Causal discovery: constraint-based methods

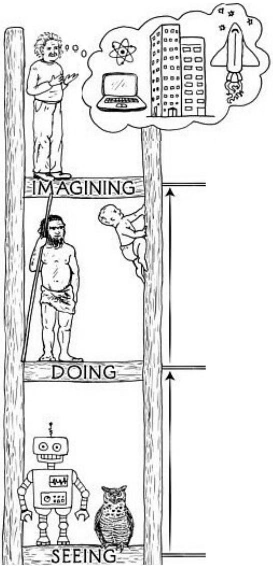
Charles K. Assaad, Emilie Devijver, Eric Gaussier

charles.assaad@ens-lyon.fr

Table of content

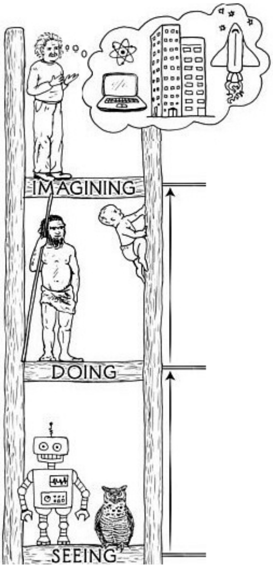
Table of content

Where does a causal graph come from?



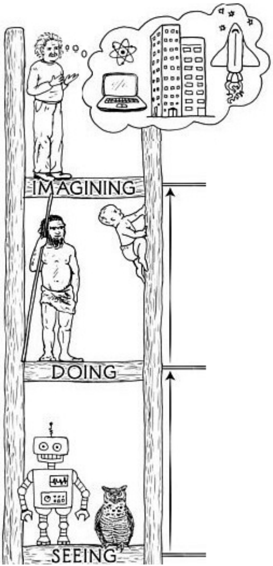
Where does a causal graph come from?

- Theory
  - Sometimes infeasible



Where does a causal graph come from?

- Theory
  - Sometimes infeasible
- Experts
  - Sometimes infeasible



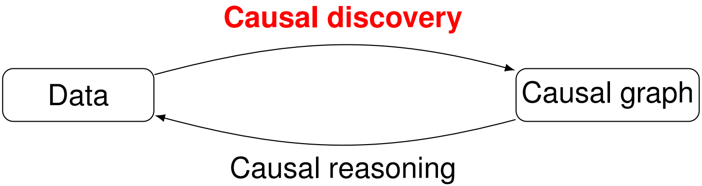
Where does a causal graph come from?

- Theory
  - Sometimes infeasible
- Experts
  - Sometimes infeasible
- Experimentations
  - Sometimes infeasible
  - Sometimes unethical
  - Costly

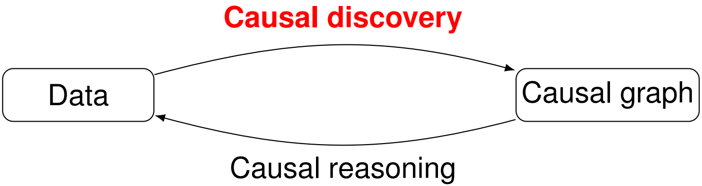
Where does a causal graph come from?

- Theory
  - Sometimes infeasible
- Experts
  - Sometimes infeasible
- Experimentations
  - Sometimes infeasible
  - Sometimes unethical
  - Costly
- Observations
  - Correlation does not imply causation!

Causal discovery (1/2)

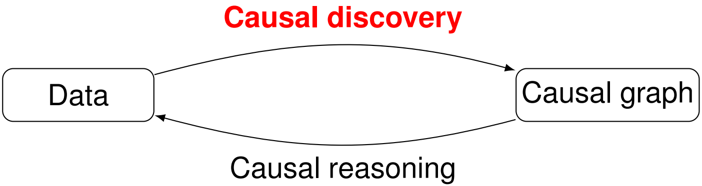


Causal discovery (1/2)



In general, causal discovery from observational data is not possible.

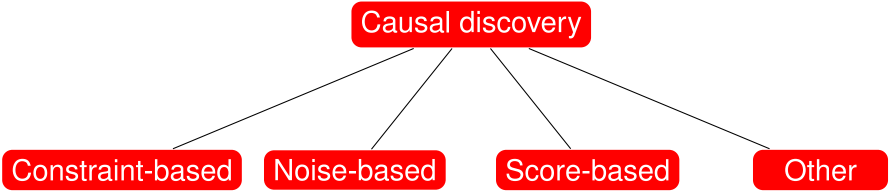
Causal discovery (1/2)



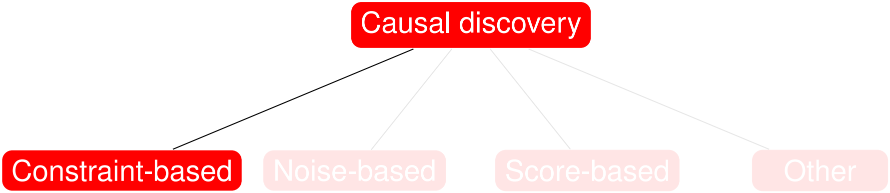
In general, causal discovery from observational data is not possible.

But it is possible under **additional assumptions**.

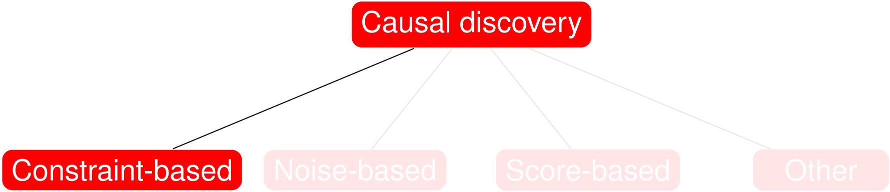
Causal discovery (2/2)



Causal discovery (2/2)



Causal discovery (2/2)



Constraint-based: run local tests of independence to create constraints on space of possible graphs.

Recap about causal graphical models (1/2)

Parental Markov Condition Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

$$\forall X \in \mathcal{V}, X \perp\!\!\!\perp_P \mathcal{V} \backslash \{Parents(X), Descendants(X)\} \mid Parents(X).$$

Collider $X \rightarrow Z \leftarrow Y$.

V-structure (or unsheilded colliders, or immorality) If the two parent vertices are not adjacent, the collider is a v-structure.

Recap about causal graphical models (1/2)

Parental Markov Condition Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

$$\forall X \in \mathcal{V}, X \perp\!\!\!\perp_P \mathcal{V} \backslash \{Parents(X), Descendants(X)\} \mid Parents(X).$$

Collider $X \rightarrow Z \leftarrow Y$.

V-structure (or unsheilded colliders, or immorality) If the two parent vertices are not adjacent, the collider is a v-structure.

Recap about causal graphical models (1/2)

Parental Markov Condition Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,

$\forall X \in \mathcal{V}, X \perp\!\!\!\perp_P \mathcal{V} \backslash \{Parents(X), Descendants(X)\} \mid Parents(X)$.

Collider $X \rightarrow Z \leftarrow Y$.

V-structure (or unsheilded colliders, or immorality) If the two parent vertices are not adjacent, the collider is a v-structure.

Recap about causal graphical models (2/2)

Theorem (probabilistic implications of d-separation) Given a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a distribution $P(\mathcal{V})$ compatible with $\mathcal{G}$ and disjoint sets $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathcal{V}$:

(i) $\mathcal{X} \perp\!\!\!\perp_G \mathcal{Y} | \mathcal{Z} \Rightarrow \mathcal{X} \perp\!\!\!\perp_P \mathcal{Y} | \mathcal{Z}$ in every distribution P compatible with $\mathcal{G}$ (Also known as the global Markov property);

(ii) If $\mathcal{X} \not\perp\!\!\!\perp_G \mathcal{Y} | \mathcal{Z}$, then there exists a distribution P compatible with $\mathcal{G}$ such that $\mathcal{X} \not\perp\!\!\!\perp_P \mathcal{Y} | \mathcal{Z}$.

Table of content

Markov equivalence for DAGs

Causal sufficiency

$$\forall X \leftarrow Z \rightarrow Y, \text{ if } X, Y \in \mathcal{V} \text{ then } Z \in \mathcal{V}.$$

Markov equivalence for DAGs

Causal sufficiency

$$\forall X \leftarrow Z \rightarrow Y, \text{ if } X, Y \in \mathcal{V} \text{ then } Z \in \mathcal{V}.$$

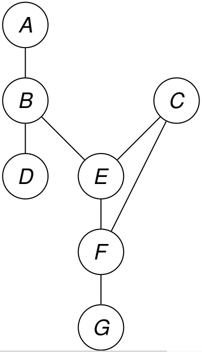
Theorem (Markov equivalence for DAGs) Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ Markov equivalent (have the same d-separations) *iff* they have the same skeleton and the same v-structures.
Proof in (Verma and Pearl, 1990)

Markov equivalence for DAGs

Causal sufficiency

$$\forall X \leftarrow Z \rightarrow Y, \text{ if } X, Y \in \mathcal{V} \text{ then } Z \in \mathcal{V}.$$

Theorem (Markov equivalence for DAGs) Two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ Markov equivalent (have the same d-separations) *iff* they have the same skeleton and the same v-structures.
Proof in (Verma and Pearl, 1990)



- Skeleton is the undirected graph with same adjacencies

A characterization of Markov equivalence classes for DAGs (1/2)

Completed partially directed acyclic graph (CPDAG) Let $[\mathcal{G}]$ be the Markov equivalence class of a DAG $\mathcal{G}$. The CPDAG $\mathcal{G}^*$ of $\mathcal{G}$ is the graph:

- With the same skeleton as $\mathcal{G}$;
- Where an edge is directed in $\mathcal{G}^*$ iff it occurs as a directed edge with the same orientation in every graph in $[\mathcal{G}]$;
- All other edges are undirected.

A characterization of Markov equivalence classes for DAGs (1/2)

Completed partially directed acyclic graph (CPDAG) Let $[\mathcal{G}]$ be the Markov equivalence class of a DAG $\mathcal{G}$. The CPDAG $\mathcal{G}^*$ of $\mathcal{G}$ is the graph:

- With the same skeleton as $\mathcal{G}$;
- Where an edge is directed in $\mathcal{G}^*$ iff it occurs as a directed edge with the same orientation in every graph in $[\mathcal{G}]$;
- All other edges are undirected.

Proposition Let $\mathcal{G}$ be a DAG and $\mathcal{G}^*$ its CPDAG. Then $\mathcal{G}$ and $\mathcal{G}^*$ are Markov equivalent.

A characterization of Markov equivalence classes for DAGs (1/2)

Completed partially directed acyclic graph (CPDAG) Let $[\mathcal{G}]$ be the Markov equivalence class of a DAG $\mathcal{G}$. The CPDAG $\mathcal{G}^*$ of $\mathcal{G}$ is the graph:

- With the same skeleton as $\mathcal{G}$;
- Where an edge is directed in $\mathcal{G}^*$ iff it occurs as a directed edge with the same orientation in every graph in $[\mathcal{G}]$;
- All other edges are undirected.

Proposition Let $\mathcal{G}$ be a DAG and $\mathcal{G}^*$ its CPDAG. Then $\mathcal{G}$ and $\mathcal{G}^*$ are Markov equivalent.
Proof: Follows immediately by Theorem (Markov equivalence for DAGs) and by Definition of CPDAG.

A characterization of Markov equivalence classes for DAGs (2/2)

Lemma Let $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ denote two CPDAGs then $\mathcal{G}_1^* = \mathcal{G}_2^*$ iff $\mathcal{G}_1$ and $\mathcal{G}_2$ belong to the same Markov equivalent class.

A characterization of Markov equivalence classes for DAGs (2/2)

Lemma Let $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ denote two CPDAGs then $\mathcal{G}_1^* = \mathcal{G}_2^*$ iff $\mathcal{G}_1$ and $\mathcal{G}_2$ belong to the same Markov equivalent class.
Proof: Follows immediately by Theorem (Markov equivalence for DAGs) and by Definition of CPDAG.

A characterization of Markov equivalence classes for DAGs (2/2)

Lemma Let $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ denote two CPDAGs then $\mathcal{G}_1^* = \mathcal{G}_2^*$ iff $\mathcal{G}_1$ and $\mathcal{G}_2$ belong to the same Markov equivalent class.

Proof: Follows immediately by Theorem (Markov equivalence for DAGs) and by Definition of CPDAG.

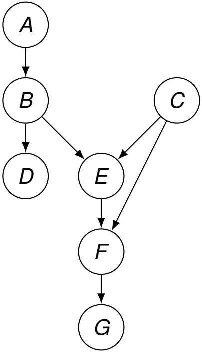
$\Longrightarrow$ All graphs in the same Markov equivalent class have the same CPDAG.

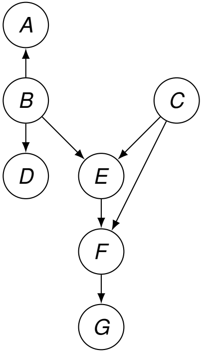
A characterization of Markov equivalence classes for DAGs (2/2)

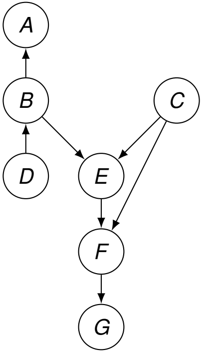
Lemma Let $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ denote two CPDAGs then $\mathcal{G}_1^* = \mathcal{G}_2^*$ iff $\mathcal{G}_1$ and $\mathcal{G}_2$ belong to the same Markov equivalent class.

Proof: Follows immediately by Theorem (Markov equivalence for DAGs) and by Definition of CPDAG.

$\Longrightarrow$ All graphs in the same Markov equivalent class have the same CPDAG.





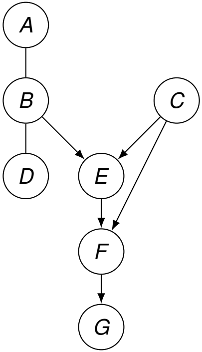


A characterization of Markov equivalence classes for DAGs (2/2)

Lemma Let $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ denote two CPDAGs then $\mathcal{G}_1^* = \mathcal{G}_2^*$ iff $\mathcal{G}_1$ and $\mathcal{G}_2$ belong to the same Markov equivalent class.

Proof: Follows immediately by Theorem (Markov equivalence for DAGs) and by Definition of CPDAG.

$\Longrightarrow$ All graphs in the same Markov equivalent class have the same CPDAG.



Constraint based question

Main question: Given $P(\mathcal{V})$ a compatible probability distribution of $\mathcal{G}$, can we discover $\mathcal{G}^*$ the CPDAG of $\mathcal{G}$?

Constraint based question

Main question: Given $P(\mathcal{V})$ a compatible probability distribution of $\mathcal{G}$, can we discover $\mathcal{G}^*$ the CPDAG of $\mathcal{G}$? No!

Constraint based question

Main question: Given $P(\mathcal{V})$ a compatible probability distribution of $\mathcal{G}$, can we discover $\mathcal{G}^*$ the CPDAG of $\mathcal{G}$? No!

Because $X \perp\!\!\!\perp_P Y \mid Z \not\Rightarrow X \perp\!\!\!\perp_G Y \mid Z$.

Faithfulness

Faithfulness We say that a graph $\mathcal{G}$ and a compatible probability distribution P are faithful to one another if all and only the conditional independence relations true in P are entailed by the Markov condition applied to $\mathcal{G}$.

faithfulness and d-sep

Theorem (implication of faithfulness on d-sep) $P(\mathcal{V})$ is faithful to directed acyclic graph $\mathcal{G}$ with vertex set $\mathcal{V}$ iff for all disjoint sets of vertices $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathcal{V}$, $\mathcal{X} \perp\!\!\!\perp_P \mathcal{Y} \mid \mathcal{Z}$ iff $\mathcal{X} \perp\!\!\!\perp_G \mathcal{Y} \mid \mathcal{Z}$.

faithfulness and d-sep

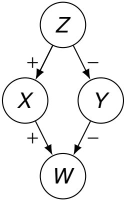
Theorem (implication of faithfulness on d-sep) $P(\mathcal{V})$ is faithful to directed acyclic graph $\mathcal{G}$ with vertex set $\mathcal{V}$ iff for all disjoint sets of vertices $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathcal{V}$, $\mathcal{X} \perp\!\!\!\perp_P \mathcal{Y} \mid \mathcal{Z}$ iff $\mathcal{X} \perp\!\!\!\perp_G \mathcal{Y} \mid \mathcal{Z}$.

Proof: Follows immediately by Theorem (probabilistic implication on d-separation) and by Definition of faithfulness.

Violation of faithfulness (1/2)

Example 1: Canceling out

Consider

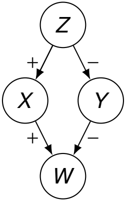


where

- $Z = \epsilon_z$
- $X = a_{zx} \times Z + \epsilon_x$
- $Y = a_{zy} \times Z + \epsilon_y$
- $W = a_{xw} \times X - \frac{a_{zx} a_{xw}}{a_{zy}} \times Y + \epsilon_w$

Example 1: Canceling out

Consider



where

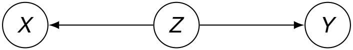
- $Z = \epsilon_z$
- $X = a_{zx} \times Z + \epsilon_x$
- $Y = a_{zy} \times Z + \epsilon_y$
- $W = a_{xw} \times X - \frac{a_{zx} a_{xw}}{a_{zy}} \times Y + \epsilon_w$

By canceling out

- $Z \perp\!\!\!\perp_P W$

Example 2: Determinism

Consider



where

- $Z = \epsilon_z$
- $X = a_{zx} \times Z + \epsilon_x$
- $Y = a_{xy} \times Z$

Violation of faithfulness (2/2)

Example 2: Determinism

Consider

$$X \longleftarrow Z \longrightarrow Y$$

where

- $Z = \epsilon_z$
- $X = a_{zx} \times Z + \epsilon_x$
- $Y = a_{xy} \times Z$

By determinism

- $X \perp\!\!\!\perp_P Z \mid Y$

Table of content

Finding skeleton and v-structures

Theorem (faithfulness, adjacencies and v-structures) If $P(\mathcal{V})$ is faithful to some directed acyclic graph, then $P(\mathcal{V})$ is faithful to directed acyclic graph $\mathcal{G}$ with vertex $\mathcal{V}$ iff:

- For $X, Y \in \mathcal{V}$, X and Y are adjacent iff $\forall \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\}$, $X \not\perp\!\!\!\perp_P Y \mid \mathcal{S}$;
- For $X, Y, Z \in \mathcal{V}$ such that X is adjacent to Z and Z is adjacent to Y and X and Y are not adjacent, $X \rightarrow Z \leftarrow Y$ in $\mathcal{G}$ iff $\forall \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\}$ suc h that $Z \in \mathcal{S}$, $X \not\perp\!\!\!\perp_P Y \mid \mathcal{S}$.

(proof on board)

Finding skeleton and v-structures

Theorem (faithfulness, adjacencies and v-structures) If $P(\mathcal{V})$ is faithful to some directed acyclic graph, then $P(\mathcal{V})$ is faithful to directed acyclic graph $\mathcal{G}$ with vertex $\mathcal{V}$ iff:

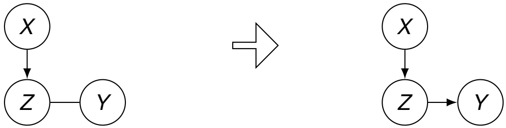
- For $X, Y \in \mathcal{V}$, X and Y are adjacent iff $\forall \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\}$, $X \not\perp\!\!\!\perp_P Y \mid \mathcal{S}$;
- For $X, Y, Z \in \mathcal{V}$ such that X is adjacent to Z and Z is adjacent to Y and X and Y are not adjacent, $X \rightarrow Z \leftarrow Y$ in $\mathcal{G}$ iff $\forall \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\}$ suc h that $Z \in \mathcal{S}$, $X \not\perp\!\!\!\perp_P Y \mid \mathcal{S}$.

(proof on board)

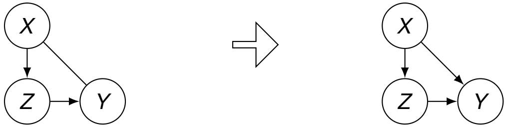
- Point 1 can be used to discover the skeleton of $\mathcal{G}$ from $P(\mathcal{V})$;
- Given the skeleton of $\mathcal{G}$, point 2 can be used to find all v-structures.

Orientation rules

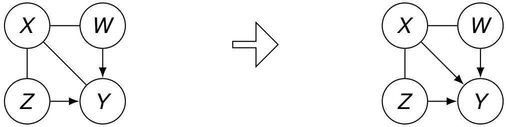
R1:



R2:



R3:



Orientation rules correctness

Pattern A pattern of a DAG $\mathcal{G}$ is a graph with the same skeleton as $\mathcal{G}$ but where only v-structures are oriented.

Orientation rules correctness

Pattern A pattern of a DAG $\mathcal{G}$ is a graph with the same skeleton as $\mathcal{G}$ but where only v-structures are oriented.

Theorem (orientation soundness) Given a pattern of some DAG, the three orientation rules $R1$, $R2$, $R3$ are sound.
(proof on board)

Orientation rules correctness

Pattern A pattern of a DAG $\mathcal{G}$ is a graph with the same skeleton as $\mathcal{G}$ but where only v-structures are oriented.

Theorem (orientation soundness) Given a pattern of some DAG, the three orientation rules $R1$, $R2$, $R3$ are sound.
(proof on board)

Theorem (orientation completeness) The result of recursively applying rules $R1$, $R2$, $R3$ to a pattern of some DAG is a CPDAG.
(proof in (Meek, 1995))

The SGS algorithm

Algorithm 1 SGS

Input: $P(\mathcal{V})$
Output: CPDAG $\mathcal{G}^*$

Form the complete undirected graph $\mathcal{G}^*$ on vertex set $\mathcal{V}$
for all $X - Y$ in $\mathcal{G}^*$
  and subsets $\mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\}$ **do**
  if $\exists \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\}$ such that $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ **then**
    Delete edge $X - Y$ from $\mathcal{G}^*$
  end if
end for
for all $X - Z - Y$ in $\mathcal{G}^*$ such that $X \notin Adj(Y, \mathcal{G}^*)$ **do**
  if $\nexists\, \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\}$ such that $Z \in \mathcal{S}$ and $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ **then**
    Orient $X \rightarrow Z \leftarrow Y$ in $\mathcal{G}^*$
  end if
end for
Recursively apply rules R1-R3 until no more edges can be oriented
Return $\mathcal{G}^*$

$Adj(Y, \mathcal{G})$: Adjacencies of Y in $\mathcal{G}$

Correctness of SGS

Theorem (correctness) Assume the distribution $P(\mathcal{V})$ is Markov and faithful to some DAG $\mathcal{G}$ and assume that we are given perfect conditional independence information about all pairs of variables. Let $\mathcal{G}^*$ be the CPDAG of $\mathcal{G}$. The SGS algorithm returns $\mathcal{G}^*$.

Correctness of SGS

Theorem (correctness) Assume the distribution $P(\mathcal{V})$ is Markov and faithful to some DAG $\mathcal{G}$ and assume that we are given perfect conditional independence information about all pairs of variables. Let $\mathcal{G}^*$ be the CPDAG of $\mathcal{G}$. The SGS algorithm returns $\mathcal{G}^*$.

Proof: By Theorem (faithfulness, adjacencies and v-structures), Theorem (orientation soundness) and Theorem (orientation completness).

Computational complexity of SGS

Running time of SGS depends *exponentially* on the *number of vertices* in the graph:

- For all pairs check all subsets;
- For all triples check all subsets.

A better approach?

Optimizing the procedure for skeleton construction

A better approach?

Optimizing the procedure for skeleton construction

By the Parental Markov condition:

$X \notin Adj(Y, \mathcal{G})$ iff $X \perp\!\!\!\perp_P Y \mid Parents(X, \mathcal{G})$ or $X \perp\!\!\!\perp_P Y \mid Parents(Y, \mathcal{G})$

A better approach?

Optimizing the procedure for skeleton construction

By the Parental Markov condition:

$X \notin Adj(Y, \mathcal{G})$ iff $X \perp\!\!\!\perp_P Y \mid Parents(X, \mathcal{G})$ or $X \perp\!\!\!\perp_P Y \mid Parents(Y, \mathcal{G})$

Since the graph $\mathcal{G}$ is unknown:

- The parent set is unknown ahead of time;
- We look at $\mathcal{S} \subseteq Adj(X, \mathcal{G}')$ and $\mathcal{S}' \subseteq Adj(Y, \mathcal{G}')$ for some G' which is a supergraph of the true unknown skeleton;
- We can pursue an iterative strategy such that we increase the size of $\mathcal{S}$ iteratively.

A better approach?

Optimizing the procedure for finding v-structures

A better approach?

Optimizing the procedure for finding v-structures

Lemma (either d-sep or d-connect) Given the distribution $P(V)$ that is Markov and faithful to some DAG $\mathcal{G}$, if $Z \in Adj(X, \mathcal{G})$, $Z \in Adj(Y, \mathcal{G})$ and $Y \notin Adj(X, \mathcal{G})$, then either Z is in every set of variables that d-separates X and Y or it is in no set of variables that d-separates X and Y.
(proof on board)

Optimizing the procedure for finding v-structures

Lemma (either d-sep or d-connect) Given the distribution $P(V)$ that is Markov and faithful to some DAG $\mathcal{G}$, if $Z \in Adj(X,\mathcal{G})$, $Z \in Adj(Y,\mathcal{G})$ and $Y \notin Adj(X,\mathcal{G})$, then either Z is in every set of variables that d-separates X and Y or it is in no set of variables that d-separates X and Y.
(proof on board)

$sepset(X, Y)$: subset that permitted the separation of X and Y during the skeleton construction.

A better approach?

Optimizing the procedure for finding v-structures

Lemma (either d-sep or d-connect) Given the distribution $P(V)$ that is Markov and faithful to some DAG $\mathcal{G}$, if $Z \in Adj(X, \mathcal{G})$, $Z \in Adj(Y, \mathcal{G})$ and $Y \notin Adj(X, \mathcal{G})$, then either Z is in every set of variables that d-separates X and Y or it is in no set of variables that d-separates X and Y.
(proof on board)

$sepset(X, Y)$: subset that permitted the separation of X and Y during the skeleton construction.

R0: For all triples $X - Z - Y \in \mathcal{G}^*$ such that $Y \notin Adj(X, \mathcal{G}^*)$, if $Z \notin sepset(X, Y)$ then orient $X \rightarrow Z \leftarrow Y$ in $\mathcal{G}^*$.

The PC algorithm

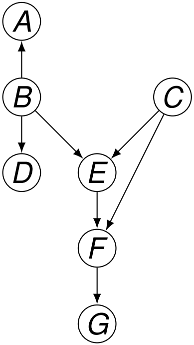
Algorithm 2 PC

Input: $P(\mathcal{V})$
Output: CPDAG $\mathcal{G}^*$

Form the complete undirected graph $\mathcal{G}^*$ on vertex set $\mathcal{V}$
Let $n = 0$
repeat
    for all $X - Y$ in $\mathcal{G}^*$ such that $|Adj(X, \mathcal{G}^*) \backslash \{Y\}| \geq n$
    and subsets $\mathcal{S} \subseteq Adj(X, \mathcal{G}^*) \backslash \{Y\}$ such that $|\mathcal{S}| = n$ **do**
        if $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ **then**
            Delete edge $X - Y$ from $\mathcal{G}^*$
            Let $sepset(X, Y) = sepset(Y, X) = \mathcal{S}$
        end if
    end for
    Let $n = n + 1$
until for each pair of adjacent vertices (X, Y), $|Adj(X, \mathcal{G}^*) \backslash \{Y\}| < n$
Apply R0
Recursively apply rules R1-R3 until no more edges can be oriented
Return $\mathcal{G}^*$

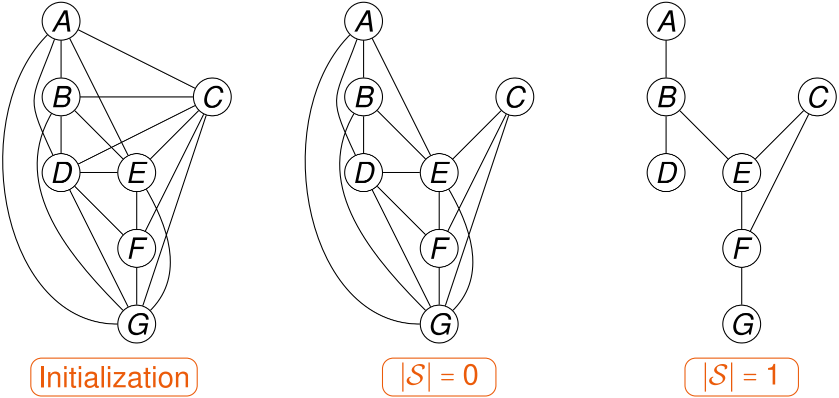
PC in action (1/3)

- Suppose the true graph on right;
- Assumptions: CMC, faithfulness, causal sufficiency.



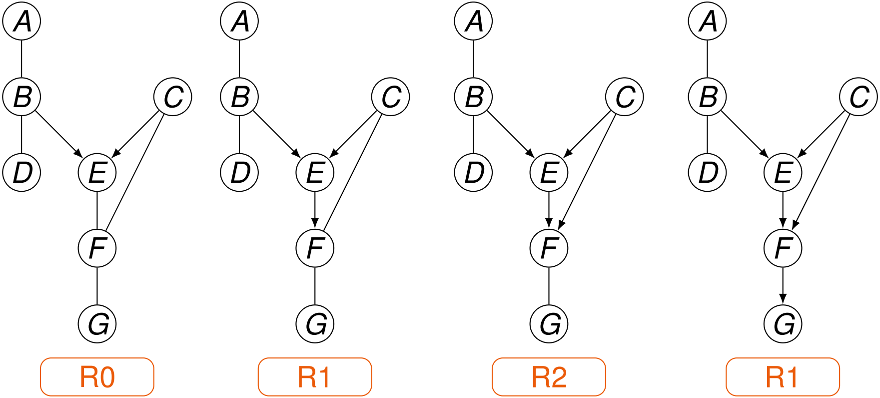
PC in action (2/3)

Skeleton construction:



PC in action (3/3)

Orientation:



Correctness of PC

Theorem (correctness) Assume the distribution $P(\mathcal{V})$ is Markov and faithful to some DAG $\mathcal{G}$ and assume that we are given perfect conditional independence information about all pairs of variables. Let $\mathcal{G}^*$ be the CPDAG of $\mathcal{G}$. The PC algorithm returns $\mathcal{G}^*$.
(proof on board)

Computational complexity of PC

Running time of PC depends *exponentially* on the *maximal degree* of the graph **but** for a fixed maximal degree running time over the *number of vertices* is *polynomial*.

Exercise 1

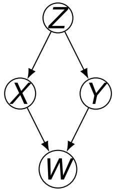
Consider data that are generated from a chain $X \rightarrow Y \rightarrow Z$. Assuming that all assumptions are satisfied, which CPDAG would a constraint based causal discovery algorithm report?

If you could supply prior knowledge to the algorithm on only one arc that is required to be present, what arc (if any) would allow the entire structure to be learned? Explain briefly.

Exercise 2

Consider data that truly come from a fork $X \leftarrow Y \rightarrow Z$.
Assuming that all assumptions are satisfied, which CPDAG would a constraint based causal discovery algorithm report?

If you could supply prior knowledge to the algorithm on only one arc that is required to be present, what arc (if any) would allow the entire structure to be learned? Explain briefly.



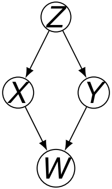
- Suppose the true graph on right;
- Assumptions: CMC, causal sufficiency, no deterministic relations;
- Generative process:

$$
\begin{aligned}
Z &= \xi_z & \xi_z &\sim N(0,1);\\
X &= a * Z + \xi_x & \xi_x &\sim N(0,1);\\
Y &= b * Z + \xi_y & \xi_y &\sim N(0,1);\\
W &= c * X - \frac{a * c}{b} * Y + \xi_w & \xi_w &\sim N(0,1).
\end{aligned}
$$

- Given a compatible distribution what would be the output of the PC algorithm?

Exercise 4

- Suppose the true graph on right;
- Assumptions: CMC, causal sufficiency, deterministic relations, no canceling out paths;
- Given a compatible distribution what would be the output of the PC algorithm?



Exercise 5

- Suppose the true graph on right;
- Assumptions: CMC, faithfulness;
- Given a compatible distribution what would be the output of the PC algorithm if Z is unobserved?

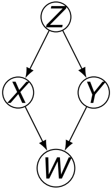


Table of content

Latent variables (1/2)

Consider $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = \mathcal{O} \cup \mathcal{L}$ such that

- $\mathcal{O}$ observable variables;
- $\mathcal{L}$ latent variables.

Latent variables (1/2)

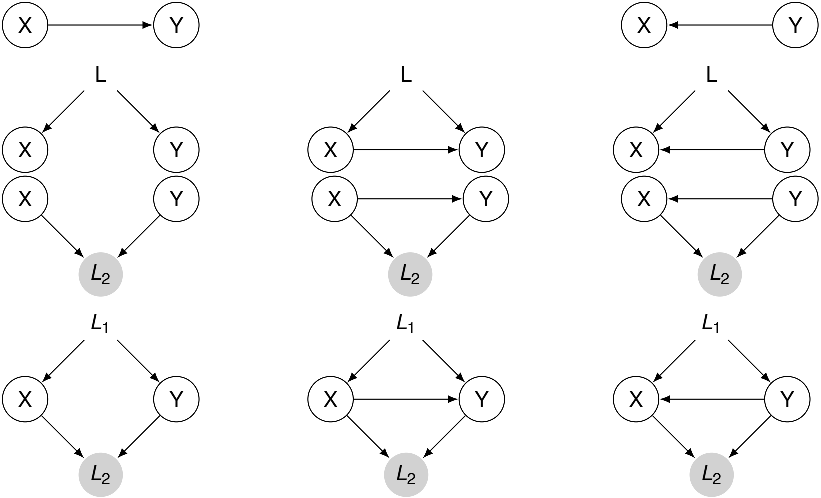
Consider $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = \mathcal{O} \cup \mathcal{L}$ such that

- $\mathcal{O}$ observable variables;
- $\mathcal{L}$ latent variables.

Latent variables are represented by a transparent border.

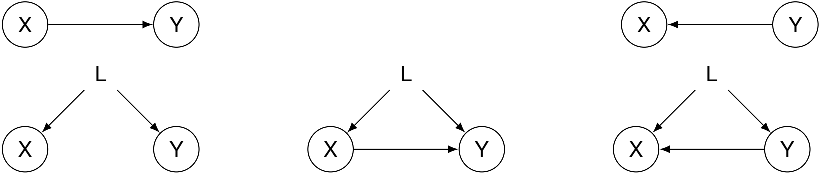
Latent variables (2/2)

Assuming acyclicity, if two observed variables X and Y are statistically dependent:

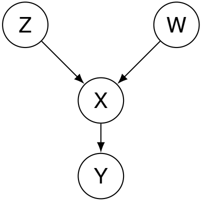


Latent variables (2/2)

Assuming acyclicity, if two observed variables X and Y are statistically dependent:



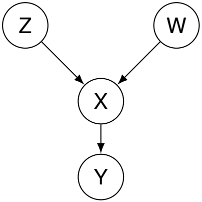
Main structures (1/2)



Y-structure

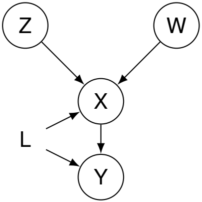
$$Z \perp\!\!\!\perp_P W$$
$$Z \not\!\perp\!\!\!\perp_P W \mid X$$
$$Y \not\!\perp\!\!\!\perp_P Z$$
$$Y \perp\!\!\!\perp_P Z \mid X$$
$$Y \not\!\perp\!\!\!\perp_P W$$
$$Y \perp\!\!\!\perp_P W \mid X$$

Main structures (1/2)



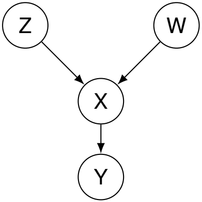
Y-structure

$$Z \perp\!\!\!\perp_P W$$
$$Z \not\!\perp\!\!\!\perp_P W \mid X$$
$$Y \not\!\perp\!\!\!\perp_P Z$$
$$Y \perp\!\!\!\perp_P Z \mid X$$
$$Y \not\!\perp\!\!\!\perp_P W$$
$$Y \perp\!\!\!\perp_P W \mid X$$



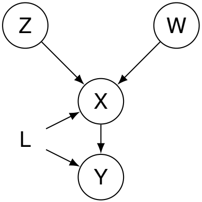
$$Z \perp\!\!\!\perp_P W$$
$$Z \not\!\perp\!\!\!\perp_P W \mid X$$
$$Y \not\!\perp\!\!\!\perp_P Z$$
$$Y \not\!\perp\!\!\!\perp_P Z \mid X$$
$$Y \not\!\perp\!\!\!\perp_P W$$
$$Y \not\!\perp\!\!\!\perp_P W \mid X$$

Main structures (1/2)



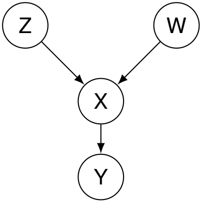
Y-structure

$$Z \perp\!\!\!\perp_P W$$
$$Z \not\perp\!\!\!\perp_P W \mid X$$
$$Y \not\perp\!\!\!\perp_P Z$$
$$Y \perp\!\!\!\perp_P Z \mid X$$
$$Y \not\perp\!\!\!\perp_P W$$
$$Y \perp\!\!\!\perp_P W \mid X$$



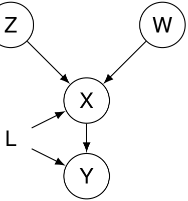
$$Z \perp\!\!\!\perp_P W$$
$$Z \not\perp\!\!\!\perp_P W \mid X$$
$$Y \not\perp\!\!\!\perp_P Z$$
$$Y \not\perp\!\!\!\perp_P Z \mid X$$
$$Y \not\perp\!\!\!\perp_P W$$
$$Y \not\perp\!\!\!\perp_P W \mid X$$

Main structures (1/2)



Y-structure

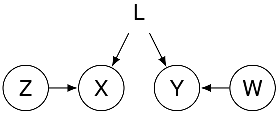
$$Z \perp\!\!\!\perp_P W$$
$$Z \not\!\perp\!\!\!\perp_P W \mid X$$
$$Y \not\!\perp\!\!\!\perp_P Z$$
$$Y \perp\!\!\!\perp_P Z \mid X$$
$$Y \not\!\perp\!\!\!\perp_P W$$
$$Y \perp\!\!\!\perp_P W \mid X$$



$$Z \perp\!\!\!\perp_P W$$
$$Z \not\!\perp\!\!\!\perp_P W \mid X$$
$$Y \not\!\perp\!\!\!\perp_P Z$$
$$Y \not\!\perp\!\!\!\perp_P Z \mid X$$
$$Y \not\!\perp\!\!\!\perp_P W$$
$$Y \not\!\perp\!\!\!\perp_P W \mid X$$

Pattern of independence can rule out latent confounding.

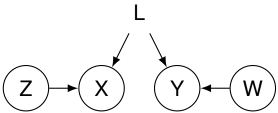
Main structures (2/2)



W-structure

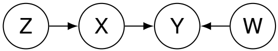
$$Z \not\perp\!\!\!\perp_P X$$
$$X \not\perp\!\!\!\perp_P Y$$
$$Y \not\perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P Y$$
$$X \perp\!\!\!\perp_P W$$
$$Z \not\perp\!\!\!\perp_P Y \mid X$$
$$X \not\perp\!\!\!\perp_P W \mid Y$$

Main structures (2/2)



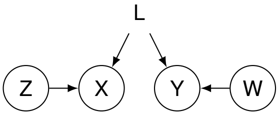
W-structure

$$Z \not\perp\!\!\!\perp_P X$$
$$X \not\perp\!\!\!\perp_P Y$$
$$Y \not\perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P Y$$
$$X \perp\!\!\!\perp_P W$$
$$Z \not\perp\!\!\!\perp_P Y \mid X$$
$$X \not\perp\!\!\!\perp_P W \mid Y$$



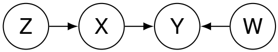
$$Z \not\perp\!\!\!\perp_P X$$
$$X \not\perp\!\!\!\perp_P Y$$
$$Y \not\perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P W$$
$$Z \not\perp\!\!\!\perp_P Y$$
$$X \perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P Y \mid X$$
$$X \not\perp\!\!\!\perp_P W \mid Y$$

Main structures (2/2)



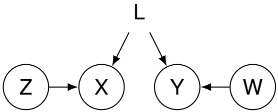
W-structure

$$Z \not\perp\!\!\!\perp_P X$$
$$X \not\perp\!\!\!\perp_P Y$$
$$Y \not\perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P Y$$
$$X \perp\!\!\!\perp_P W$$
$$Z \not\perp\!\!\!\perp_P Y \mid X$$
$$X \not\perp\!\!\!\perp_P W \mid Y$$



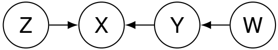
$$Z \not\perp\!\!\!\perp_P X$$
$$X \not\perp\!\!\!\perp_P Y$$
$$Y \not\perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P W$$
$$Z \not\perp\!\!\!\perp_P Y$$
$$X \perp\!\!\!\perp_P W$$
$$Z \perp\!\!\!\perp_P Y \mid X$$
$$X \not\perp\!\!\!\perp_P W \mid Y$$

Main structures (2/2)



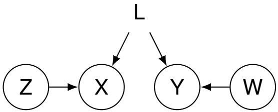
W-structure

$Z \not\perp\!\!\!\perp_P X$
$X \not\perp\!\!\!\perp_P Y$
$Y \not\perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P Y$
$X \perp\!\!\!\perp_P W$
$Z \not\perp\!\!\!\perp_P Y \mid X$
$X \not\perp\!\!\!\perp_P W \mid Y$



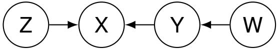
$Z \not\perp\!\!\!\perp_P X$
$X \not\perp\!\!\!\perp_P Y$
$Y \not\perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P Y$
$X \not\perp\!\!\!\perp_P W$
$Z \not\perp\!\!\!\perp_P Y \mid X$
$X \not\perp\!\!\!\perp_P W \mid Y$

Main structures (2/2)



W-structure

$Z \not\perp\!\!\!\perp_P X$
$X \not\perp\!\!\!\perp_P Y$
$Y \not\perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P Y$
$X \perp\!\!\!\perp_P W$
$Z \not\perp\!\!\!\perp_P Y \mid X$
$X \not\perp\!\!\!\perp_P W \mid Y$



$Z \not\perp\!\!\!\perp_P X$
$X \not\perp\!\!\!\perp_P Y$
$Y \not\perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P W$
$Z \perp\!\!\!\perp_P Y$
$X \not\perp\!\!\!\perp_P W$
$Z \not\perp\!\!\!\perp_P Y \mid X$
$X \not\perp\!\!\!\perp_P W \mid Y$

Pattern of independence can suggest latent confounding.

Graphical representation of causal graphs with latent counfounding

- DAGs are not sufficient to represent a graph over $\mathcal{O}$ alone;
- Acyclic directed mixed graphs (ADMG) are sufficient to represent a graph over $\mathcal{O}$ alone.

Graphical representation of causal graphs with latent counfounding

- DAGs are not sufficient to represent a graph over $\mathcal{O}$ alone;
- Acyclic directed mixed graphs (ADMG) are sufficient to represent a graph over $\mathcal{O}$ alone.

Acyclic directed mixed graphs: Given a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ such that $\mathcal{V} = \mathcal{O} \cup \mathcal{L}$, the corresponding ADMG is $\mathcal{M} = (\mathcal{V}', \mathcal{E}')$ with $\mathcal{V}' = \mathcal{O}$ such that for any $X, Y \in \mathcal{O}$:

- $X \rightarrow Y$ in $\mathcal{M}$ if there exists a directed path from from X to Y in $\mathcal{G}$;
- $X \leftrightarrow Y$ in $\mathcal{M}$ if there exists a path π from X to Y of the form $X \leftarrow \cdots \rightarrow Y$ such that:
  - $\forall W \in \pi$, $W \in \mathcal{L}$ or $W \in \{X, Y\}$;
  - there is no colliders on π.

Mixed graphs limitations

- In ADMG Markov equivalence is complicated;
- ADMG are not maximal:

Maximality A graph is maximal if for every pair of vertices X and Y

$$X \notin Adj(Y, \mathcal{M}) \implies \exists \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\} \text{ such that } X \perp\!\!\!\perp_P Y \mid \mathcal{S}.$$

Mixed graphs limitations

- In ADMG Markov equivalence is complicated;
- ADMG are not maximal:

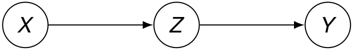
Maximality A graph is maximal if for every pair of vertices X and Y

$$X \notin Adj(Y, \mathcal{M}) \implies \exists \mathcal{S} \subseteq \mathcal{V} \backslash \{X, Y\} \text{ such that } X \perp\!\!\!\perp_P Y \mid \mathcal{S}.$$

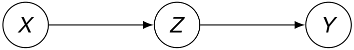
- $\implies$ ADMGs cannot be learned in PC-style procedure.

DAGs to ADMG examples

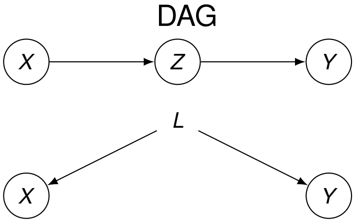
DAG

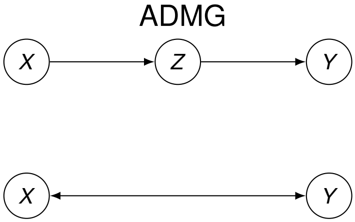


ADMG

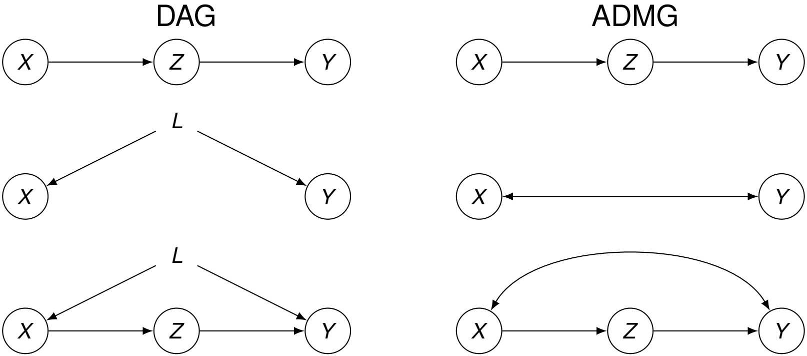


DAGs to ADMG examples

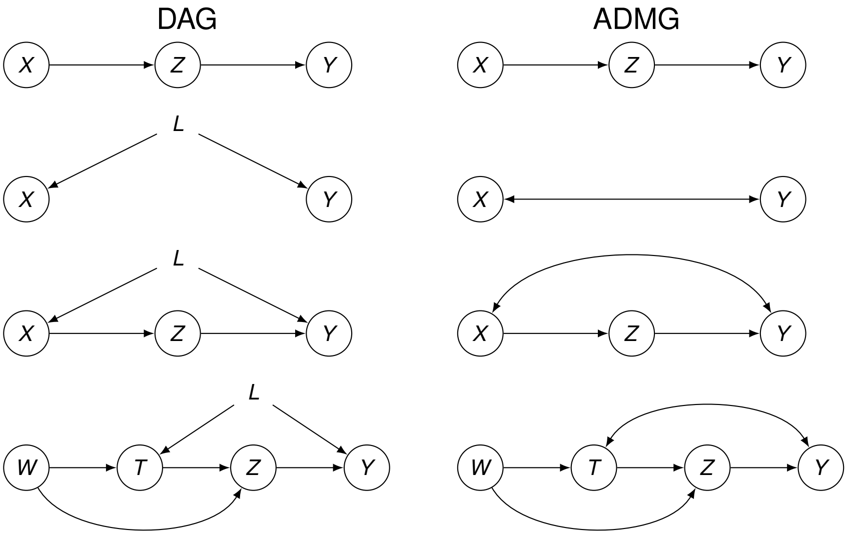




DAGs to ADMG examples



DAGs to ADMG examples

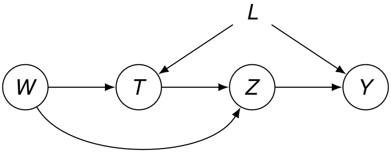


Inducing path

Inducing path: An inducing path relative to $\mathcal{L}$ is a path on which every vertex not in $\mathcal{L}$ except the endpoints is a collider on the path and every collider is an ancestor of an endpoint of the path.

Inducing path

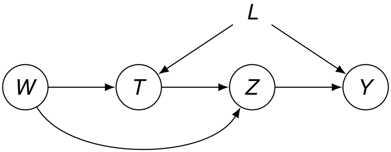
Inducing path: An inducing path relative to $\mathcal{L}$ is a path on which every vertex not in $\mathcal{L}$ except the endpoints is a collider on the path and every collider is an ancestor of an endpoint of the path.



W and Y have an inducing path relative to L.

Inducing path

Inducing path: An inducing path relative to $\mathcal{L}$ is a path on which every vertex not in $\mathcal{L}$ except the endpoints is a collider on the path and every collider is an ancestor of an endpoint of the path.



W and Y have an inducing path relative to L.

Theorem (inducing path implies d-connection): If $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is DAG such that $\mathcal{V} = \mathcal{O} \cup \mathcal{L}$. X and Y are not d-seperated by a subset $\mathcal{S} \subseteq \mathcal{O} \backslash \{X, Y\}$ iff there is an inducing path relative to $\mathcal{L}$ between X and Y.

(proof in (Spirtes et al, 2000))

Maximal ancestral graphs

Maximal ancestral graphs[1]: Given a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ such that $\mathcal{V} = \mathcal{O} \cup \mathcal{L}$, the corresponding MAG is $\mathcal{M} = (\mathcal{V}', \mathcal{E}')$ with $\mathcal{V}' = \mathcal{O}$ such that for any $X, Y \in \mathcal{O}$:

- For each pair of vertices $X, Y \in \mathcal{O}$, $X - Y$ in $\mathcal{M}$ iff there is an inducing path between them relative to $\mathcal{L}$ in $\mathcal{G}$;
- For each pair of adjacent vertices $X - Y$ in $\mathcal{M}$:
  - $X \rightarrow Y$ if X is an ancestor of Y in $\mathcal{G}$;
  - $Y \rightarrow X$ if Y is an ancestor of X in $\mathcal{G}$;
  - $X \leftrightarrow Y$ otherwise.

[1] MAGs can also handle selection bias by using undirected edges.

Maximal ancestral graphs

Maximal ancestral graphs[1]: Given a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ such that $\mathcal{V} = \mathcal{O} \cup \mathcal{L}$, the corresponding MAG is $\mathcal{M} = (\mathcal{V}', \mathcal{E}')$ with $\mathcal{V}' = \mathcal{O}$ such that for any $X, Y \in \mathcal{O}$:

- For each pair of vertices $X, Y \in \mathcal{O}$, $X - Y$ in $\mathcal{M}$ iff there is an inducing path between them relative to $\mathcal{L}$ in $\mathcal{G}$;
- For each pair of adjacent vertices $X - Y$ in $\mathcal{M}$:
  - $X \rightarrow Y$ if X is an ancestor of Y in $\mathcal{G}$;
  - $Y \rightarrow X$ if Y is an ancestor of X in $\mathcal{G}$;
  - $X \leftrightarrow Y$ otherwise.

- MAGs do not contain any directed and almost directed cycles (ancestrality);
- In a MAG there is no inducing path between any two non-adjacent vertices (maximality).

[1] MAGs can also handle selection bias by using undirected edges.

MAGs interpretation, advatanges and limitation

Interpretation:

- $X \rightarrow Y$ in a MAG: X is an ancestor of Y in the underlying DAG;
- $X \leftrightarrow Y$ in a MAG: X is not an ancestor of Y and Y is not an ancestor of X, which means there is a hidden coufounder between X and Y.

MAGs interpretation, advatanges and limitation

Interpretation:

- $X \to Y$ in a MAG: X is an ancestor of Y in the underlying DAG;
- $X \leftrightarrow Y$ in a MAG: X is not an ancestor of Y and Y is not an ancestor of X, which means there is a hidden coufounder between X and Y.

Advatanges:

- Markov equivalence is possible;
- They are maximal;
- They have some properties of ADMG:
  For any disjoint sets of vertices $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathcal{O}$:

$$\mathcal{X} \perp\!\!\!\perp_M \mathcal{Y} \mid \mathcal{Z} \implies \mathcal{X} \perp\!\!\!\perp_P \mathcal{Y} \mid \mathcal{Z}$$

$$\mathcal{X} \perp\!\!\!\perp_M \mathcal{Y} \mid \mathcal{Z} \iff \mathcal{X} \perp\!\!\!\perp_G \mathcal{Y} \mid \mathcal{Z}$$

MAGs interpretation, advatanges and limitation

Interpretation:

- $X \rightarrow Y$ in a MAG: X is an ancestor of Y in the underlying DAG;
- $X \leftrightarrow Y$ in a MAG: X is not an ancestor of Y and Y is not an ancestor of X, which means there is a hidden coufounder between X and Y.

Advatanges:

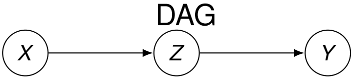
- Markov equivalence is possible;
- They are maximal;
- They have some properties of ADMG:
  For any disjoint sets of vertices $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathcal{O}$:

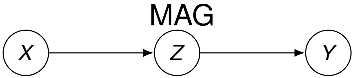
$$\mathcal{X} \perp\!\!\!\perp_M \mathcal{Y} \mid \mathcal{Z} \implies \mathcal{X} \perp\!\!\!\perp_P \mathcal{Y} \mid \mathcal{Z}$$

$$\mathcal{X} \perp\!\!\!\perp_M \mathcal{Y} \mid \mathcal{Z} \iff \mathcal{X} \perp\!\!\!\perp_G \mathcal{Y} \mid \mathcal{Z}$$

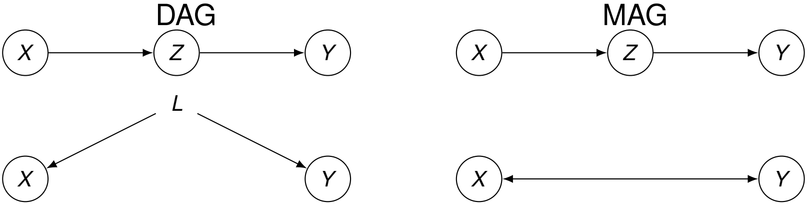
- $\implies$ MAGs can be learned in PC-style procedure!

DAGs to MAGs examples

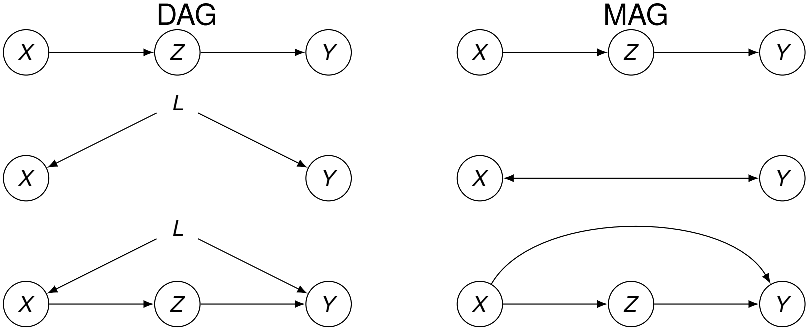




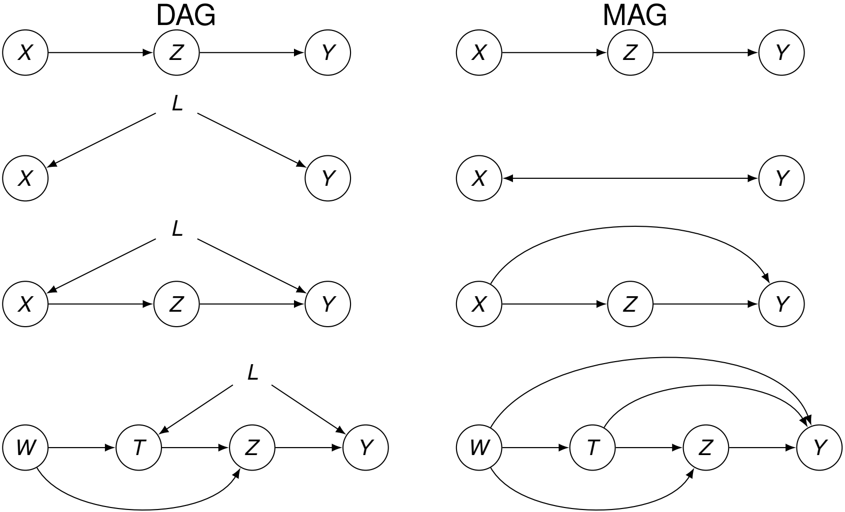
DAGs to MAGs examples



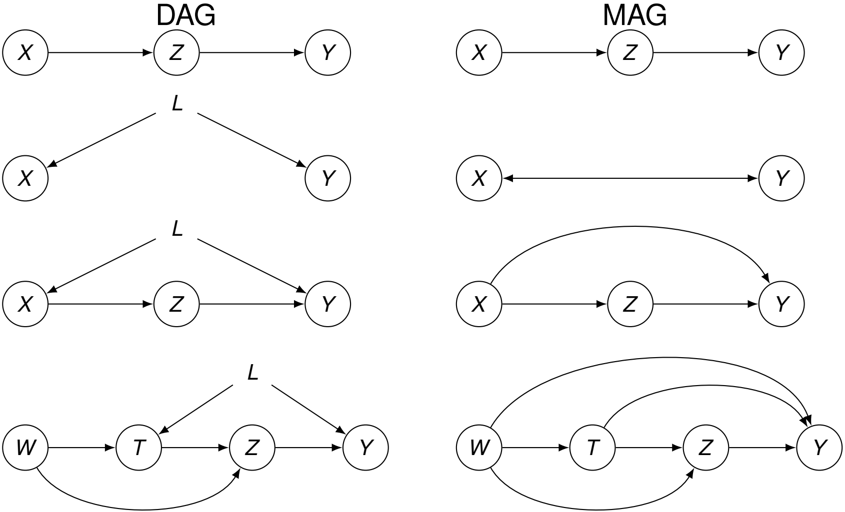
DAGs to MAGs examples



DAGs to MAGs examples



DAGs to MAGs examples

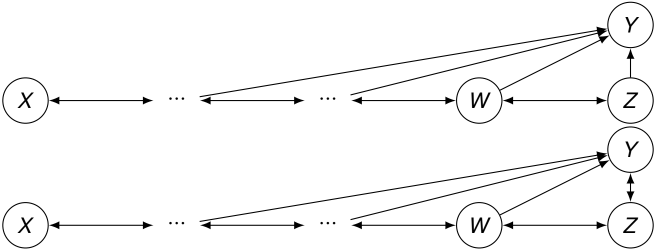


MAGs are less informative than ADMGs.

Discriminating path

Discriminating path: In a MAG, a path between X and Y, $\pi = < X, \cdots, W, Z, Y >$, is a discriminating path for Z if:

- π includes at least three edges;
- Z is a non-endpoint vertex on π;
- X is not adjacent to Y, and every vertex between X and Z is a collider on π and is a parent of Y.



Markov equivalence classes for MAGs

Theorem (Markov equivalence for MAGs) Two MAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ are Markov equivalent iff:

- They have the same adjacencies;
- They have the same v-structures;
- If a path π is a discriminating path for a vertex Z in both graphs, then Z is a collider on the path in one graph iff it is a collider on the path in the other.

(proof in (Spirtes and Richardson, 1997))

A characterization of Markov equivalence classes for MAGs

Maximally informative partial ancestral graph (MIPAG) Let $[\mathcal{M}]$ be the Markov equivalence class of a MAG $\mathcal{M}$. A MIPAG $\mathcal{M}^*$ for $[\mathcal{M}]$ is a graph with possibly three kinds of marks and hence six kinds of edges:

$$-, \rightarrow, \leftrightarrow, \circ\!-, \circ\!\!-\!\!\circ, \circ\!\!\rightarrow$$

such that:

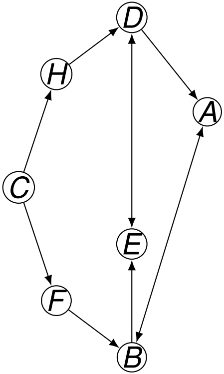
- $\mathcal{M}^*$ has the same adjacencies as $\mathcal{M}$ (and any member of $[\mathcal{M}]$);
- Every non-circle mark in $\mathcal{M}^*$ is an invariant mark in $[\mathcal{M}]$;
- Every circle in $\mathcal{M}^*$ corresponds to a variant mark in $[\mathcal{M}]$.

dsep sets

In MAGs, $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ such that $\mathcal{S} \subseteq \mathcal{O}$
$\not\Rightarrow X \perp\!\!\!\perp_P Y \mid Parents(X, \mathcal{M})$ or $X \perp\!\!\!\perp_P Y \mid Parents(Y, \mathcal{M})$

dsep sets

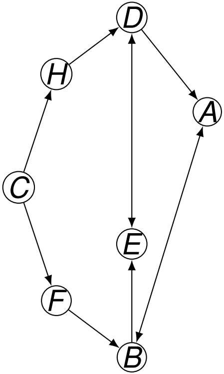
In MAGs, $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ such that $\mathcal{S} \subseteq \mathcal{O}$
$\not\Longrightarrow X \perp\!\!\!\perp_P Y \mid Parents(X, \mathcal{M})$ or $X \perp\!\!\!\perp_P Y \mid Parents(Y, \mathcal{M})$



$A \not\perp\!\!\!\perp_P E \mid B, D$
$A \perp\!\!\!\perp_P E \mid B, D, F$
$A \perp\!\!\!\perp_P E \mid B, D, H$

dsep sets

In MAGs, $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ such that $\mathcal{S} \subseteq \mathcal{O}$
$\not\Longrightarrow X \perp\!\!\!\perp_P Y \mid Parents(X, \mathcal{M})$ or $X \perp\!\!\!\perp_P Y \mid Parents(Y, \mathcal{M})$



$A \not\perp\!\!\!\perp_P E \mid B, D$
$A \perp\!\!\!\perp_P E \mid B, D, F$
$A \perp\!\!\!\perp_P E \mid B, D, H$

dsep set: $Z \in dsep(X, Y)$ iff there is an undirected path between X and Z on which every vertex except the endpoint is a collider, and each vertex is an ancestor of X or Y.

Possible dsep sets

Given a pair of vertices X, Y, how to find the d-sep sets without examining every subset of $\mathcal{O} \backslash \{X, Y\}$?

Possible dsep sets

Given a pair of vertices X, Y, how to find the d-sep sets without examining every subset of $\mathcal{O} \backslash \{X, Y\}$?

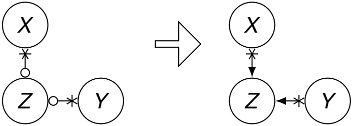
Possible-d-sep set (pds): $Z \in pds(X, Y)$ iff there is an undirected path π between X and Z such that every subpath $< A, B, C >$ on the path is either a v-structure or form a triangle.

Possible dsep sets

Given a pair of vertices X, Y, how to find the d-sep sets without examining every subset of $\mathcal{O} \backslash \{X, Y\}$?

Possible-d-sep set (pds): $Z \in pds(X, Y)$ iff there is an undirected path π between X and Z such that every subpath $< A, B, C >$ on the path is either a v-structure or form a triangle.

If there exists $\mathcal{S} \subseteq \mathcal{O} \backslash \{X, Y\}$ such that $X \perp\!\!\!\perp Y \mid \mathcal{S}$ in MAG $\mathcal{M}$ then $\mathcal{S} \in pds(X, Y, \mathcal{M})$.

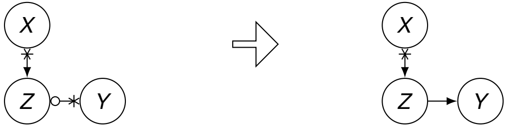


R0′: for all $X \ast\!\!-\!\!\circ\, Z \circ\!\!-\!\!\ast\, Y$ in $\mathcal{M}^*$ s.t. $Y \notin Adj(X, \mathcal{M}^*)$, if $Z \notin sepset(X, Y)$ then orient $X \ast\!\!\rightarrow Z \leftarrow\!\!\ast\, Y$ in $\mathcal{M}^*$.

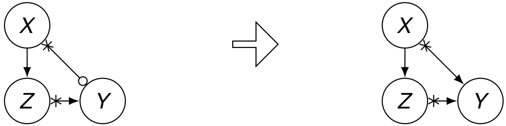
Asterix (*) represents a wildcard that denotes any of the three marks.

Orientation rules (1/4)

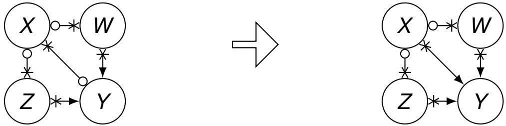
R1′:



R2′:



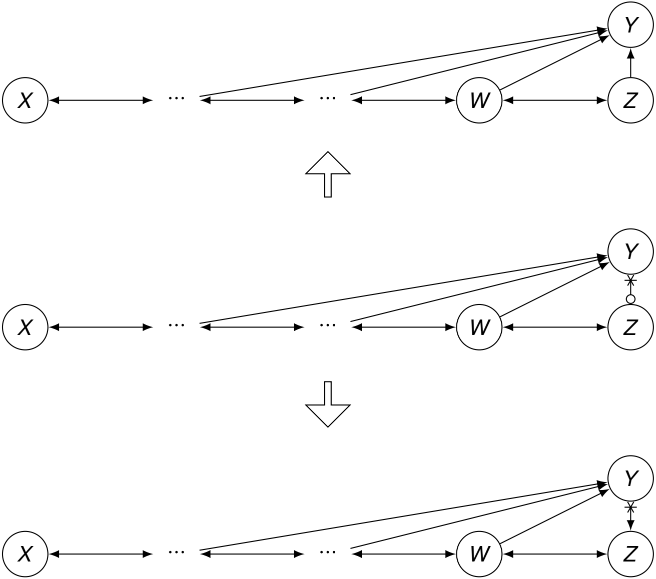
R3′:



Asterix (*) represents a wildcard that denotes any of the three marks.
R2′ also works if $X {*}\!\!\rightarrow Z \rightarrow Y$.

Orientation rules (2/4)

R4′:



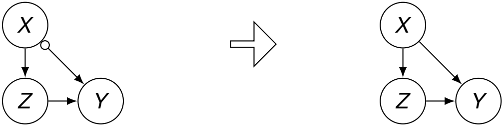
Orientation rules (3/4)

Uncovered potentially directed path: In a MIPAG, a path $\pi =< V_0, \cdots, V_n >$ is an uncovered potentially directed path if:

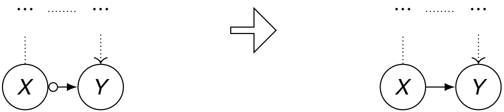
- For every $1 \leq i \leq n-1$, V_{i-1} and V_{i+1} are non adjacent;
- For every $0 \leq i \leq n-1$, the edge between V_i and V_{i+1} is not into V_i or out of V_{i+1}.

Orientation rules (4/4)

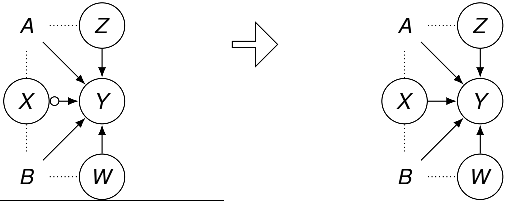
R8′:



R9′:



R10′:



R5′-R7′ are used to detect selection bias.
R8′ also works if $X \multimap Z \rightarrow Y$.
Dotted lines represents uncovered potentially directed path.

Orientation rules correctness

Pattern A pattern of a MAG $\mathcal{M}$ is a graph with the same skeleton as $\mathcal{G}$ but where only v-structures are oriented.

Orientation rules correctness

Pattern A pattern of a MAG $\mathcal{M}$ is a graph with the same skeleton as $\mathcal{G}$ but where only v-structures are oriented.

Theorem (orientation soundness) Given a pattern of some MAG, the three orientation rules R1′, R2′, R3′, R4′, R8′, R9′, R10′ are sound.
(proof on board)

Orientation rules correctness

Pattern A pattern of a MAG $\mathcal{M}$ is a graph with the same skeleton as $\mathcal{G}$ but where only v-structures are oriented.

Theorem (orientation soundness) Given a pattern of some MAG, the three orientation rules R1′, R2′, R3′, R4′, R8′, R9′, R10′ are sound.
(proof on board)

Theorem (orientation completeness) The result of recursively applying rules R1′, R2′, R3′, R4′, R8′, R9′, R10′ to a pattern of some MAG is a MIPAG.
(proof in (Zhang, 2008))

The FCI algorithm

Algorithm 3 FCI

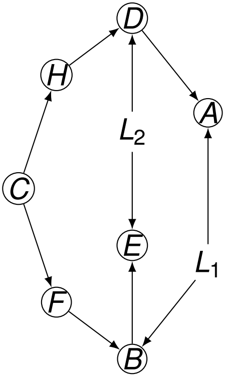
Input: $P(\mathcal{V})$

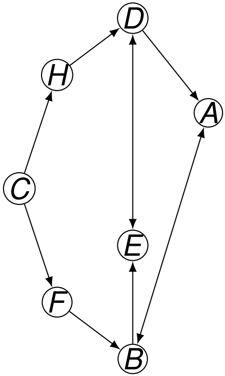
Output: MIPAG $\mathcal{M}^*$

Form the complete graph $\mathcal{M}^*$ on vertex set $\mathcal{V}$ with $\circ\!\!-\!\!\circ$ edges
Let $n = 0$
repeat
    for all $X \circ\!\!-\!\!\circ Y$ in $\mathcal{M}^*$ s.t. $|Adj(X, \mathcal{M}^*)| \geq n$ and subsets $\mathcal{S} \subseteq Adj(X, \mathcal{M}^*) \backslash \{Y\}$ s.t. $|\mathcal{S}| = n$ **do**
        if $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ **then**
            Delete edge $X \circ\!\!-\!\!\circ Y$ from $\mathcal{M}^*$
            Let $sepset(X, Y) = sepset(Y, X) = \mathcal{S}$
        end if
    end for
    Let $n = n + 1$
until for each pair of adjacent vertices (X, Y), $|Adj(X, \mathcal{M}^*) \backslash \{Y\}| \leq n$
Apply R0′
for all $X *\!\!-\!\!* Y$ in $\mathcal{M}^*$ and there exists $\mathcal{S} \in pds(X, Y, \mathcal{M}^*)$ or $\mathcal{S} \in pds(Y, X, \mathcal{M}^*)$ **do**
    if $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$ **then**
        Delete edge $X \circ\!\!-\!\!\circ Y$ from $\mathcal{M}^*$
        Let $sepset(X, Y) = sepset(Y, X) = \mathcal{S}$
    end if
end for
Reorient all edges as $\circ\!\!-\!\!\circ$ and reapply R0′
Recursively apply rules R1′-R10′ until no more edges can be oriented
Return $\mathcal{M}^*$

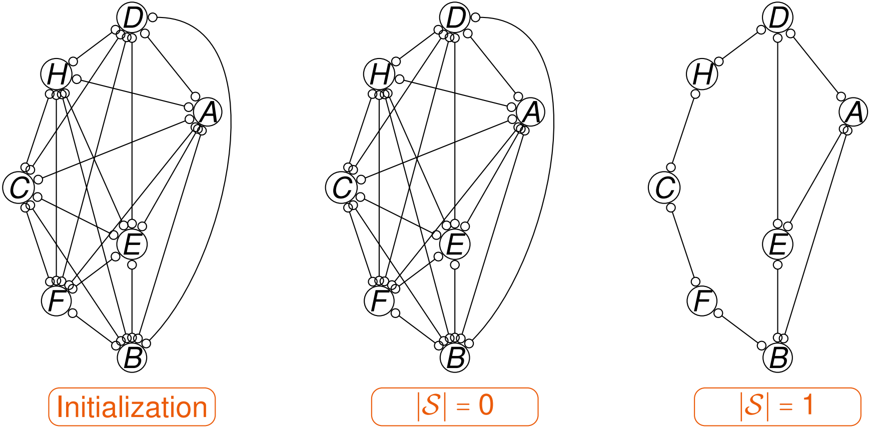
FCI in action (1/4)

- Suppose the true graph below left and its corresponding MAG below right;
- Assumptions: CMC, faithfulness.



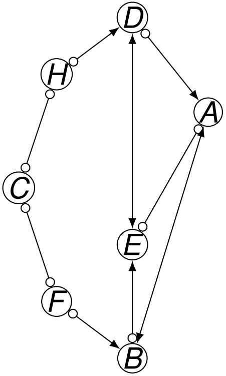


FCI in action (2/4)

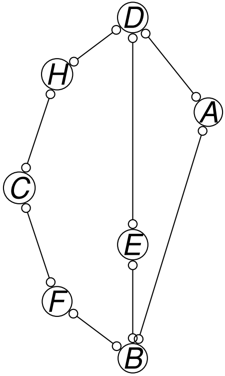


FCI in action (3/4)

Finding possible-d-sep

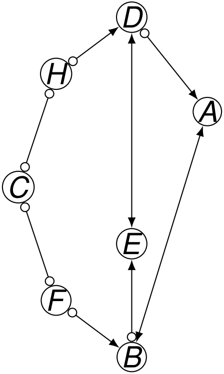


R0′



pds

FCI in action (4/4)



R0′

Correctness of FCI

Theorem (correctness) Assume the distribution $P(\mathcal{V})$ is Markov and faithful to some MAG $\mathcal{M}$ and assume that we are given perfect conditional independence information about all pairs of variables. Let $\mathcal{M}^*$ be the MIPAG of $\mathcal{M}$. The FCI algorithm returns $\mathcal{M}^*$.
(proof in (Zhang, 2008))

Computational complexity of FCI

Running time of FCI is greater than the Running time of PC:

- computing pds sets;
- testing conditional independence given all subsets of the pds sets.

Exercise 6

- Suppose the true MAG on the right;
- Assumptions: CMC, faithfulness;
- Given a compatible distribution what would be the output of the FCI algorithm?

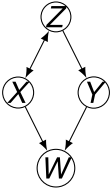


Table of content

Conditional independence tests

With finit data, SGS, PC and FCI needs a procedure for deciding whether $X \perp\!\!\!\perp_P Y \mid \mathcal{S}$.

In practice, test the null hypothesis:

$$H_0 : X \perp\!\!\!\perp_P Y \mid \mathcal{S}$$

and reject the null hypothesis if some test statistic $T(x) < \alpha$, where α is a user-specified significance threshold. That is, if we reject the null hypothesis, we keep the edge, and if we fail to reject, we remove the edge.

Examples of conditional independence tests

Tests	Assumptions
Fisher Z-transform	Linear, gaussian
χ^2 test	Multinomial discrete
Kernel-based CI test	-
Local permutation test	-

Consistency

Theorem (consistency) Assume the distribution $P(\mathcal{V})$ is Markov and faithful to some DAG $\mathcal{G}$. Let $\mathcal{G}^*$ be the CPDAG of $\mathcal{G}$ and let $\hat{\mathcal{G}}^*$ be the output of SGS, PC with some consistent conditional independence test and significative level α. Then there is a sequence of $\alpha_n \to 0 (n \to \infty)$ such that $\lim_{n\to\infty} \Pr(\hat{\mathcal{G}}^* = \mathcal{G}^*) = 1$. (proof in (Spirtes et al, 2000))

Theorem (consistency) Assume the distribution $P(\mathcal{V})$ is Markov and faithful to some DAG $\mathcal{G}$. Let $\mathcal{G}^*$ be the CPDAG of $\mathcal{G}$ and let $\hat{\mathcal{G}}^*$ be the output of SGS, PC with some consistent conditional independence test and significative level α. Then there is a sequence of $\alpha_n \to 0 (n \to \infty)$ such that $\lim_{n\to\infty} \Pr(\hat{\mathcal{G}}^* = \mathcal{G}^*) = 1$. (proof in (Spirtes et al, 2000))

Same result for FCI on MIPAG.

Exercise 7

As the significance level is lowered to 0, what would you expect to happen to the graph skeleton learned by constraint based causal discovery algorithms? As the significance level is increased to 1? Explain.

Table of content

Conclusion

- Under faithfulness and causal sufficiency constraint-based methods can discover a CPDAG (SGS, PC).
- Under faithfulness and causal sufficiency constraint-based methods can discover a MIPAG (FCI).
- Advantages:
  - Nonparametric (in principle);
  - PC and FCI are relatively scalable;
  - Lots of work on improvements.
- Drawbacks:
  - Cannot discover the entire true graph;
  - Faithfulness is not testable;
  - Cannot parallelize;
  - No confidence intervals;
  - Individual errors may propagate.

Conclusion

- Under faithfulness and causal sufficiency constraint-based methods can discover a CPDAG (SGS, PC).
- Under faithfulness and causal sufficiency constraint-based methods can discover a MIPAG (FCI).
- Advantages:
  - Nonparametric (in principle);
  - PC and FCI are relatively scalable;
  - Lots of work on improvements.
- Drawbacks:
  - Cannot discover the entire true graph;
  - Faithfulness is not testable;
  - Cannot parallelize;
  - No confidence intervals;
  - Individual errors may propagate.

Conclusion

- Under faithfulness and causal sufficiency constraint-based methods can discover a CPDAG (SGS, PC).
- Under faithfulness and causal sufficiency constraint-based methods can discover a MIPAG (FCI).
- Advantages:
  - Nonparametric (in principle);
  - PC and FCI are relatively scalable;
  - Lots of work on improvements.
- Drawbacks:
  - Cannot discover the entire true graph;
  - Faithfulness is not testable;
  - Cannot parallelize;
  - No confidence intervals;
  - Individual errors may propagate.

Conclusion

- Under faithfulness and causal sufficiency constraint-based methods can discover a CPDAG (SGS, PC).
- Under faithfulness and causal sufficiency constraint-based methods can discover a MIPAG (FCI).
- Advantages:
  - Nonparametric (in principle);
  - PC and FCI are relatively scalable;
  - Lots of work on improvements.
- Drawbacks:
  - Cannot discover the entire true graph;
  - Faithfulness is not testable;
  - Cannot parallelize;
  - No confidence intervals;
  - Individual errors may propagate.

Some extensions

- Incorporating background knowledge;
- Order independent;
- Selection bias (R5'-R7' in FCI);
- Really fast FCI;
- Time series.

Some extensions

- Incorporating background knowledge;
- Order independent;
- Selection bias (R5'-R7' in FCI);
- Really fast FCI;
- Time series.

Some extensions

- Incorporating background knowledge;
- Order independent;
- Selection bias (R5'-R7' in FCI);
- Really fast FCI;
- Time series.

Some extensions

- Incorporating background knowledge;
- Order independent;
- Selection bias (R5'-R7' in FCI);
- Really fast FCI;
- Time series.

Some extensions

- Incorporating background knowledge;
- Order independent;
- Selection bias (R5'-R7' in FCI);
- Really fast FCI;
- Time series.

References (1/3)

Direct inspirations for part 1

1. *Equivalence and Synthesis of Causal Models*, T. Verma, J. Pearl. Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence, 1990
2. *Causation, Prediction, and Search*, P. Spirtes, C. Glymour , R. Scheines. MIT Press, 2nd edition, 2000
3. *An algorithm for deciding if a set of observed independencies has a causal explanation*, T. Verma, J. Pearl. Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, 1992
4. *Causal inference and causal explanation with background knowledge*, C. Meek. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995
5. *A characterization of Markov equivalence classes for acyclic digraphs*, S. A. Andersson and D. Madigan and M. D. Perlman. Annals of Statistics, 1997
6. *Learning equivalence classes of bayesian-network structures*, D. M. Chickering. JMLR, 2002

References (2/3)

Direct inspirations for part 2

1. Unmeasured Confounding and More Recent Developments/Challenges in Causal Discovery, D. Malinsky (slides available at `http://www.dmalinsky.com/uploads/4/9/9/3/49933045/simons_part2.pdf`)
2. *On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias*, J. Zhang. Artificial Intelligence, 2008
3. *Towards Characterizing Markov Equivalence Classes for Directed Acyclic Graphs with Latent Variables*, A. Ali, T. Richardson, P. Spirtes, J. Zhang. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 2005
4. *A Polynomial Time Algorithm for Determining DAG Equivalence in the Presence of Latent Variables and Selection Bias*, P. Spirtes, T. Richardson. Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics, 1997

References (3/3)

Additional readings

1. *Weakening faithfulness: some heuristic causal discovery algorithms*, Zhalma, J. Zhang, W. Mayer. International Journal of Data Science and Analytics, 2017
2. *Order-Independent Constraint-Based Causal Structure Learning*, D. Colombo, M. Maathuis. JMLR, 2014
3. *Learning high-dimensional directed acyclic graphs with latent and selection variables*, D. Colombo, M. Maathuis, M. Kalisch, T. Richardson. The Annals of Statistics, 2011
4. *Uniform Consistency in Causal Inference*, J. Robins. Biometrika, 2003
5. *Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm*, M. Kalisch, P. Bühlmann. JMLR, 2007
6. *Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets*, J. Runge. Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence, 2020