness assumption, these properties translate to independences
- If X and Y are *not* adjacent in the true DAG, then either $(X \perp\!\!\!\perp Y \mid Pa(X))$ or $(X \perp\!\!\!\perp Y \mid Pa(Y))$
- Conversely, if the variables are adjacent then any CI query

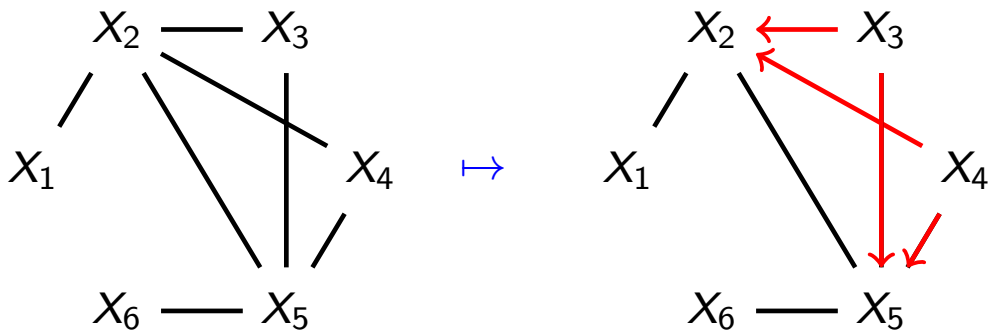
$$(X \perp\!\!\!\perp Y \mid \mathbf{S})$$

will be answered negatively

Causal Structural Learning: PC Algorithm

Orienting v-Structures

- Deleting an edge $X_i - X_j$ in constructing the skeleton, the algorithm stores the witness set $S(i,j) := \mathbf{S}$, for which the query
$$(X_i \perp\!\!\!\perp X_j \mid \mathbf{S})$$
has been answered positively
- Based on the sets $S(i,j)$, the algorithm orients the v-structures in the skeleton

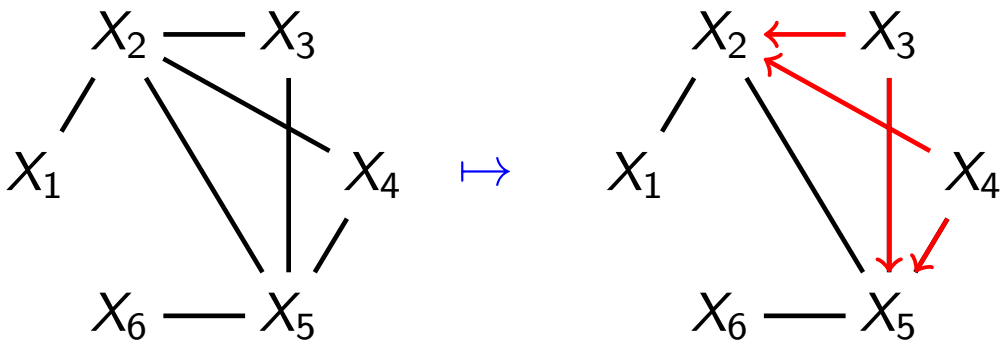


- The property, that
> *G and G' are Markov equivalent if and only if G and G' have the same skeleton and the same set of v-structures*

suggests how to orient the v-structures in the graph correctly

Causal Structural Learning: PC Algorithm

Orienting v-Structures



- If X_i and X_j are not directly connected in the obtained skeleton, set $S(i,j)$ d-separates the nodes
- If, as in the example above, the skeleton contains as induced graph the structure

$$X_4 - X_2 - X_3$$

then it can be oriented as

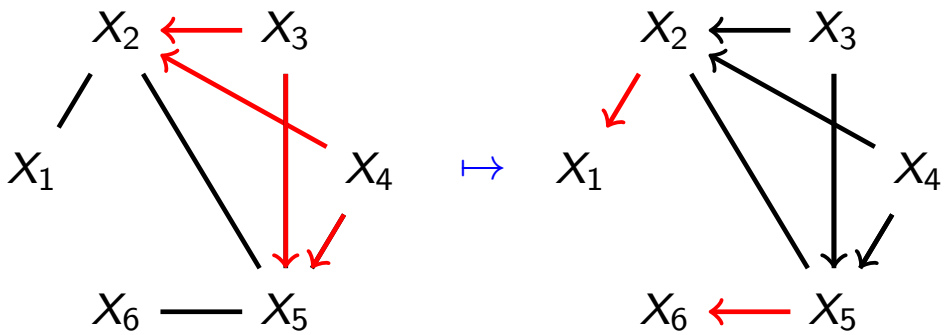
$$X_4 \to X_2 \leftarrow X_3$$

if and only if $X_2 \notin S(3,4)$

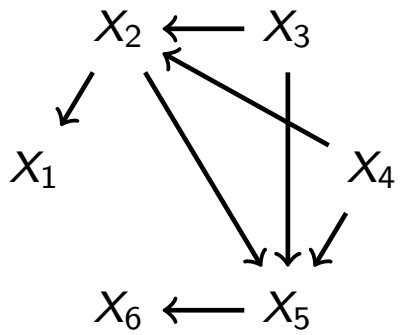
Causal Structural Learning: PC Algorithm

Propagate Orientations

- After the orientation of v-structures, the algorithm orients some further edges

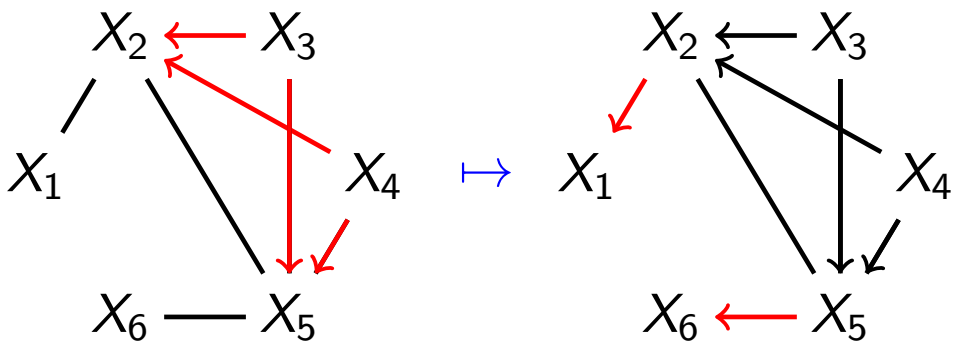


- To this this aim the algorithm uses *orientation rules* that has been shown to be complete and are known as Meek's orientation rules (that we will present next)
- Call a *pattern* of a DAG G the PDAG H such that it has the same skeleton as G and has an oriented edge $A \to B$ iff there is a vertex C, which is not adjacent to A, such that $C \to B$ is an edge in G, too.
- Thus, in the pattern H of G, the only directed edges are the ones which are part of a v-structure in G; E.g., the PDAG above (left) is a pattern of the DAG:



Causal Structural Learning: PC Algorithm

Propagate Orientations



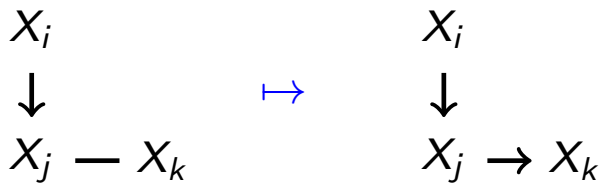
- Due to Meek (1995), we know that
  - when starting with a pattern H of some DAG G and
  - repeatedly executing the following three rules until none of them applies

  we obtain a CPDAG G^* representing the Markov equivalent DAGs

Causal Structural Learning: PC Algorithm

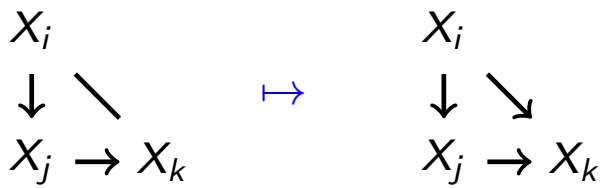
Propagate Orientations: Meek's Rules

R1: **Prohibited v-structures**



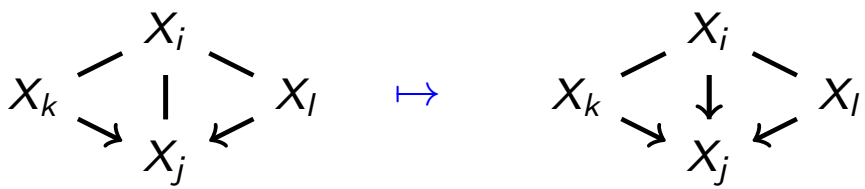
Orient $X_j - X_k$ into $X_j \to X_k$ whenever there is an arrow $X_i \to X_j$ s.t. X_i and X_k are nonadjacent

R2: **Acyclicity rule**



Orient $X_i - X_j$ into $X_i \to X_j$ whenever there is a chain $X_i \to X_k \to X_j$

R3: **Quartet rule**

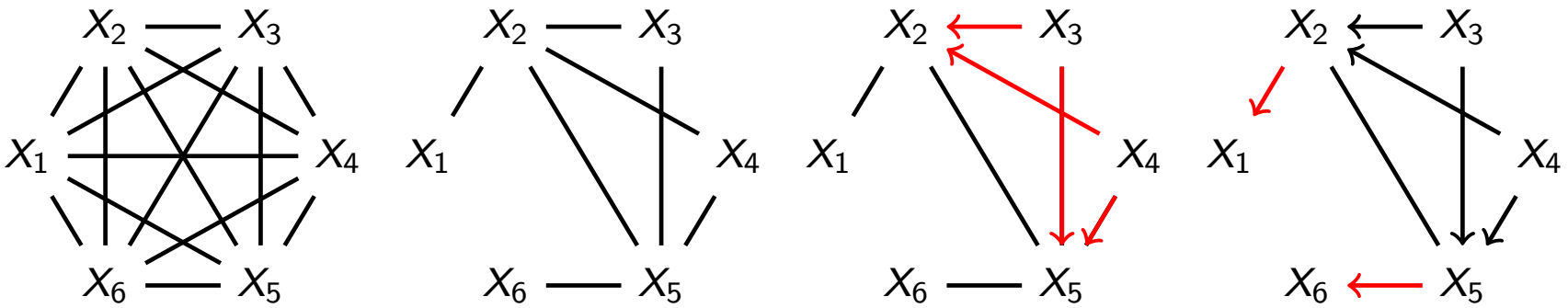


Orient $X_i - X_j$ into $X_i \to X_j$ if two chains $X_i - X_k \to X_j$ and $X_i - X_l \to X_j$ exist s.t. X_k and X_l are nonadjacent

Causal Structural Learning: PC Algorithm

The PC Algorithm
Input: Set of variables **V**, access to CI statements / dataset over **V** and significance level α
Output: A CPDAG

1. Initialize with the complete undirected graph on **V**
2. Using CI tests of order $l = 0, 1, 2, \ldots$, find the skeleton and separating sets of removed edges
3. Based the separating sets, orient v-structures in the skeleton
4. Propagate orientations of v-structures to as many remaining undirected edges as possible
5. Return the final graph



1. Complete graph | 2. Skeleton | 3. v-structures | 4. Propagation

PC Algorithm: Finding Skeleton

- In this oracle-version of the algorithm, we assume that perfect knowledge about all necessary conditional independence relations is available
- An independence oracle is a(n abstract) device that answers, in unit time, the query $(X_i \perp\!\!\!\perp X_j \mid \mathbf{Z})$

The PC-algorithm: Finding the Skeleton (Oracle-Version)

Input: Vertex Set $\mathbf{V}$, Access to CI statements
Output: Estimated skeleton G, separation sets S

Let G be the complete undirected graph on $\mathbf{V}$
Let $l = 0$
repeat
  repeat
    Take new adjacent $X_i, X_j \in \mathbf{V}$ s.t. $|N(X_i) \setminus \{X_j\}| \geq l$
    repeat
      Choose new $\mathbf{Z} \subseteq N(X_i) \setminus \{X_j\}$ with $|\mathbf{Z}| = l$
      if $(X_i \perp\!\!\!\perp X_j \mid \mathbf{Z})$ **then**
        Delete edge $X_i - X_j$ from G
        Save $\mathbf{Z}$ in $S(i,j)$ and $S(j,i)$
      end if
    until X_i and X_j are not adjacent or no new $\mathbf{Z} \subseteq N(X_i) \setminus \{X_j\}$, with $|\mathbf{Z}| = l$, exists
  until all pairs of adjacent X_i and X_j s.t. $|N(X_i) \setminus \{X_j\}| \geq l$ have been selected
  Set $l = l + 1$
until for each adjacent X_i, X_j: $|N(X_i) \setminus \{X_j\}| < l$
return G and S

PC Algorithm: Extending the Skeleton to a CPDAG

The PC-algorithm: Finding v-Structures and Orientation Propagation

Input: Skeleton G , separation sets S
Output: CPDAG G

Find v Structures:
for all pairs of nonadjacent variables X_i, X_j with common neighbour X_k **do**
    if $X_k \notin S(i,j)$ **then**
        Replace $X_i - X_k - X_j$ in G_{skel} by $X_i \rightarrow X_k \leftarrow X_j$
    end if
end for

Propagate Orientations:
repeat in G:
    R1: Orient $X_j - X_k$ into $X_j \rightarrow X_k$ whenever there is an arrow $X_i \rightarrow X_j$ s.t. X_i and X_k are nonadjacent
    R2: Orient $X_i - X_j$ into $X_i \rightarrow X_j$ whenever there is a chain $X_i \rightarrow X_k \rightarrow X_j$
    R3: Orient $X_i - X_j$ into $X_i \rightarrow X_j$ if two chains $X_i - X_k \rightarrow j$ and $X_i - X_l \rightarrow X_j$ exist s.t. X_k and X_l are nonadjacent
until no further rule can be applied.

PC Algorithm

Theorem

If P is faithful to a DAG G, the oracle version of PC algorithm finds a CPDAG D such that $D = G^*$, i.e. such that D represents the Markov equivalence class $[G]$. Moreover it uses at most

$$2\binom{n}{2}\sum_{i=0}^{d}\binom{n-1}{i} \leq \frac{n^{d+1}}{(d-1)!}$$

independence checks where d is the maximal degree of any vertex in G.

So worst case complexity is exponential, but algorithm fast for sparse graphs. Sampling properties are less well understood although consistency results exist.

PC Algorithm

Finding the Skeleton with Statistical CI Tests

- We have assumed that in the oracle version, PC has the has an access to an *independence-oracle* that answers specific conditional independence queries
- In practice, however, the algorithm – in line 8 of the "Finding the Skeleton" module – needs to infer the CI statements *from a finite amount of data*
- To this aim we need to implement an *oracle query* $(X_i \perp\!\!\!\perp X_j \mid \mathbf{Z})$ (in line 8) by a procedure for making the requisite *statistical decisions* about conditional independence

PC Algorithm

Finding the Skeleton with Statistical CI Tests

- In practice, we are given a finite sample

$$\mathcal{D} = ((x_i^1, x_j^1, \mathbf{z}^1), (x_i^2, x_j^2, \mathbf{z}^2), \ldots, (x_i^m, x_j^m, \mathbf{z}^m)) \sim P_{X_i, X_j, \mathbf{Z}}$$

  sampled i.i.d. (independent and identically) according to the distribution $P_{X_i, X_j, \mathbf{Z}}$
- The task is to decide whether X_i and X_j are conditionally independent given $\mathbf{Z}$ or not
- This can be done by *statistical hypothesis tests*
- We consider

  H_0, the so-called *null hypothesis* that $(X_i \perp\!\!\!\perp X_j \mid \mathbf{Z})$ and

  H_A, the *alternative hypothesis*, that $(X_i \not\perp\!\!\!\perp X_j \mid \mathbf{Z})$

PC Algorithm

Finding the Skeleton with Statistical CI Tests

- For the hypothesis test one usually constructs a *test statistic* $T_m(\mathcal{D})$ that maps $\mathcal{D}$ to a real number, and one decides to

$$\begin{cases} H_0 & \text{if } T_m(\mathcal{D}) \leq c \\ H_A & \text{if } T_m(\mathcal{D}) > c \end{cases}$$

- Ideally, we would like to have $T_m(\mathcal{D}) \leq c$ iff $(X_i \perp\!\!\!\perp X_j \mid \mathbf{Z})$
- The threshold $c \in \mathbb{R}$ is chosen such that we can control the
- *Type I error*: $T_m(\mathcal{D})$ rejects a true independence statement
- Under the null hypothesis we require

$$P(\text{Type I error}) = P(T_m(\mathcal{D}) > c \mid H_0) \ \leq \ \alpha$$

  where value α is known as the *significance level* of the test
- Moreover, α should also controll:
- *Type II error*: $T_m(\mathcal{D})$ accepts a false independence statement
- **Note** Using test T_m to learn a skeleton of a DAG, the *smaller the value of* α the larger is the value c and thus the sparser is the induced graph
- The probability of observing a value of $T_m(\mathcal{D}') > c$ under the null hypothesis of independence is known as the *p-value* of the test:

$$p\text{-value}(c) := P_{\mathcal{D}' \sim P_{X_i, X_j, \mathbf{Z}}, |\mathcal{D}'|=m}\left(T_m(\mathcal{D}') > c \mid H_0\right)$$

PC Algorithm

Finding the Skeleton with Statistical CI Tests

PC-algorithm for simulated data: Estimating the equivalence class of DAGs with corresponding *Gaussian distribution*[1]

- Next slide presents measures of the accuracy of the skeleton (false positive rate *FPR* and the true positive rate *TPR* of the edges) and structural Hamming distance *SHD* to the true cpdag, dependent on the sample size s for graphs with

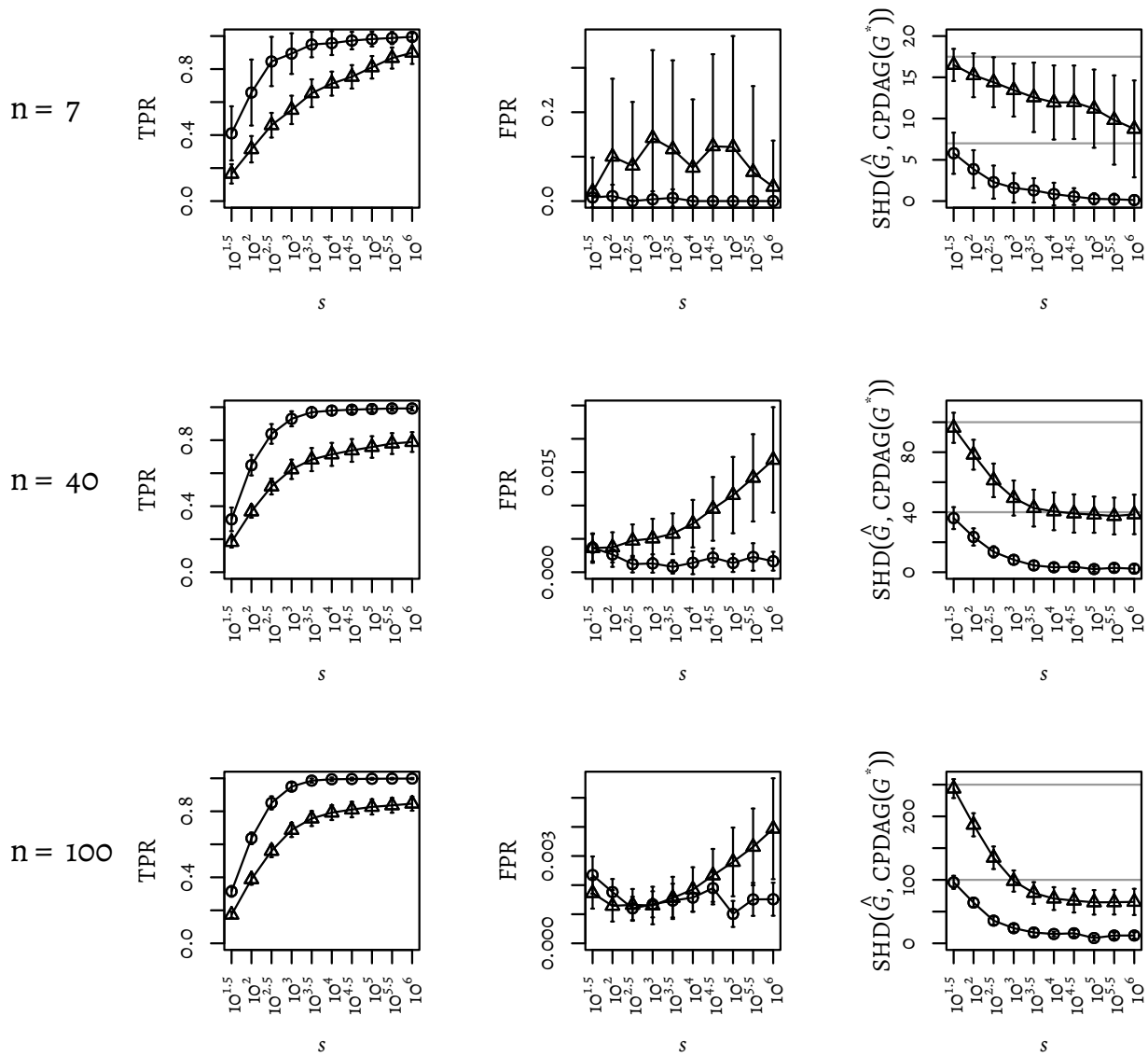
$$n \in \{7, 40, 100\}$$

  nodes.
- Data points marked with circles correspond to an expected degree of the true graph of $d = 2$, triangles to $d = 5$.
- Lines are added to improve the distinguishability of these two settings.
- Data points are computed as the mean over $r = 40$ replications and the error bars represent 95% confidence intervals.
- For comparison, faint background lines in the right plots show the mean total number of edges of the corresponding PC estimated graphs.

[1] Kalisch, M. and Bühlman, P., 2007. *Estimating high-dimensional directed acyclic graphs with the PC-algorithm*. Journal of Machine Learning Research, 8(3)

PC Algorithm

Finding the Skeleton with Statistical CI Tests



Literature

- P. Spirtes, C.N. Glymour, R. Scheines, and D. Heckerman. Causation, prediction, and search. MIT press, 2000 (Ch. 5, 6)
- D. M. Chickering. Optimal structure identification with greedy search. Journal of Machine Learning Research (2002): 507-554.
- J. Peters, D. Janzing, and B. Schölkopf. Elements of causal inference: foundations and learning algorithms. MIT press, 2017 (Ch. 7)
- J. Pearl (2009),Causality. Cambridge University Press, 2009 (Ch. 2)