Section 3

Learning of Bayesian Networks

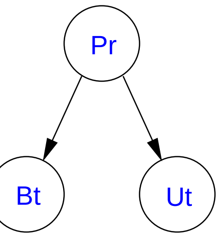
Learning probabilities from a database

We have:

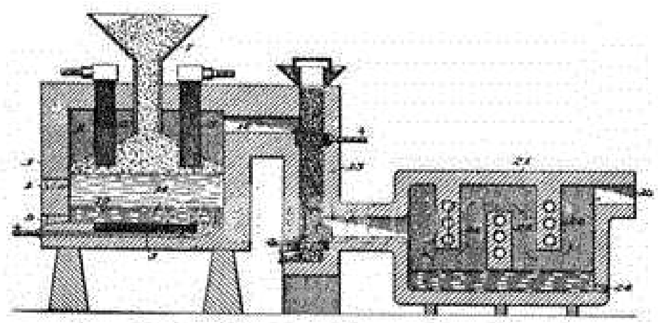
- A Bayesian network structure.
- A database of cases over (some of) the variables.

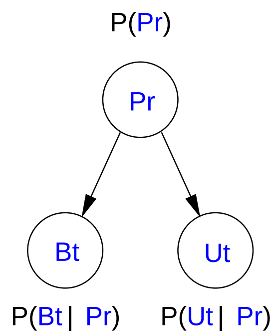
We want:

- A Bayesian network model (with probabilities) representing the database.



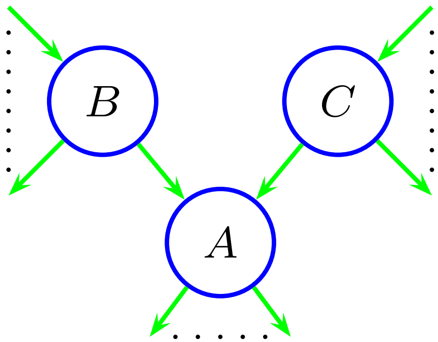
Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?





Complete data: maximum likelihood estimation

You get a maximum likelihood estimate as the fraction of counts over the total number of counts.



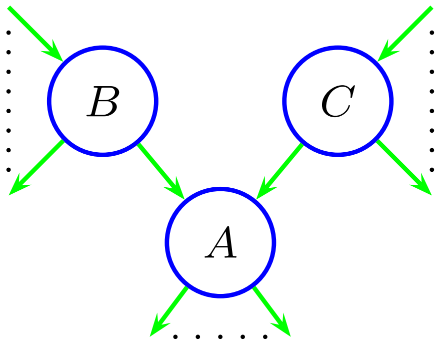
We want $P(A = a \mid B = b, C = c)$

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) =$$

Complete data: maximum likelihood estimation

You get a maximum likelihood estimate as the fraction of counts over the total number of counts.



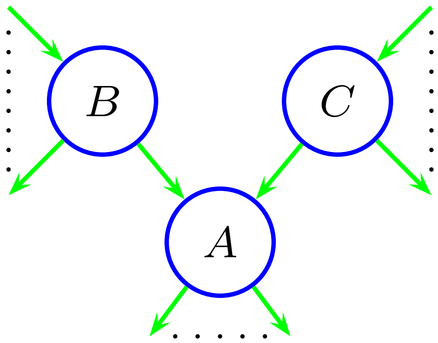
We want $P(A = a \mid B = b, C = c)$

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) = \frac{\hat{P}(A = a, B = b, C = c)}{\hat{P}(B = b, C = c)}$$

Complete data: maximum likelihood estimation

You get a maximum likelihood estimate as the fraction of counts over the total number of counts.



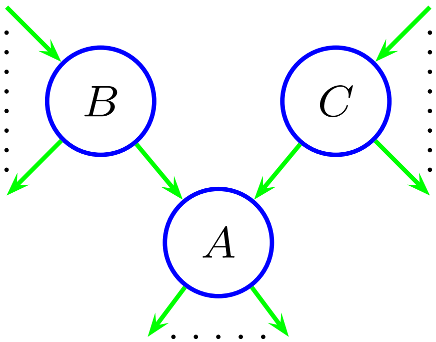
We want $P(A = a \mid B = b, C = c)$

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) = \frac{\hat{P}(A = a, B = b, C = c)}{\hat{P}(B = b, C = c)} = \frac{\left[\frac{N(A=a, B=b, C=c)}{N}\right]}{\left[\frac{N(B=b, C=c)}{N}\right]}$$

Complete data: maximum likelihood estimation

You get a maximum likelihood estimate as the fraction of counts over the total number of counts.



We want $P(A = a \mid B = b, C = c)$

To find the maximum likelihood estimate $\hat{P}(A = a \mid B = b, C = c)$ we simply calculate:

$$\hat{P}(A = a \mid B = b, C = c) = \frac{\hat{P}(A = a, B = b, C = c)}{\hat{P}(B = b, C = c)} = \frac{\left[\frac{N(A=a, B=b, C=c)}{N}\right]}{\left[\frac{N(B=b, C=c)}{N}\right]}$$

$$= \frac{N(A = a, B = b, C = c)}{N(B = b, C = c)}.$$

So we have a simple counting problem!

Complete data: maximum likelihood estimation

Unfortunately, maximum likelihood estimation has a drawback:

		Last three letters							
		aaa	aab	aba	abb	baa	bba	bab	bbb
First two letters	aa	2	2	2	2	5	7	5	7
	ab	3	4	4	4	1	2	0	2
	ba	0	1	0	0	3	5	3	5
	bb	5	6	6	6	2	2	2	2

By using this table to estimate e.g. $P(T_1 = b, T_2 = a, T_3 = T_4 = T_5 = a)$ we get:

$$\hat{P}(T_1 = b, T_2 = a, T_3 = T_4 = T_5 = a) = \frac{N(T_1 = b, T_2 = a, T_3 = T_4 = T_5 = a)}{N} = 0$$

This is not reliable!

Complete data: maximum likelihood estimation

Bayesian fix: an even prior distribution corresponds to adding a virtual count of 1:

		Last three letters							
		aaa	aab	aba	abb	baa	bba	bab	bbb
First two letters	aa	2	2	2	2	5	7	5	7
	ab	3	4	4	4	1	2	0	2
	ba	0	1	0	0	3	5	3	5
	bb	5	6	6	6	2	2	2	2

From this table we get:

		T_1	
		a	b
T_2	a	32	17
	b	20	31

$\Rightarrow$

		T_1	
		a	b
T_2	a	$32+1$	$17+1$
	b	$20+1$	$31+1$

$\Rightarrow$

		T_1	
		a	b
T_2	a	$(\frac{33}{54})$	$(\frac{18}{50})$
	b	$(\frac{21}{54})$	$(\frac{32}{50})$

$N(T_1, T_2)$ $\qquad$ $N'(T_1, T_2)$ $\qquad$ $P(T_2 \mid T_1) = \frac{N'(T_1, T_2)}{N'(T_1)}$

Incomplete data

How do we handle cases with missing values:

- Faulty sensor readings.
- Values have been intentionally removed.
- Some variables may be unobservable.

$\boxed{\text{If you do not exploit cases with missing values, you may get misleading results}}$

How is the data missing?

We need to take into account how the data is missing:

Missing completely at random The probability that a value is missing is independent of both the observed and unobserved values.

Missing at random The probability that a value is missing depends only on the observed values.

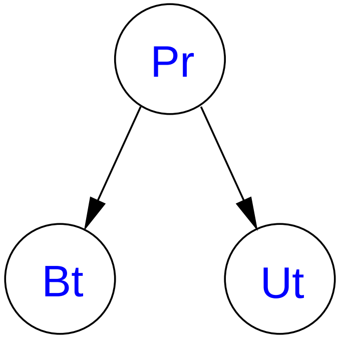
Non-ignorable Neither MAR nor MCAR.

Examples:

- MCAR: A monitoring system that is not completely stable and where some sensor values are not stored properly.
- MAR: A database containing the results of two tests, where the second test has only performed (as a "backup test") when the result of the first test was negative.
- Non-Ign: An exit poll, where an extreme right-wing party is running for parliament.

The EM algorithm (requires MAR)

$P_0(Pr) = (0.5, 0.5)$
$P_0(Ut = \text{pos} \,|\, Pr) = (0.5, 0.5)$
$P_0(Bt = \text{pos} \,|\, Pr) = (0.5, 0.5)$

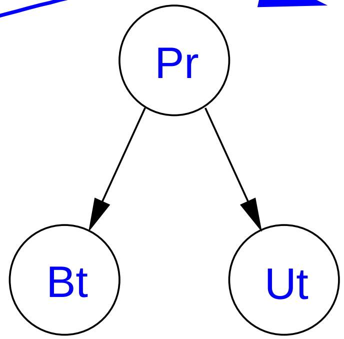


Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm (requires MAR)

E-step 1

$$P_0(Pr) = (0.5, 0.5)$$
$$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$$
$$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$$



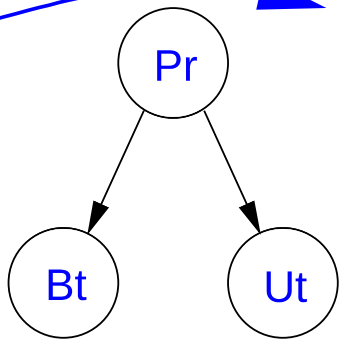
$$\mathbb{E}_0[N(Pr)] = (4, 1)$$
$$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (0.5 + 1 + 0.5 + 0 + 0.25, \\ 0.5 + 0 + 0 + 0 + 0.25)$$
$$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (0.5 + 0 + 1 + 1 + 0 \\ , 0.5 + 0 + 0 + 0 + 0)$$

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm (requires MAR)

E-step 1

$P_0(Pr) = (0.5, 0.5)$
$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$
$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$



$\mathbb{E}_0[N(Pr)] = (4, 1)$
$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (2.25, 0.75)$
$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (2.5, 0.5)$

M-step 2

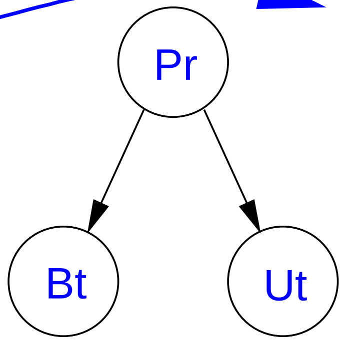
$P_1(Pr) = (\frac{4}{5}, \frac{1}{5})$
$P_1(Ut = \text{pos} \mid Pr) = (\frac{2.25}{4}, \frac{0.75}{1})$
$P_1(Bt = \text{pos} \mid Pr) = (\frac{2.5}{4}, \frac{0.5}{1})$

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm (requires MAR)

E-step 1

$P_0(Pr) = (0.5, 0.5)$
$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$
$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$

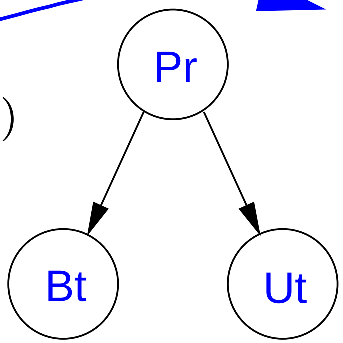


$\mathbb{E}_0[N(Pr)] = (4, 1)$
$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (2.25, 0.75)$
$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (2.5, 0.5)$

M-step 2

E-step 3

$P_1(Pr) = (\frac{4}{5}, \frac{1}{5})$
$P_1(Ut = \text{pos} \mid Pr) = (\frac{2.25}{4}, \frac{0.75}{1})$
$P_1(Bt = \text{pos} \mid Pr) = (\frac{2.5}{4}, \frac{0.5}{1})$



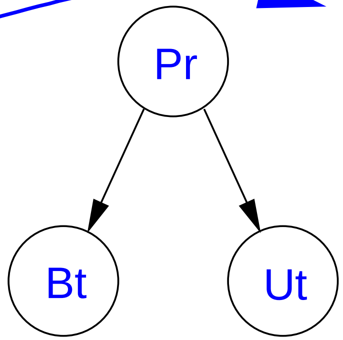
$\mathbb{E}_1[N(Pr)] = (\qquad , \qquad)$
$\mathbb{E}_1[N(Ut = \text{pos}, Pr)] = (\qquad , \qquad)$
$\mathbb{E}_1[N(Bt = \text{pos}, Pr)] = (\qquad , \qquad)$

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm (requires MAR)

$P_0(Pr) = (0.5, 0.5)$
$P_0(Ut = \text{pos} \mid Pr) = (0.5, 0.5)$
$P_0(Bt = \text{pos} \mid Pr) = (0.5, 0.5)$

E-step 1

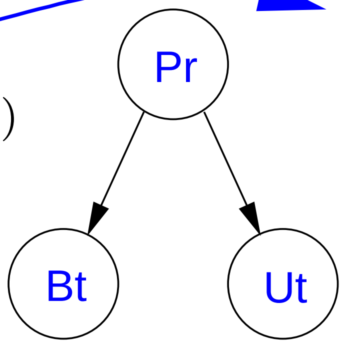


$\mathbb{E}_0[N(Pr)] = (4, 1)$
$\mathbb{E}_0[N(Ut = \text{pos}, Pr)] = (2.25, 0.75)$
$\mathbb{E}_0[N(Bt = \text{pos}, Pr)] = (2.5, 0.5)$

M-step 2

$P_1(Pr) = (\frac{4}{5}, \frac{1}{5})$
$P_1(Ut = \text{pos} \mid Pr) = (\frac{2.25}{4}, \frac{0.75}{1})$
$P_1(Bt = \text{pos} \mid Pr) = (\frac{2.5}{4}, \frac{0.5}{1})$

E-step 3



$\mathbb{E}_1[N(Pr)] = (\quad , \quad)$
$\mathbb{E}_1[N(Ut = \text{pos}, Pr)] = (\quad , \quad)$
$\mathbb{E}_1[N(Bt = \text{pos}, Pr)] = (\quad , \quad)$

M-step 4

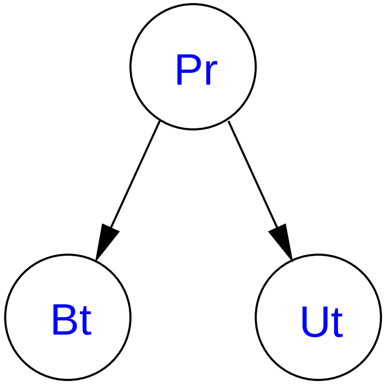
$P_2(Pr) = (\cdot, \cdot)$
$P_2(Ut = \text{pos} \mid Pr) = (\cdot, \cdot)$
$P_2(Bt = \text{pos} \mid Pr) = (\cdot, \cdot)$

$\vdots$

Until convergence

Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

The EM algorithm in general



Cases	Pr	Bt	Ut
1.	?	pos	pos
2.	yes	neg	pos
3.	yes	pos	?
4.	yes	pos	neg
5.	?	neg	?

1. Let $\boldsymbol{\theta}^0 = \{\theta_{ijk}\}$ be some start estimates ($P(X_i = j \mid \mathrm{pa}(X_i = k) = \theta_{ijk}$).
2. Repeat until convergence:

E-step: For each variable X_i calculate the table of expected counts:

$$\mathbb{E}_{\boldsymbol{\theta}^t}[N(X_i, \mathrm{pa}(X_i) \mid \mathcal{D}] = \sum_{\mathbf{d} \in \mathcal{D}} P(X_i, \mathrm{pa}(X_i) \mid \mathbf{d}, \boldsymbol{\theta}^t).$$

$P(X_i, \mathrm{pa}(X_i) \mid \mathbf{d}, \boldsymbol{\theta}^t)$ is achieved from the BN with parameters $\boldsymbol{\theta}^t$.

M-step: Use the expected counts as if they were actual counts:

$$\hat{\theta}_{ijk} = \frac{\mathbb{E}_{\boldsymbol{\theta}^i}[N(X_i = k, \mathrm{pa}(X_i) = j \mid \mathcal{D}]}{\sum_{k=1}^{|\mathrm{sp}(X_i)|} \mathbb{E}_{\boldsymbol{\theta}^i}[N(X_i = k, \mathrm{pa}(X_i) = j \mid \mathcal{D}]}.$$

Learning the structure of Bayesian networks

Some agent produces samples $\mathcal{D}$ of cases from a Bayesian network M over the universe $\mathcal{U}$.

➤ These cases are handed over to you, and you should now reconstruct M from the cases.

Assumptions:

➤ The sample is fair ($P_{\mathcal{D}}(\mathcal{U})$ reflects the distribution determined by M).

➤ All links in M are essential.

A naïve procedure:

- For each Bayesian network structure N:
  - Calculate the distance between $P_N(\mathcal{U})$ and $P_{\mathcal{D}}(\mathcal{U})$ (e.g. Kullback-Leibler divergence).
- Return the network N that minimizes the distance, and where all links are essential.

But this is hardly feasible!

The space of network structures is huge!

The number of DAG structures (as a function of the number of nodes):

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \frac{n!}{(n-i)!i!} 2^{i(n-i)} f(n-i).$$

Some example calculations:

Nodes	Number of DAGs	Nodes	Number of DAGs
1	1	13	$1.9 \cdot 10^{31}$
2	3	14	$1.4 \cdot 10^{36}$
3	25	15	$2.4 \cdot 10^{41}$
4	543	16	$8.4 \cdot 10^{46}$
5	29281	17	$6.3 \cdot 10^{52}$
6	$3.8 \cdot 10^{6}$	18	$9.9 \cdot 10^{58}$
7	$1.1 \cdot 10^{9}$	19	$3.3 \cdot 10^{65}$
8	$7.8 \cdot 10^{11}$	20	$2.35 \cdot 10^{72}$
9	$1.2 \cdot 10^{15}$	21	$3.5 \cdot 10^{79}$
10	$4.2 \cdot 10^{18}$	22	$1.1 \cdot 10^{87}$
11	$3.2 \cdot 10^{22}$	23	$7.0 \cdot 10^{94}$
12	$5.2 \cdot 10^{26}$	24	$9.4 \cdot 10^{102}$

Two approaches to structural learning

Score based learning:

- Produces a series of candidate structures.
- Returns the structure with highest score.

Constraint based learning:

- Establishes a set of conditional independence statements for the data.
- Builds a structure with d-separation properties corresponding to the independence statements found.

Constraint based learning

Some notation:

- To denote that A is conditionally independent of B given $\mathcal{X}$ in the database we shall use

$$I(A, B, \mathcal{X}).$$

Some assumptions:

- The database is a faithful sample from a Bayesian network M: A and B are d-separated given $\mathcal{X}$ in M if and only if $I(A, B, \mathcal{X})$.
- We have an oracle that correctly answers questions of the type:

"Is $I(A, B, \mathcal{X})$?"

The algorithm: Use the oracle's answers to first establish a skeleton of a Bayesian network:

- The skeleton is the undirected graph obtained by removing directions on the arcs.

Next, when the skeleton is found we then start looking for the directions on the arcs.

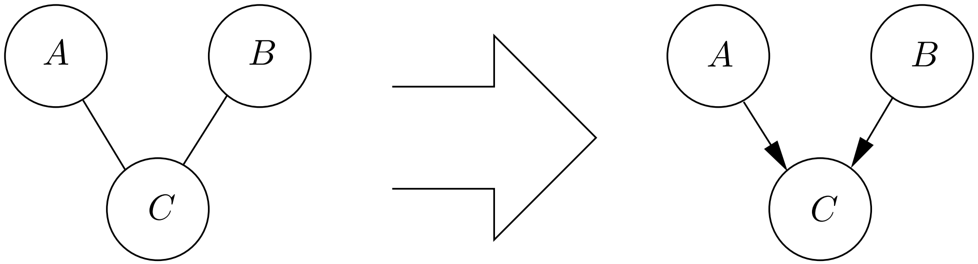
Finding the skeleton

The idea: if there is a link between A and B in M then they cannot be d-separated, and as the data is faithful it can be checked by asking questions to the oracle:

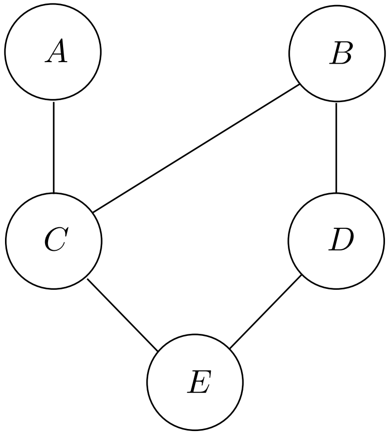
- The link $A - B$ is part of the skeleton if and only if $\neg I(A, B, \mathcal{X})$, for all $\mathcal{X}$.

Setting the directions on the links I

Rule 1: If you have three nodes, A, B, C such that $A - C$ and $B - C$, but not $A - B$, then introduce the v-structure $A \rightarrow C \leftarrow B$ if there exists an $\mathcal{X}$ (possibly empty) such that $I(A, B, \mathcal{X})$ and $C \notin \mathcal{X}$.

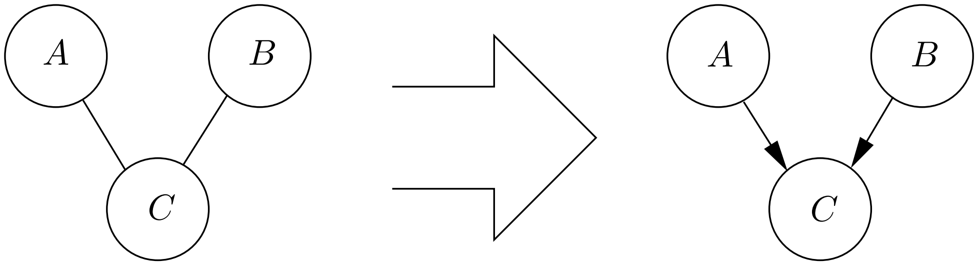


Example: Assume that we get the independencies $I(A, B)$, $I(A, B, D)$, $I(A, D)$, $I(A, D, B)$, $I(A, D, \{B, C\})$, $I(A, D, \{B, C, E\})$, $I(C, D, B)$, $I(C, D, \{A, B\})$, $I(B, E, \{C, D\})$, $I(B, E, \{C, D, A\})$, $I(A, E, \{C, D\})$ and $I(A, E, \{C, D, B\})$.

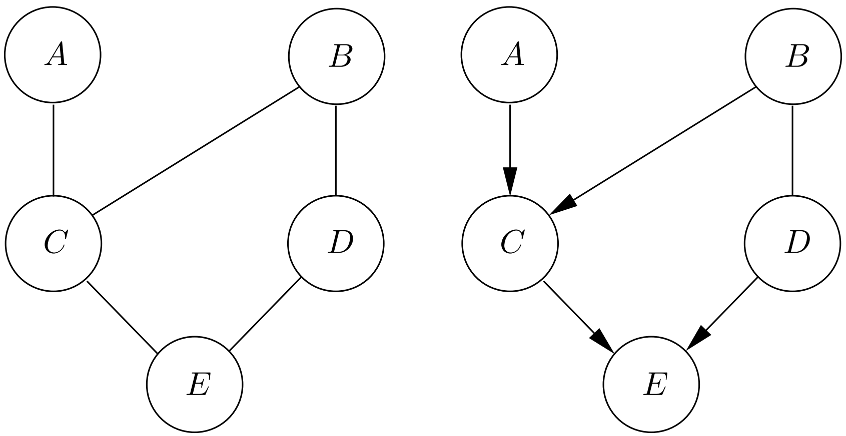


Setting the directions on the links I

Rule 1: If you have three nodes, A, B, C such that $A - C$ and $B - C$, but not $A - B$, then introduce the v-structure $A \rightarrow C \leftarrow B$ if there exists an $\mathcal{X}$ (possibly empty) such that $I(A, B, \mathcal{X})$ and $C \notin \mathcal{X}$.



Example: Assume that we get the independencies $I(A, B)$, $I(A, B, D)$, $I(A, D)$, $I(A, D, B)$, $I(A, D, \{B, C\})$, $I(A, D, \{B, C, E\})$, $I(C, D, B)$, $I(C, D, \{A, B\})$, $I(B, E, \{C, D\})$, $I(B, E, \{C, D, A\})$, $I(A, E, \{C, D\})$ and $I(A, E, \{C, D, B\})$.



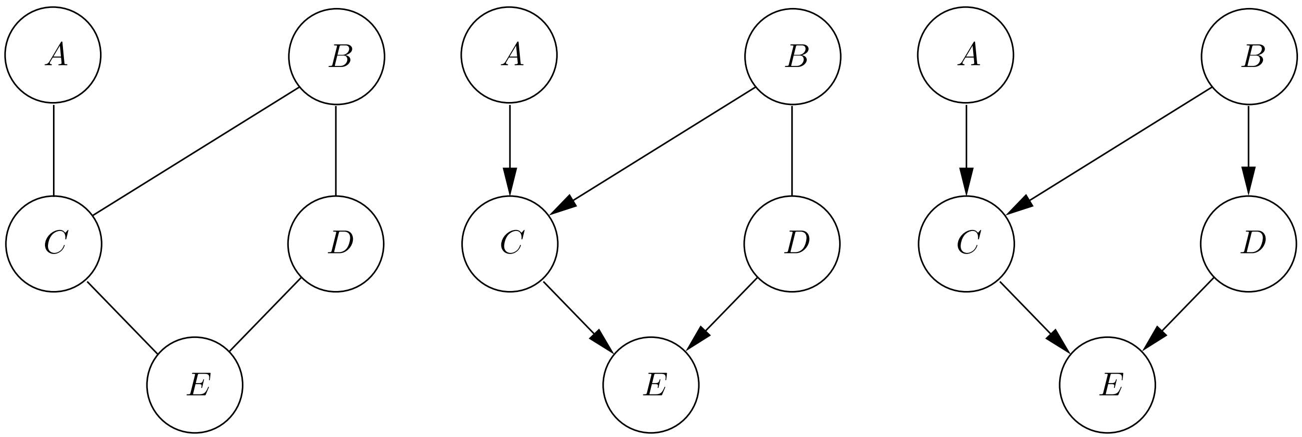
Setting the directions on the links II

Rule 2 [Avoid new v-structures]: When Rule 1 has been exhausted, and you have $A \to C - B$ (and no link between A and B), then direct $C \to B$.

Rule 3 [Avoid cycles]: If $A \to B$ introduces a directed cycle in the graph, then do $A \leftarrow B$

Rule 4 [Choose randomly]: If none of the rules 1-3 can be applied anywhere in the graph, choose an undirected link and give it an arbitrary direction.

Example:



Skeleton Rule 1 Rule 4

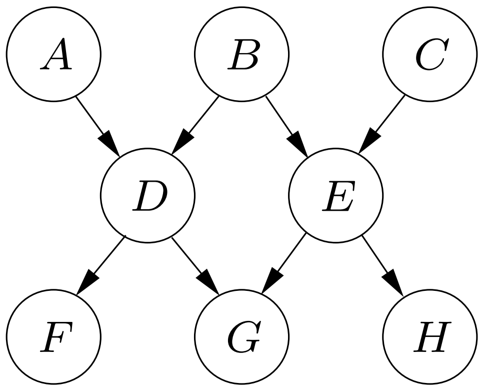
From independence tests to skeleton

Until now, we have assumed that all questions of the form "Is $I(A, B, \mathcal{X})$?" can be answered (allowing us to establish the skeleton). However, questions come at a price, and we would like to ask as few questions as possible.

To reduce the number of questions we exploit the following property:

Theorem: The nodes A and B are not linked if and only if $I(A, B, \mathrm{pa}(A))$ or $I(A, B, \mathrm{pa}(B))$.

It is sufficient to ask questions $I(A, B, \mathcal{X})$, where $\mathcal{X}$ is a subset of A's or B's neighbors.



An active path from A to G must go through a parent of G.

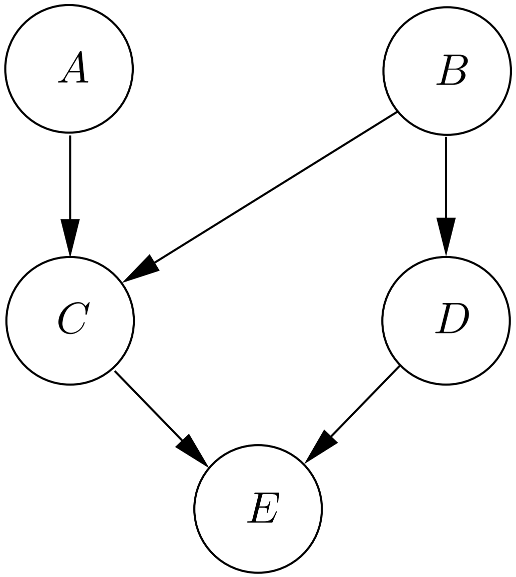
The PC algorithm

The PC algorithm:

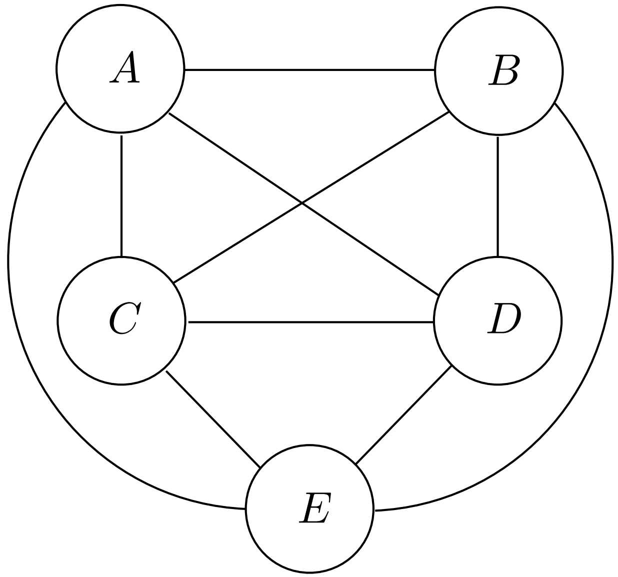
1. Start with the complete graph;
2. $i := 0$;
3. **while** a node has at least $i + 1$ neighbors
   - **for all** nodes A with at least $i + 1$ neighbors
     - **for all** neighbors B of A
       - **for all** neighbor sets $\mathcal{X}$ such that $|\mathcal{X}| = i$ and $\mathcal{X} \subseteq (\mathrm{nb}(A) \setminus \{B\})$
         - **if** $I(A, B, \mathcal{X})$ **then** remove the link $A - B$ and store "$I(A, B, \mathcal{X})$"
   - $i := i + 1$

Example

We start with the complete graph and ask the questions $I(A,B)$?, $I(A,C)$?, $I(A,D)$?, $I(A,E)$?, $I(B,C)$?, $I(B,D)$?, $I(B,E)$?, $I(C,D)$?, $I(C,E)$?, $I(D,E)$?



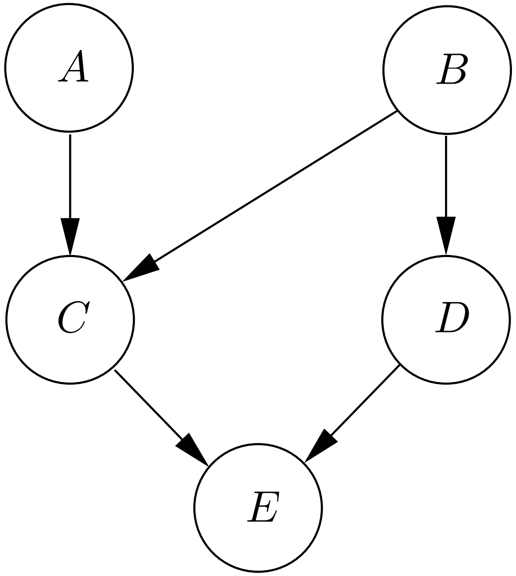
The original model



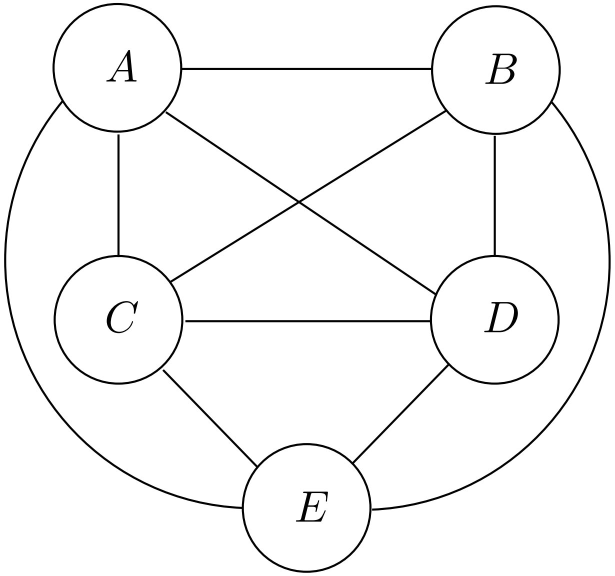
The complete graph

Example

We start with the complete graph and ask the questions $I(A, B)$?, $I(A, C)$?, $I(A, D)$?, $I(A, E)$?, $I(B, C)$?, $I(B, D)$?, $I(B, E)$?, $I(C, D)$?, $I(C, E)$?, $I(D, E)$?.



The original model



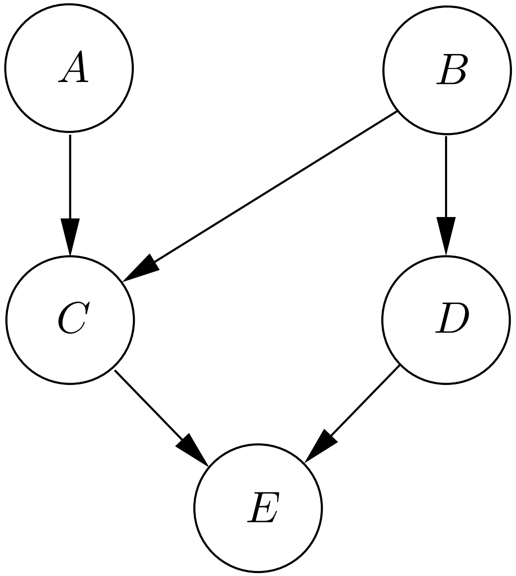
The complete graph

We get a "yes" for $I(A, B)$? and $I(A, D)$?:

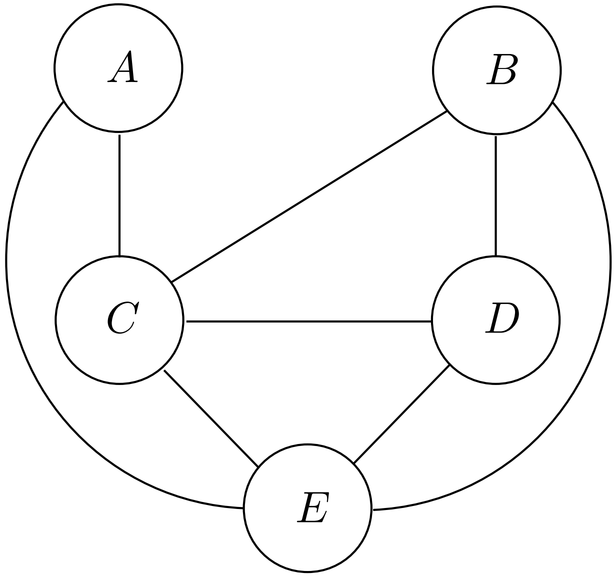
- the links $A - B$ and $A - D$ are therefore removed.

Example

We now condition on one variable and ask the questions $I(A, C, E)?$, $I(A, E, C)?$, $I(B, C, D)?$, $I(B, C, E)?$, $I(B, D, C)?$, $I(B, D, E)?$, $I(B, E, C)?$, $I(B, E, D)?$, $I(C, B, A)?$, $\ldots$,$I(C, D, A)?$, $I(C, D, B)?$.



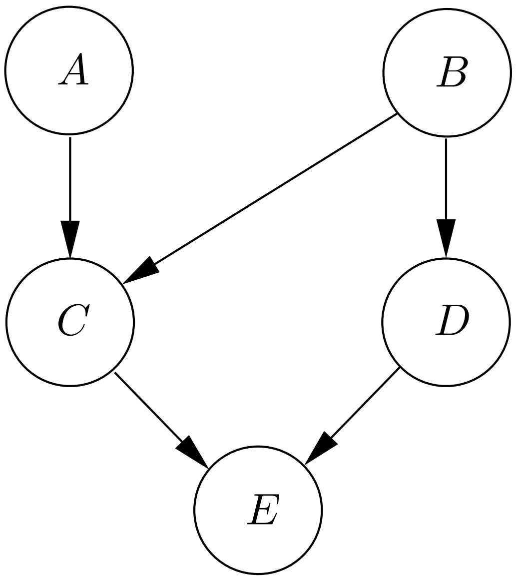
The original model



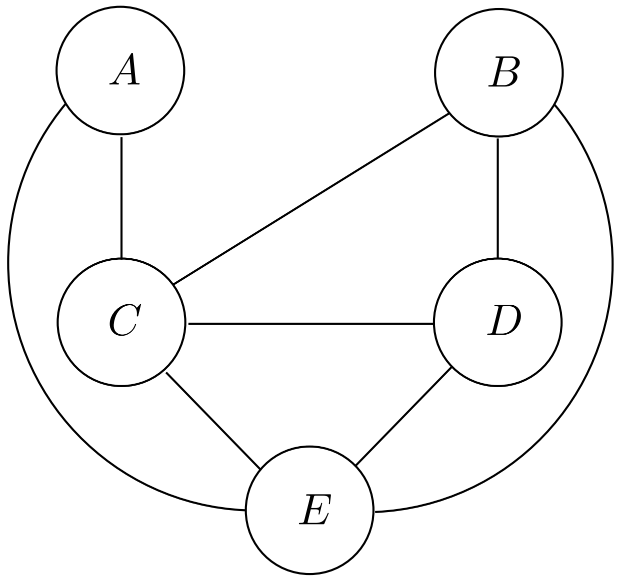
After one iteration

Example

We now condition on one variable and ask the questions $I(A, C, E)?$, $I(A, E, C)?$, $I(B, C, D)?$, $I(B, C, E)?$, $I(B, D, C)?$, $I(B, D, E)?$, $I(B, E, C)?$, $I(B, E, D)?$, $I(C, B, A)?$, $\ldots$,$I(C, D, A)?$, $I(C, D, B)?$.



The original model



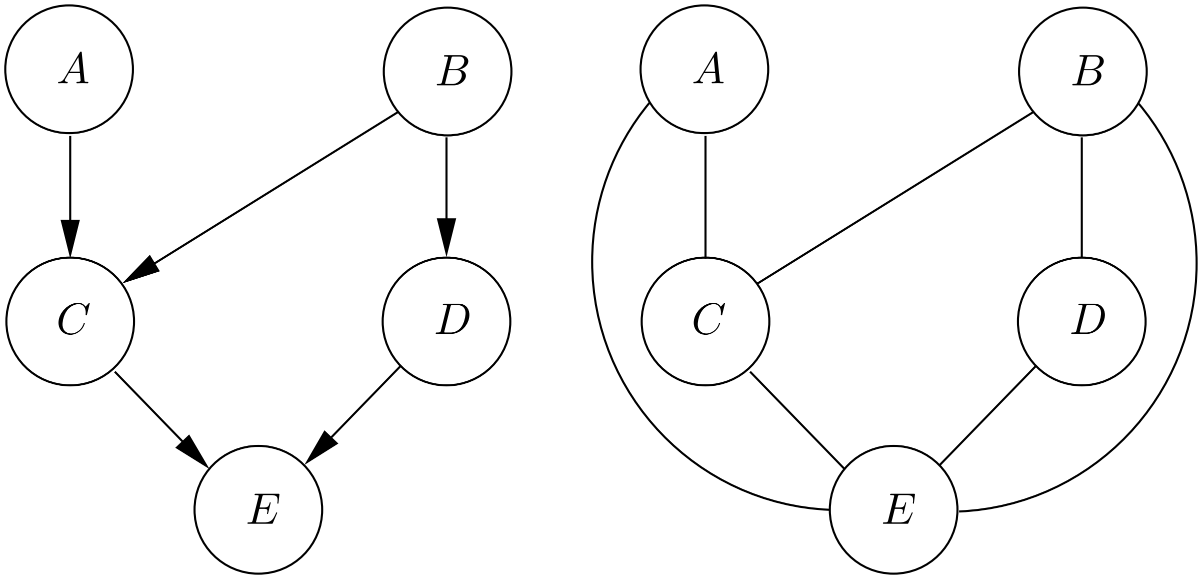
After one iteration

The question $I(C, D, B)?$ has the answer "yes":

➤ we therefore remove the link $C - D$.

Example

We now condition on two variables and ask questions like $I(B, C, \{D, E\})?$.



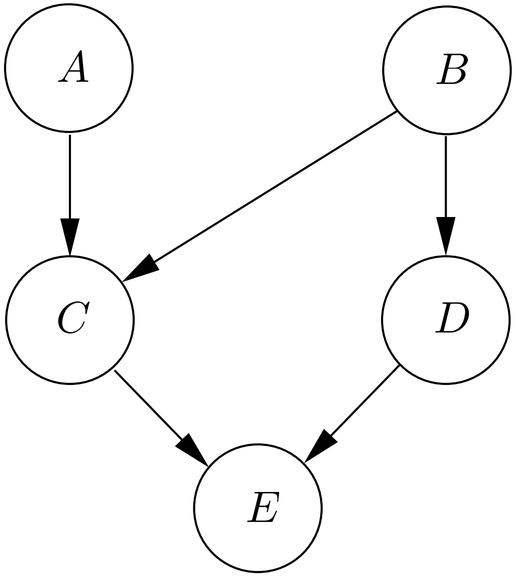
The original model After two iterations

The questions $I(B, E, \{C, D\})?$ and $I(E, A, \{C, D\})?$ have the answer “yes”:

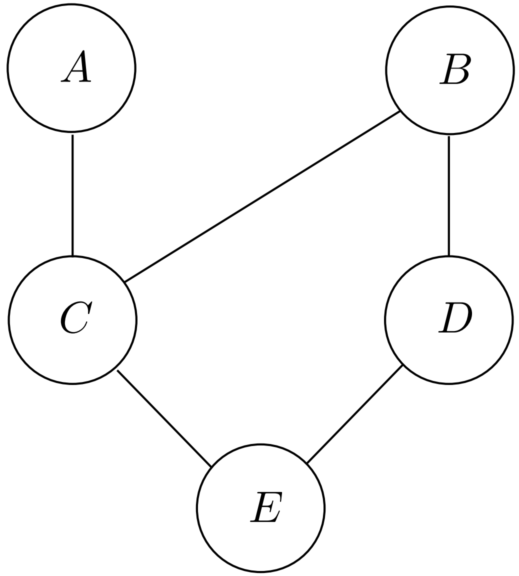
- we therefore remove the links $B - E$ and $A - E$.

Example

We now condition on three variables, but since no node has four neighbors we are finished.



The original model



After three iterations

The identified set of independence statements are then $I(A, B)$, $I(A, D)$, $I(C, D, B)$, $I(A, E, \{C, D\})$, and $I(B, E, \{C, D\})$. They are sufficient for applying rules 1-4.

Real world data

The oracle is a statistical test, e.g. conditional mutual information:

$$CE(A,B|X) = \sum_X P(X) \sum_{A,B} P(A,B|X) \log \frac{P(A,B|X)}{P(A|X)P(B|X)}.$$

$$I(A,B,X) \Leftrightarrow CE(A,B|X) = 0.$$

However, all tests have false positives and false negatives, which may provide false results/causal relations!

Similarly, false results may also be caused to hidden variables: