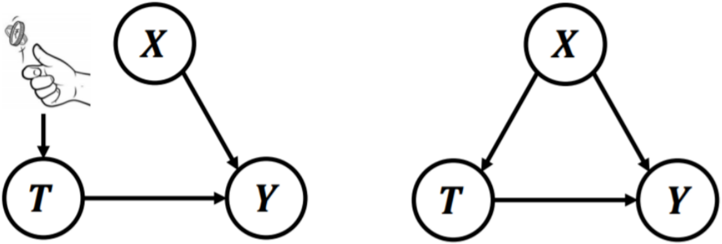
Causal discovery

Credit : Neal

Refresher

We have considered two settings

- RCTs $[T \perp\!\!\!\perp (Y(0), Y(1))]$
- Neyman Rubin model $[T \perp\!\!\!\perp (Y(0), Y(1)) | X]$



Refresher

Under these assumptions, we have addressed several questions

- Estimate the Average Treatment Effect (ATE)

$$ATE := \mathbb{E}[Y_i(1) - Y_i(0)]$$

- Taking into account Heteregeneous Treatment Effects and estimate the Conditional Average Treatment Effect (CATE)

$$CATE(x) := \mathbb{E}[Y_i(1) - Y_i(0) | X = x]$$

Refresher

Underlying assumptions in Course 1-Course 3

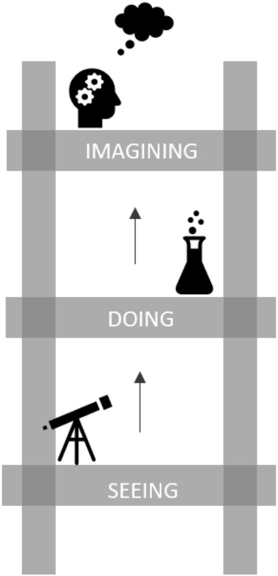
- The graph is assumed to be known
- It involves only three variables

Some natural questions

- What about more complex situations?
  Some answers in Course 4
- How can we learn the graph?
  Some answers in Course 5

Challenges and principles

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



Challenges and principles

- In general, causal discovery from observational data is not possible.
- But it is possible under additional assumptions.
- Several approaches in the litterature
  - Constraint based methods : run local tests of independence to create constraints on space of possible graphs.
  - Noise based methods : find footprints in the noise that imply causal asymmetry.
  - ...

Principles of constrain based methods

Main steps

- Find skelton
- Find v-structures
- Orient other edges using basic rules

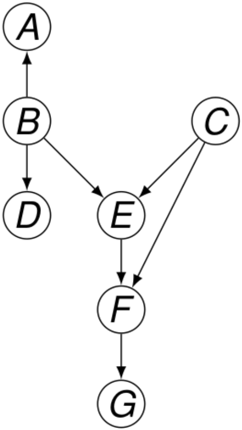
Principles of constrain based methods

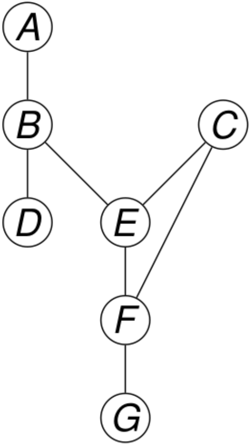
Algorithm 1 SGS

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 $\mathcal{G}^*$

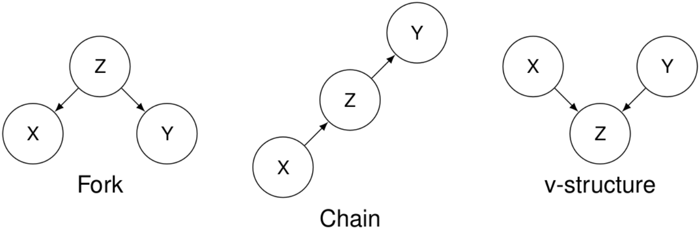
Principles of constrain based methods





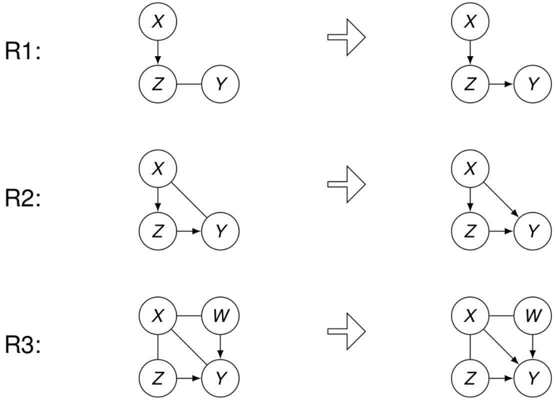
a DAG and its corresponding skeleton

Principles of constrain based mthods



Fork, chains and v-structures

Principles of constrain based methods



Basic rules

The concept of d separation

Blocked paths

A path is said to be blocked by a set of vertices Z if:

- it contains a chain $A \rightarrow B \rightarrow C$ or a fork $A \leftarrow B \rightarrow C$ and $B \in Z$, or
- it contains a collider $A \rightarrow B \leftarrow C$ such that no descendant of B is in Z

The concept of d separation

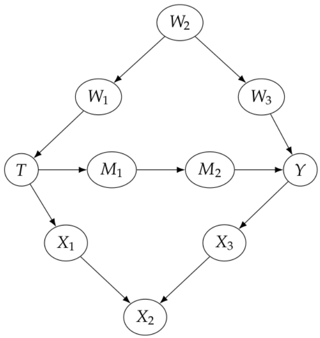
Definition

Two (sets of) nodes X and Y are d-separated by a set of nodes Z if all of the paths between (any node in) X and (any node in) Y are blocked by Z. We denote $X \perp\!\!\!\perp_G Y|Z$

Theorem

Two DAGs G_1 and G_2 have the same d-separations iff they have the same skeleton and the same v-structures.

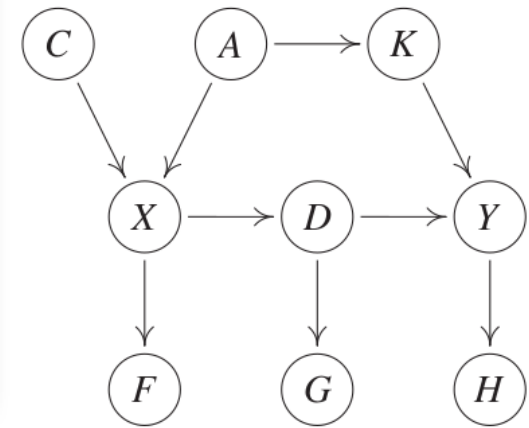
The concept of d separation



Are T and Y d separated by

- the empty set?
- $\{W_2\}$?
- $\{W_2, M_1\}$?
- $\{W_1, M_2\}$?
- $\{W_1, M_2, X_2\}$?

The concept of d separation



For this DAG : $C \perp\!\!\!\perp_G G | \{X\}$ and $C \not\perp\!\!\!\perp_G G | \{X, H\}$

The concept of d separation

Link with the conditional independency concept?

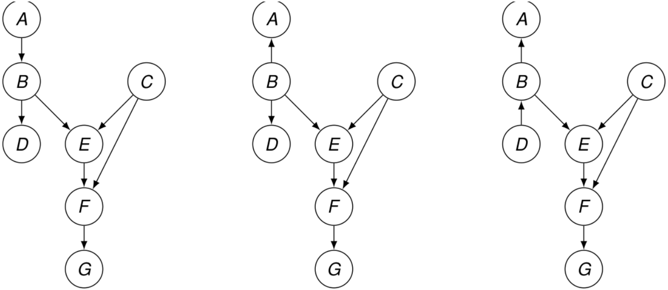
- Markov assumption : $X \perp\!\!\!\perp_G Y|Z$ then $X \perp\!\!\!\perp_{\mathbb{P}} Y|Z$
- We can assume that the converse holds, this is the faithfulness assumption

The concept of *d* separation

Markov equivalence class

- Under these two assumptions, estimate the *d* separation in a graph consists in estimating conditional dependencies
- Graphs having the same *d* separation are said to be Markov equivalent

The concept of *d* separation



The concept of *d* separation

Completed partially directed acyclic graph (CPDAG)

Let [G] be the Markov equivalence class of a DAG G. The CPDAG G^* of G is the graph:

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

The concept of *d* separation

- CPDAG coincide with Markov quivalnce classes undr two additional assumption : causal sufficiency (no latent variables) and acyclicity
- Rules R1-R3 ensure these assumptions

The PC algorithm

- PC algorithm : optimized vrsion of SGS
- Infer causal structure with the PC algorithm?
  - Infer mutual dependencies between variables : skeleton of the causal graph
  - Distinguish between causes and effects : orientation of the v-structures of the causal graph

The PC algorithm

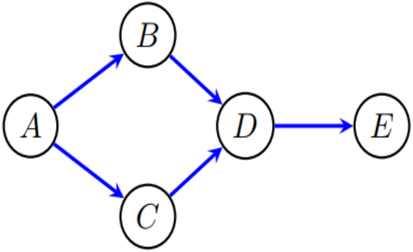
Independence tests: some examples

Type of variable	An example of independence test
Discrete	χ^2 test
Gaussian	Test based on the precision matrix
Non Gaussian continuous	Non parametric tests

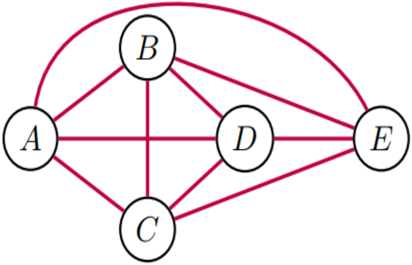
See notebook `CI.ipynb` for more details

The PC algorithm

An example



Unknown true graph

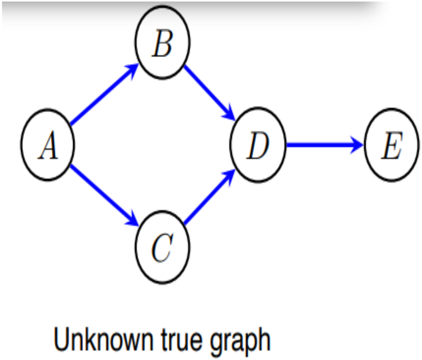


Graph after $k = 0$

Figure: The initial graph is complete

The PC algorithm

An example



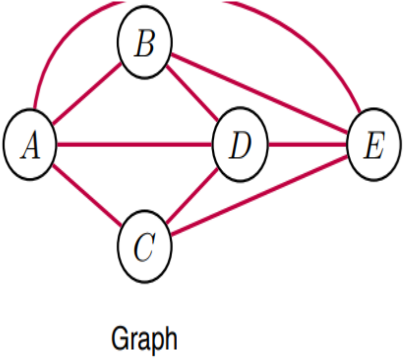


Figure: One conditions with respect to $S = \emptyset$

The PC algorithm

An example

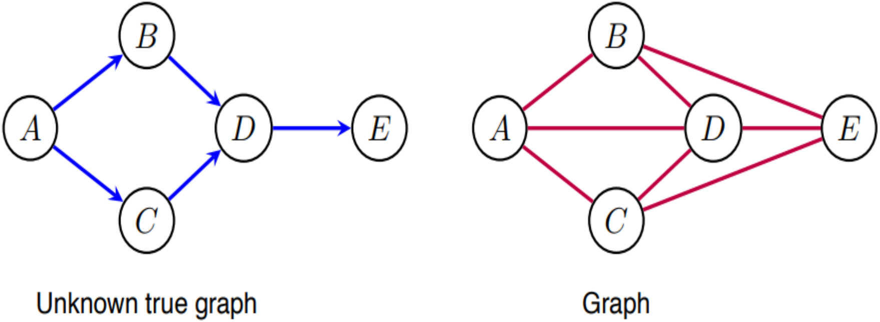


Figure: *B* and *C* are separated with respect to *A*

The PC algorithm

An example

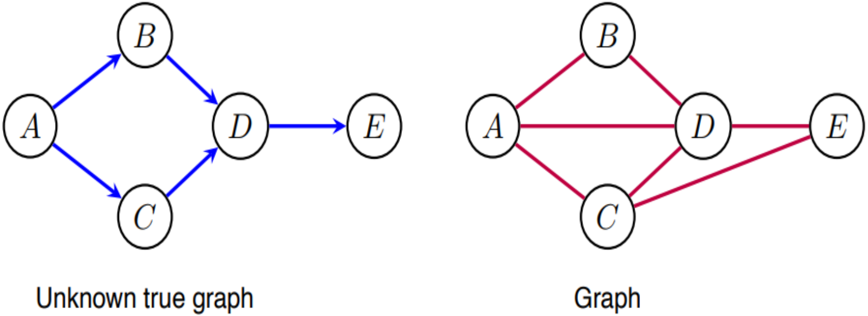


Figure: *A* and *E* are separated with respect to *D*

The PC algorithm

An example

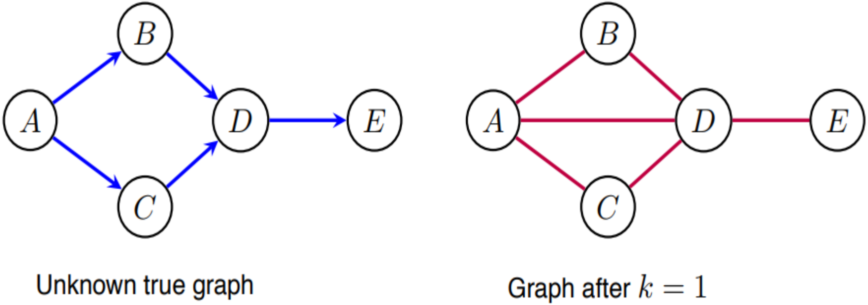
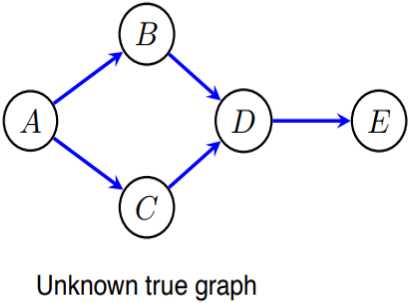


Figure: *B* and *E* are separated with respect to *D*

The PC algorithm

An example



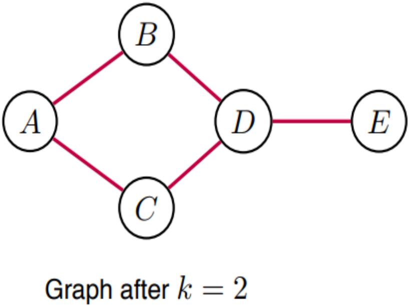


Figure: C and E are separated with respect to D

The PC algorithm

An example

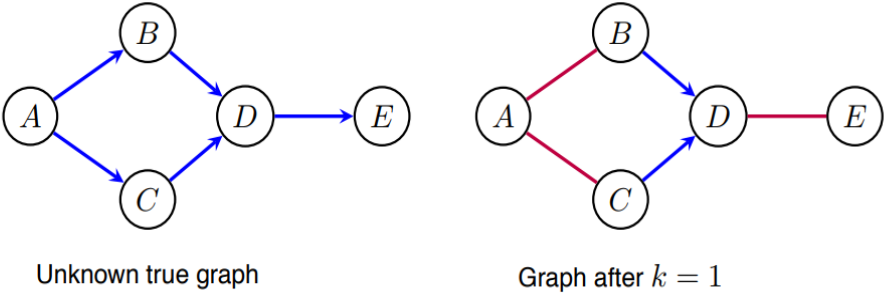


Figure: *A* and *D* are separated with respect to $\{B, C\}$

The PC algorithm

An example

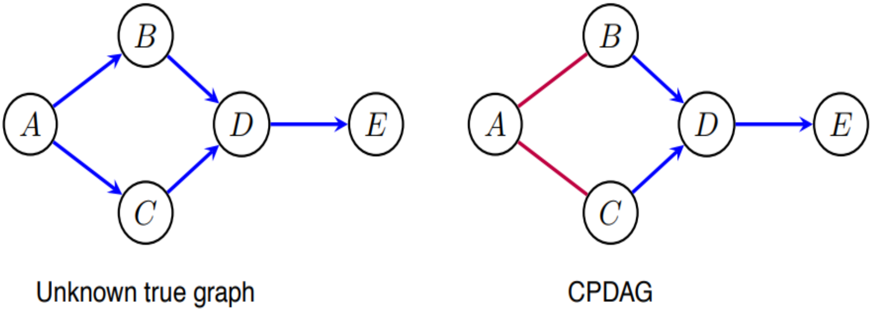


Figure: *D* is not in the set of nodes separating *B* and *C*

The PC algorithm

An example

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

The PC algorithm

An example

Algorithm 2 Extending the skeleton to a CPDAG

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

Cause or consequence?

- Can we distinguish cause from effect?
- That is distinguish between these two causal graphs

$$X \to Y$$

or

$$Y \to X$$

using observational data.

`Not always possible!`

Cause or consequence?

The example of linear structural equation [f linear]

X cause Y if there exists $a \in \mathbb{R}, \varepsilon^Y$ s.t.

$$Y = aX + \varepsilon^Y, X \perp\!\!\!\perp \varepsilon^Y.$$

Distinguish cause from consequence? [Shimizu et al., 2006]

Assume that $Y = aX + \epsilon^Y, X \perp\!\!\!\perp \epsilon^Y$ where all r.v. are continuous. Then

$$\exists b \in \mathbb{R}, \varepsilon^X \text{ s.t. } X = bY + \varepsilon^X, Y \perp\!\!\!\perp \varepsilon^X$$

iff (X, ε^X) are Gaussian random variables.

Existence of a non-linear extension of this result.

More on the Bigaussian case (1)

See notebook `noise.ipynb`

Causal model in the Bigaussian case

Let $(X, Y) \sim \mathcal{N}((0,0), \Sigma)$.

$$Y = aX + \varepsilon^Y, \varepsilon^Y \perp X \text{ where } X, \varepsilon^Y \sim \mathcal{N}(0, \sigma) \text{ with } a = C_{X,Y}/V_X$$

- Caveat : $X = \frac{1}{a}(Y - \epsilon^Y)$ but $\varepsilon^Y \not\perp Y$: ε^Y and Y not **independants.** Non causal model.
- There exists (b, ϵ^X) s.t.

$$X = bY + \epsilon^X, \epsilon^X \perp X,$$

with $b = \frac{aV_X}{a^2 V_X + V_{\epsilon_Y}}$

More on the Bigaussian case (2)

Simulations : sample size $n = 2000$.

	1	2
1	10.00	3.00
2	3.00	2.00

Table: Σ

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0234	0.0336	0.70	0.4868
X	0.3081	0.0110	28.03	0.0000

Table: Residual standard error: 1.063 on 998 degrees of freedom. Multiple R-squared: 0.4405, Adjusted R-squared: 0.44 ; F-statistic: 785.8 on 1 and 998 DF, p-value: $< 2.2e-16$

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.0052	0.0725	-0.07	0.9430
Y	1.4300	0.0510	28.03	0.0000

Table: Residual standard error: 2.291 on 998 degrees of freedom; Multiple R-squared: 0.4405, Adjusted R-squared: 0.44 ;F-statistic: 785.8 on 1 and 998 DF, p-value: $< 2.2e-16$

More on the Bigaussian case (3)

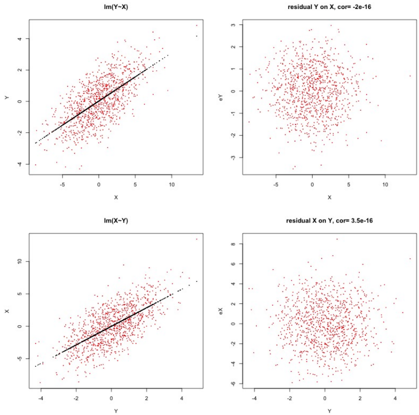
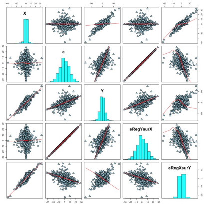


Figure: In Gaussian case, we cannot distinguish cause from effect

Non Gaussian example

Model

$$Z \sim \mathcal{N}(0,1),\ X = Z^3,\ \epsilon^Y \sim \mathcal{N}(0,9) \text{ and } Y = 2X + \epsilon^Y$$



Non Gaussian example

- In the NON Gaussian setting cause can be distinguished from effect
- Independence tested and accepted for couples (X, ϵ^Y), $(X, eRegYsurX)$ but not (Y, ϵ^Y), ni $(Y, eRegXsurY)$.
- X has an influence on Y but Y does not influence X.

Noise based algorithm

Theorem (LiNGAM)

Assume a linear SCM with graph $G = (V, E)$ and a compatible distribution P(V) such that or all $Y \in V$

$$Y = \sum_{X \in Pa(Y)} a_{xy} X + \xi_Y$$

where all ξ_Y are jointly independent and non-Gaussian distributed. Additionally, we require that for all $Y \in V$, $X \in Pa(Y)$, $axy \neq 0$. Then, the graph G is identifiable from P(V).

Noise based algorithm

LINGAM

Algorithm 1 LiNGAM

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

Form an empty graph $\mathcal{G}$ on vertex set $\mathcal{V} = \{X_1, \cdots, X_p\}$
Let $S = \{1, \cdots, p\}$ and $\mathcal{T} = []$
repeat
  $H = []$
  for $i \in S$ **do**
    for $j \in S \backslash \{i\}$ **do**
      $\hat{\xi}_{ij} = X_j - \frac{cov(X_i, X_j)}{var(X_i)} X_i$
    end for
    $h = \sum_{j \in S \backslash \{i\}} \mathcal{I}(X_i, \hat{\xi}_{ij})$
    $H = [H, h]$
  end for
  $i^* = arg \min_{i \in S} H$
  $S = S \backslash \{i^*\}$
  $\mathcal{T} = [\mathcal{T}, i^*]$
  $\forall j \in S, X_j = \hat{\xi}_{i^* j}$
until $|S| = 0$
Append$(\mathcal{T}, S_0)$
Construct a strictly lower triangular matrix by following the order in $\mathcal{T}$, and estimate the connection strengths $a_{i,j}$ by using some conventional covariance-based regression.
if $a_{i,j} > 0$ **then**
  Add $X_i \rightarrow X_j$ to $\mathcal{G}$
end if
Return $\mathcal{G}$

Noise based algorithm

Theorem (Anm)

Assume a linear SCM with graph $G = (V, E)$ and a compatible distribution P(V) such that or all $Y \in V$

$$Y = f((X \in Pa(Y)) + \xi_Y$$

where all ξ_Y are jointly independent. Then, the graph G is identifiable from P(V).

Noise based algorithm

ANM

Algorithm 2 ANM

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

Output: $\mathcal{G}$

Form an empty graph $\mathcal{G}$ on vertex set $\mathcal{V} = \{X_1, \cdots, X_p\}$

Let $S = \{1, \cdots, p\}$ and $\mathcal{T} = []$

repeat

  $H = []$

  for $j \in S$ **do**

    $\hat{f}_{.j}$: Regress X^j on $\{X_i\}_{i \in S \setminus \{j\}}$

    $\hat{\xi}_{.j} = X_j - \hat{f}_{.j}(X_i)$

    $h = \hat{I}(\{X_i\}_{i \in S \setminus \{j\}}, \xi_{.j})$

    $H = [H, h]$

  end for

  $i^* = arg \min_{i \in S} H$

  $S = S \setminus \{i^*\}$

  $\mathcal{T} = [i^*, \mathcal{T}]$

until $|S| = 0$

for $j \in \{2, \cdots, p\}$ **do**

  for $i \in \{\mathcal{T}_1, \cdots, \mathcal{T}_{j-1}\}$ **do**

    $\hat{f}_{.j}$: Regress X^j on $\{X_k\}_{k \in \{\mathcal{T}_1, \cdots, \mathcal{T}_{j-1}\} \setminus \{i\}}$

    $\hat{\xi}_{.j} = X_j - \hat{f}_{.j}(X_i)$

    if $\{X_k\}_{k \in \{\mathcal{T}_1, \cdots, \mathcal{T}_{j-1}\} \setminus \{i\}} \not\perp\!\!\!\perp_P \xi_{.j}$ **then**

      Add $X_i \rightarrow X_j$ to $\mathcal{G}$

    end if

  end for

end for

Return $\mathcal{G}$