Bayesian Networks

11. Structure Learning / Constrained-based

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Business Economics and Information Systems
& Institute for Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

The Very Last Step

- Assume, we know the whole structure of a bn except a single edge.
- This edge represents a single independence statement.
- Check it and include edge based on outcome of that test.

Exact Check / Example (1/3)

If X and Y are independent, then

$$p(X, Y) = p(X) \cdot p(Y)$$

observed

$Y =$	0	1
$X = 0$	3	6
1	1	2

observed relative frequencies $p(X, Y)$:

$Y =$	0	1
$X = 0$	0.25	0.5
1	0.083	0.167

expected relative frequencies $p(X)\, p(Y)$:

$Y =$	0	1
$X = 0$	0.25	0.5
1	0.083	0.167

Exact Check / Example (2/3)

If X and Y are independent, then

$$p(X, Y) = p(X) \cdot p(Y)$$

observed

$Y =$	0	1
$X = 0$	3000	6000
1	1000	2000

observed relative frequencies $p(X, Y)$:

$Y =$	0	1
$X = 0$	0.25	0.5
1	0.083	0.167

expected relative frequencies $p(X)\, p(Y)$:

$Y =$	0	1
$X = 0$	0.25	0.5
1	0.083	0.167

Exact Check / Example (3/3)

If X and Y are independent, then

$$p(X, Y) = p(X) \cdot p(Y)$$

observed

$Y =$	0	1
$X = 0$	2999	6001
1	1000	2000

observed relative frequencies $p(X, Y)$:

$Y =$	0	1
$X = 0$	0.2499167	0.5000833
1	0.0833333	0.1666667

expected relative frequencies $p(X)\, p(Y)$:

$Y =$	0	1
$X = 0$	0.2499375	0.5000625
1	0.0833125	0.1666875

Gamma function (repetition, see I.2)

Definition 1. Gamma function

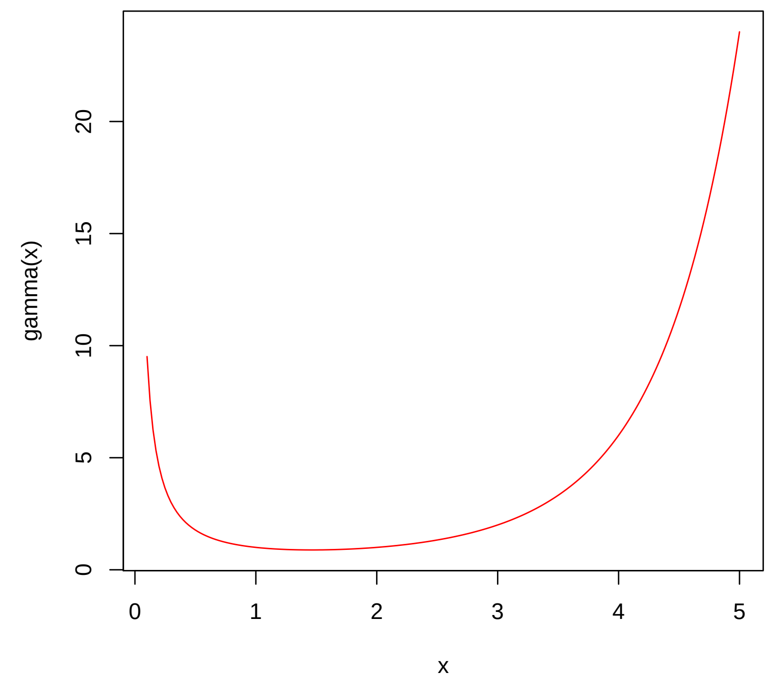
$$\Gamma(a) := \int_0^\infty t^{a-1} e^{-t} dt$$

converging for $a > 0$.

Lemma 1 (Γ is generalization of factorial)**.**

(i) $\Gamma(n) = (n-1)!$ *for* $n \in \mathbb{N}$*.*

(ii) $\frac{\Gamma(a+1)}{\Gamma(a)} = a$*.*



Incomplete Gamma Function

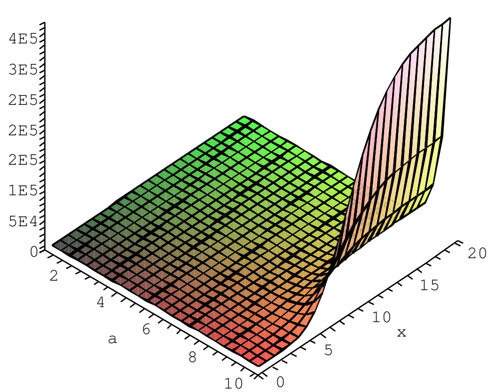
Definition 2. Incomplete Gamma function

$$\gamma(a,x) := \int_0^x t^{a-1} e^{-t} dt$$

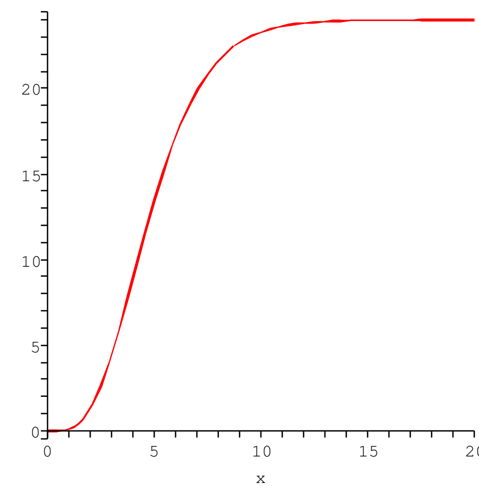
defined for $a > 0$ and $x \in [0, \infty]$.

Lemma 2.

$$\gamma(a,\infty) = \Gamma(a)$$



γ



$\gamma(5,x)$

χ^2 distribution

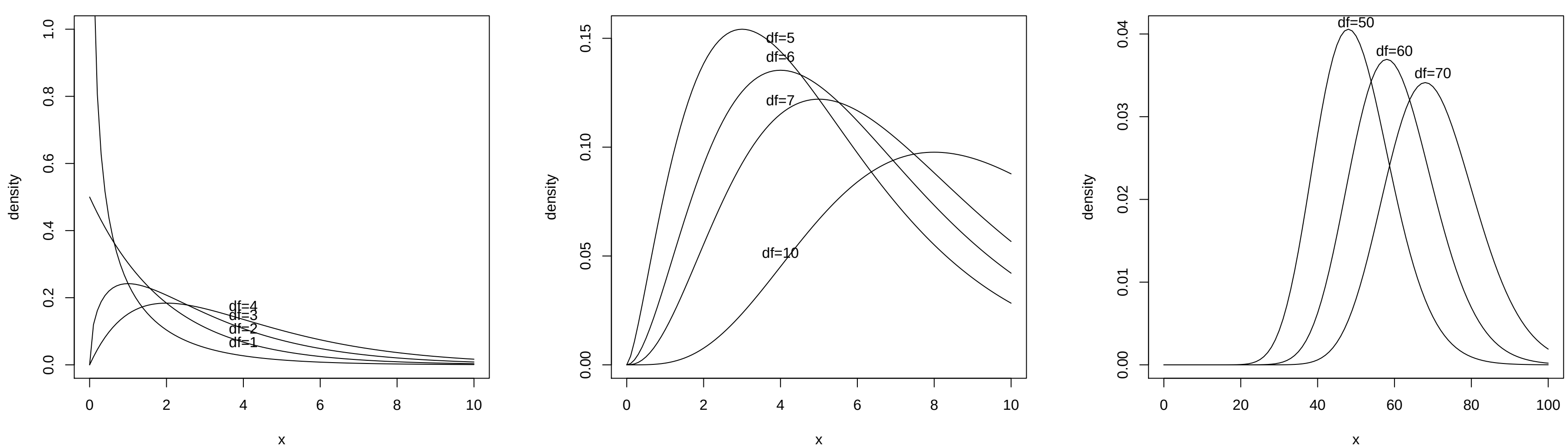
Definition 3. chi-square distribution (χ^2) has density

$$p(x) := \frac{1}{2^{\frac{\mathrm{df}}{2}}\,\Gamma(\frac{\mathrm{df}}{2})}\, x^{\frac{\mathrm{df}}{2}-1}\, e^{-\frac{x}{2}};$$

defined on $]0,\infty[$.

Its cumulative distribution function (cdf) is:

$$p(X<x) := \frac{\gamma(\frac{\mathrm{df}}{2},\frac{x}{2})}{\Gamma(\frac{\mathrm{df}}{2})};$$



$\chi^2_1, \chi^2_2, \chi^2_3, \chi^2_4$ $\chi^2_5, \chi^2_6, \chi^2_7, \chi^2_{10}$ $\chi^2_{50}, \chi^2_{60}, \chi^2_{70}$

χ^2 distribution

Lemma 3.

$$E(\chi^2(x, \mathrm{df})) = \mathrm{df}$$

A Java implementation of the incomplete gamma function (and thus of χ^2 distribution) can be found, e.g., in COLT (`http://dsd.lbl.gov/~hoschek/colt/`), package `cern.jet.stat`, class `Gamma`.

Be careful, sometimes

$$\tilde{\gamma}(a, x) := \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt = \frac{1}{\Gamma(a)} \gamma(a, x) \qquad \text{(e.g., R)}$$

or

$$\tilde{\gamma}(a, x) := \int_x^\infty t^{a-1} e^{-t} dt = \Gamma(a) - \gamma(a, x) \qquad \text{(e.g., Maple)}$$

are referenced as incomplete gamma function.

Count Variables / Just 2 Variables

Let X, Y be random variables, $D \subseteq \mathrm{dom}(X) \times \mathrm{dom}(Y)$
and for two values $x \in \mathrm{dom}(X), y \in \mathrm{dom}(Y)$

$$c_{X=x} := |\{d \in D \mid d_{|X} = x\}|$$
$$c_{Y=y} := |\{d \in D \mid d_{|X} = x\}|$$
$$c_{X=x,Y=y} := |\{d \in D \mid d_{|X} = x, d_{|Y} = y\}|$$

their counts.

χ^2 and G^2 statistics / Just 2 Variables

If X, Y are independent, then

$$p(X,Y) = p(X)\,p(Y)$$

and thus

$$E(c_{X=x,Y=y} \,|\, c_{X=x}, c_{Y=y}) = \frac{c_{X=x} \cdot c_{Y=y}}{|D|}$$

Then the statistics

$$\chi^2 := \sum_{x \in \mathrm{dom}(X)} \sum_{y \in \mathrm{dom}(Y)} \frac{(c_{X=x,Y=y} - \frac{c_{X=x} \cdot c_{Y=y}}{|D|})^2}{\frac{c_{X=x} \cdot c_{Y=y}}{|D|}}$$

as well as

$$G^2 := 2 \sum_{x \in \mathrm{dom}(X)} \sum_{y \in \mathrm{dom}(Y)} c_{X=x,Y=y} \cdot \ln \left(\frac{c_{X=x,Y=y}}{\left(\frac{c_{X=x} \cdot c_{Y=y}}{|D|} \right)} \right)$$

are asymptotically χ^2-distributed with
$\mathrm{df} = (|\,\mathrm{dom}(X)| - 1)\,(|\,\mathrm{dom}(Y)| - 1)$ degrees of freedom.

Testing Independency / informal

Generally, the statistics have the form

$$\chi^2 = \sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

$$G^2 = \sum \text{observed} \ln \left(\frac{\text{observed}}{\text{expected}} \right)$$

$\chi^2 = 0$ and $G^2 = 0$ for exact independent variables.

The larger χ^2 and G^2, the more likely / stronger the depedency between X and Y.

Testing Independency / more formally

More formally, under the

null hypothesis of independency of X and Y,

the probability for χ^2 and G^2 to have the computed values (or even larger ones) is

$$p_{\chi^2_{\mathrm{df}}}(X > \chi^2) \quad \text{and} \quad p_{\chi^2_{\mathrm{df}}}(X > G^2)$$

Let p_0 be a given threshold called **significance level** and often choosen as 0.05 or 0.01.

- If $p(X > \chi^2) < p_0$, we can **reject the null hypothesis** and thus accept its

  alternative hypothesis of dependency of X and Y.

  i.e., add the edge between X and Y.
- If $p(X > \chi^2) \geq p_0$, we cannot reject the null hypothesis. Here, we then will accept the null hypothesis, i.e., not add the edge between X and Y.

Example 1

observed

$Y =$	0	1
$X = 0$	3	6
1	1	2

margins

$Y =$	0	1	$\sum$
$X = 0$	3	6	9
1	1	2	3
$\sum$	4	8	12

expected

$Y =$	0	1	$\sum$
$X = 0$	3	6	9
1	1	2	3
$\sum$	4	8	12

$$\chi^2 = G^2 = 0 \quad \text{and} \quad p(X > 0) = 1$$

Hence, for any significance level
X and Y are considered independent.

Example 2 (1/2)

observed

$Y =$	0	1
$X = 0$	6	1
1	2	4

margins

$Y =$	0	1	$\sum$
$X = 0$	6	1	7
1	2	4	6
$\sum$	8	5	13

expected

$Y =$	0	1	$\sum$
$X = 0$	4.31	2.69	7
1	3.69	2.31	6
$\sum$	8	5	13

$$\chi^2 = \frac{(6-4.31)^2}{4.31} + \frac{(1-2.69)^2}{2.69} + \frac{(2-3.69)^2}{3.69} + \frac{(4-2.31)^2}{2.31}$$
$$= 3.75,$$

$$\rightsquigarrow p_{\chi^2_1}(X > 3.75) = 0.053$$

i.e., with a significance level of $p_0 = 0.05$
we would **not** be able to reject the null hypothesis of independency of X and Y.

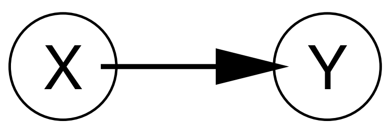
Example 2 (2/2)

If we use G^2 instead of χ^2,

$$G^2 = 3.94, \quad p_{\chi^2_1}(X > 3.94) = 0.047$$

with a significance level of $p_0 = 0.05$
we would have to reject the null hypothesis of independency of X and Y.

Here, we then accept the alternative, depedency of X and Y.



count variables / general case

Let $\mathcal{V}$ be a set of random variables.

We write $v \in \mathcal{V}$ as abbreviation for $v \in \prod \mathrm{dom}(\mathcal{V})$.

For a dataset $D \subseteq \prod \mathrm{dom}(\mathcal{V})$ and

- each subset $\mathcal{X} \subseteq \mathcal{V}$ of variables and
- each configuration $x \in \mathcal{X}$ of these variables

let

$$c_{\mathcal{X}=x} := |\{d \in D \mid d_{|\mathcal{X}} = x\}|$$

be a (random) variable containing the frequencies of occurences of $\mathcal{X} = x$ in the data.

G^2 statistics / general case

Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \mathcal{V}$ be three disjoint subsets of variables. If

$$I(\mathcal{X}, \mathcal{Y} \,|\, \mathcal{Z})$$

then

$$p(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = \frac{p(\mathcal{X}, \mathcal{Z})\, p(\mathcal{Y}, \mathcal{Z})}{p(\mathcal{Z})}$$

and thus for each configuration $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $z \in \mathcal{Z}$

$$E(c_{\mathcal{X}=x,\mathcal{Y}=y,\mathcal{Z}=z} \,|\, c_{\mathcal{X}=x,\mathcal{Z}=z}, c_{\mathcal{Y}=y,\mathcal{Z}=z}) = \frac{c_{\mathcal{X}=x,\mathcal{Z}=z}\, c_{\mathcal{Y}=y,\mathcal{Z}=z}}{c_{\mathcal{Z}=z}}$$

The statistics

$$G^2 := 2 \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \sum_{z\in\mathcal{Z}} c_{\mathcal{X}=x,\mathcal{Y}=y,\mathcal{Z}=z} \cdot \ln\left(\frac{c_{\mathcal{X}=x,\mathcal{Y}=y,\mathcal{Z}=z} \cdot c_{\mathcal{Z}=z}}{c_{\mathcal{X}=x,\mathcal{Z}=z} \cdot c_{\mathcal{Y}=y,\mathcal{Z}=z}}\right)$$

is asymptotically χ^2-distributed with

$$\mathrm{df} = \prod_{X\in\mathcal{X}} (|\operatorname{dom} X| - 1) \prod_{Y\in\mathcal{Y}} (|\operatorname{dom} Y| - 1) \prod_{Z\in\mathcal{Z}} |\operatorname{dom} Z|$$

degrees of freedom.

Recommendations

Recommendations [SGS00, p. 95]:

- As heuristics, reduce degrees of freedom by 1 for each **structural zero**:

$$\mathrm{df}^{\text{reduced}} := \mathrm{df} - |\{(x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \mid c_{\mathcal{X}=x, \mathcal{Y}=y, \mathcal{Z}=z} = 0\}|$$

- Use G^2 instead of χ^2.
- If $|D| < 10\,\mathrm{df}$, assume conditional dependency.

Problems:

- null hypothesis is accepted if it is not rejected. (especially problematic for small samples)
- repeated testing.

1. Checking Probalistic Independencies

2. Markov Equivalence and DAG patterns

3. PC Algorithm

Markov-equivalence

Definition 4. Let G, H be two graphs on a set V (undirected or DAGs).
G and H are called **markov-equivalent**, if they have the same independency model, i.e.

$$I_G(X, Y|Z) \Leftrightarrow I_H(X, Y|Z), \quad \forall X, Y, Z \subseteq V$$

The notion of markov-equivalence for undirected graphs is uninteresting, as every undirected graph is markov-equivalent only to itself (corollary of uniqueness of minimal representation!).

Markov-equivalence

Why is markov-equivalence important?

1. in structure learning, the set of all graphs over V is our search space.
   $\rightsquigarrow$ if we can restrict searching to equivalence classes, the search space becomes smaller.
2. if we interpret the edges of our graph as causal relationships between variables, it is of interest,
   - which edges are necessary
     (i.e., occur in all instances of the equivalence class), and
   - which edges are only possible
     (i.e., occur in some instances of the equivalence class, but not in some others; i.e., there are alternative explanations).

Markov-equivalence

Definition 5. Let G be a directed graph. We call a chain

$$p_1 - p_2 - p_3$$

uncoupled if there is no edge between p_1 and p_3.

Lemma 4 (markov-equivalence criterion, [PGV90]). *Let G and H be two DAGs on the vertices V.*

G and H are markov-equivalent if and only if

(i) G and H have the same links ($u(G) = u(H)$) and

(ii) G and H have the same uncoupled head-to-head meetings.

The set of uncoupled head-to-head meetings is also denoted as **V-structure** of G.

Markov-equivalence / examples

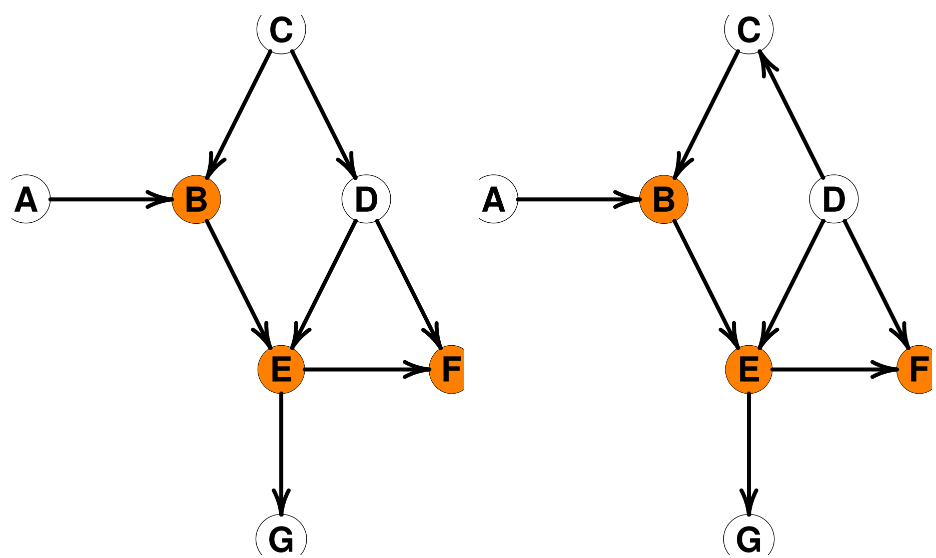


Figure 1: Example for markov-equivalent DAGs.

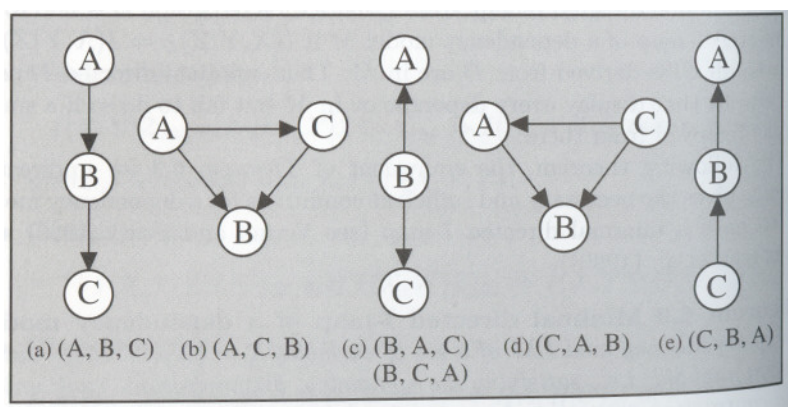


Figure 2: Which minimal DAG-representations of I are equivalent? [CGH97, p. 240]

Directed graph patterns

Definition 6. Let V be a set and $E \subseteq V^2 \cup \mathcal{P}^2(V)$ a set of ordered and unordered pairs of elements of V with $(v,w),(w,v) \notin E$ for $v,w \in V$ with $\{v,w\} \in E$.
Then $G := (V,E)$ is called a **directed graph pattern**. The elements of V are called vertices, the elemtents of E **edges**: unordered pairs are called **undirected edges**, ordered pairs **directed edges**.

We say, a directed graph pattern H is **a pattern of the directed graph** G, if there is an orientation of the unoriented edges of H that yields G, i.e.

$$(v,w) \in E_G \Rightarrow \begin{cases} (v,w) \in E_H \text{ or} \\ \{v,w\} \in E_H \end{cases}$$

$$(v,w) \in E_G \Leftarrow (v,w) \in E_H$$

$$\left.\begin{array}{l} (v,w) \in E_G \text{ or} \\ (w,v) \in E_G \end{array}\right\} \Leftarrow \{v,w\} \in E_H$$

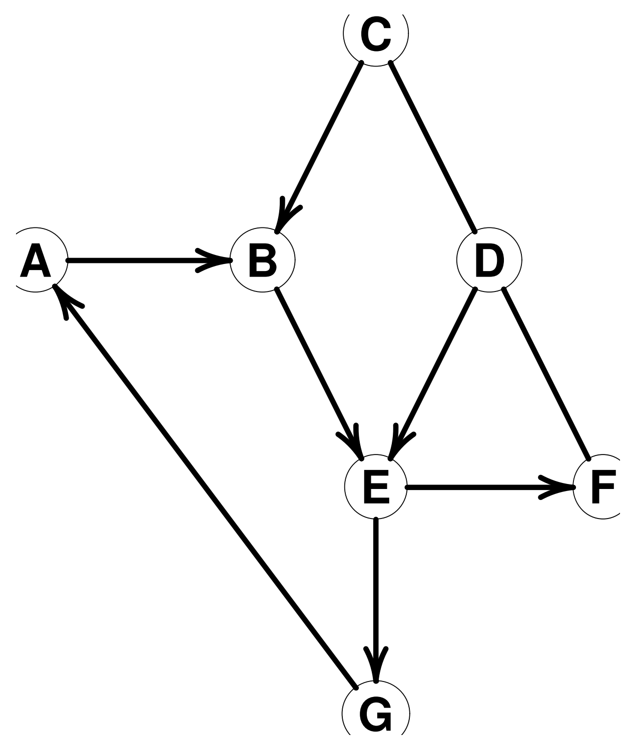


Figure 3: Directed graph pattern.

DAG patterns

Definition 7. A directed graph pattern H is called an **acyclic directed graph pattern** (DAG pattern), if

- it is the directed graph pattern of a DAG G

  or equivalently

- H does not contain a completely directed cycle, i.e. there is no sequence $v_1, \ldots, v_n \in V$ with $(v_i, v_{i+1}) \in E$ for $i = 1, \ldots, n-1$ (i.e. the directed graph got by dropping undirected edges is a DAG).

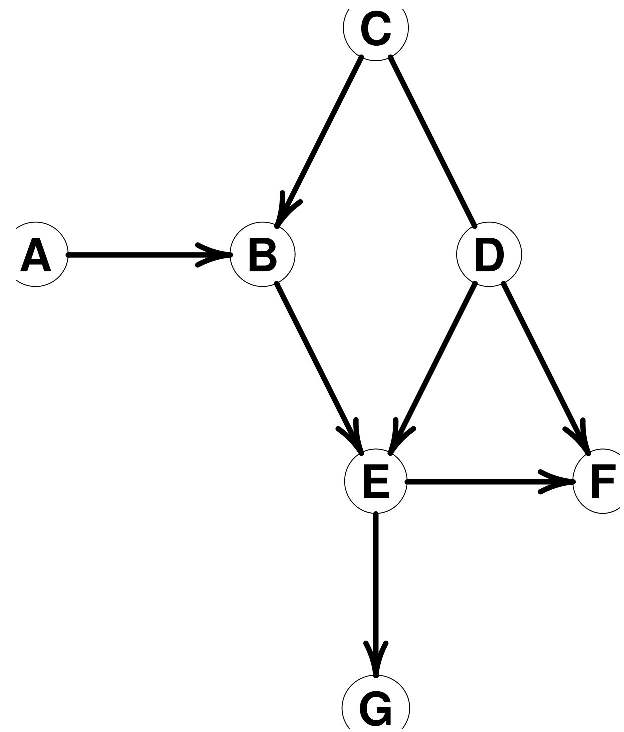


Figure 4: DAG pattern.

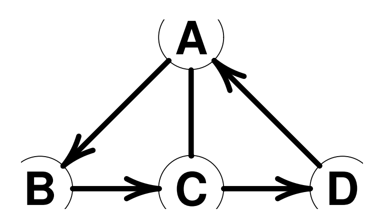


Figure 5: Directed graph pattern that is not a DAG pattern.

DAG patterns represent markov equivalence classes

Lemma 5. *Each markov equivalence class corresponds uniquely to a DAG pattern G:*

(i) The markov equivalence class consists of all DAGs that G is a pattern of, i.e., that give G by dropping the directions of some edges that are not part of an uncoupled head-to-head meeting,

(ii) The DAG pattern contains a directed edge (v, w), if all representatives of the markov equivalence class contain this directed edge, otherwise (i.e. if some representatives have (v, w), some others (w, v)) the DAG pattern contains the undirected edge $\{v, w\}$.

The directed edges of the DAG pattern are also called **irreversible** or **compelled**, the undirected edges are also called **reversible**.

DAG patterns represent markov equivalence classes / example

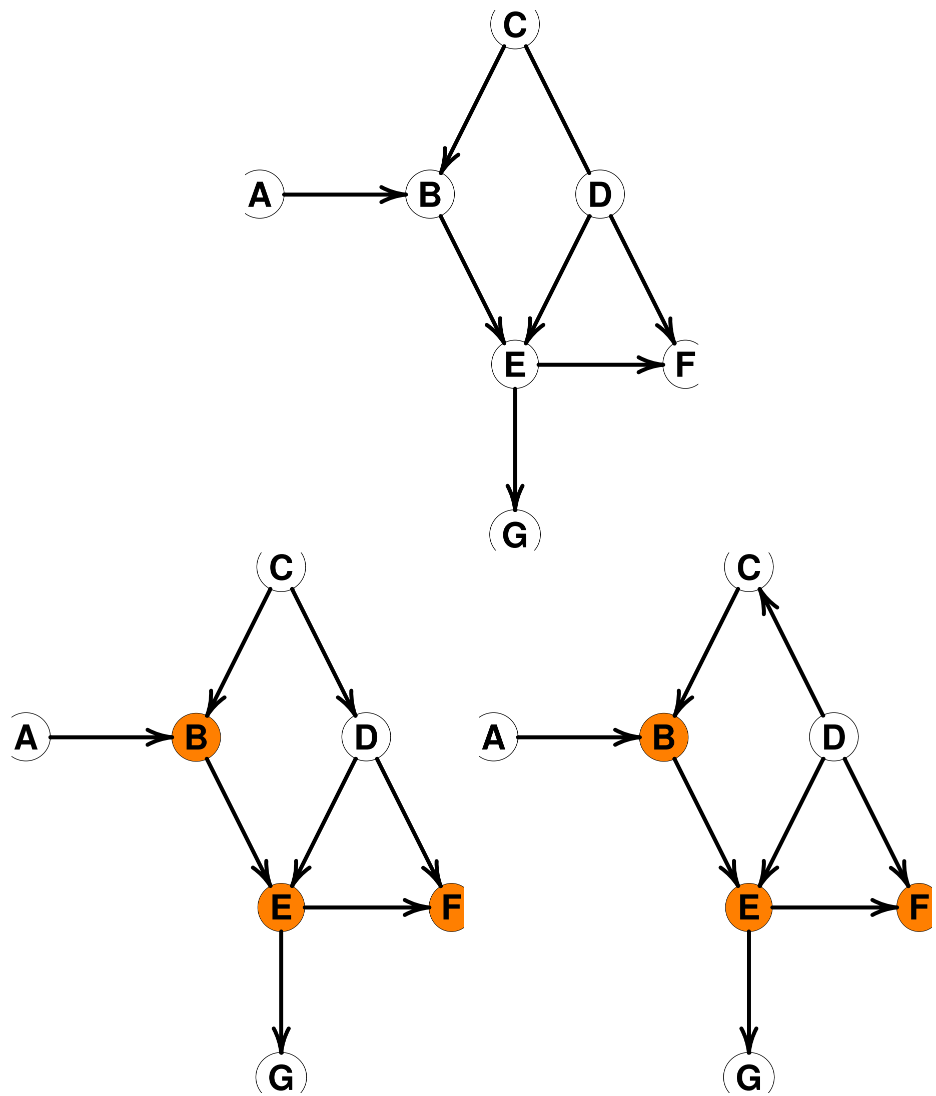
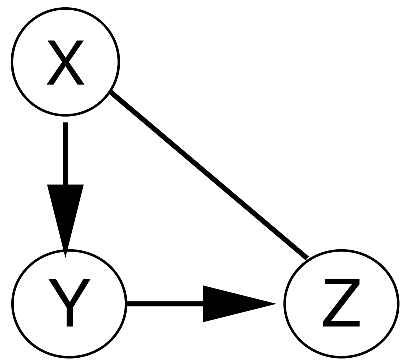


Figure 6: DAG pattern and its markov equivalence class representatives.

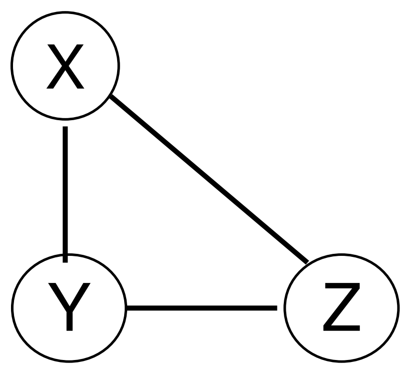
DAG patterns represent markov equivalence classes

But beware, not every DAG pattern represents a Markov-equivalence class !

Example:



is not a DAG pattern of a Markov-equivalence class, but

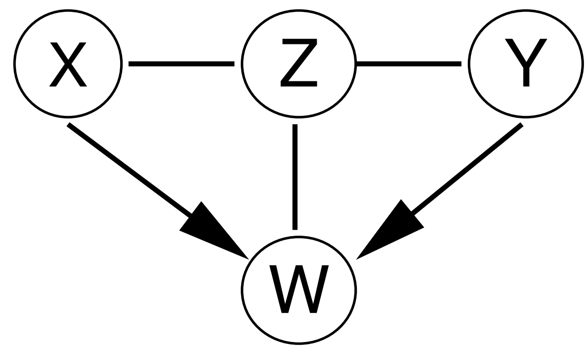


is.

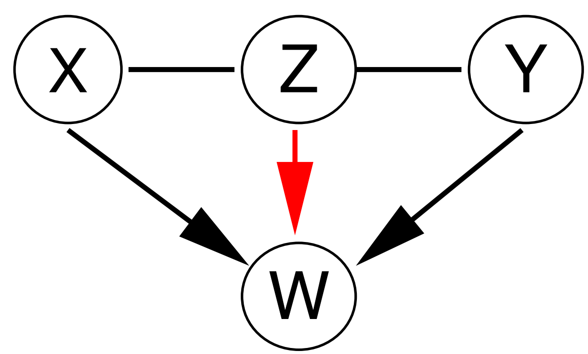
DAG patterns represent markov equivalence classes

But just skeleton plus uncoupled head-to-head meetings do not make a DAG pattern that represents a markov-equivalence class either.

Example:



is not a DAG pattern that represents a Markov-equivalence class, as any of its represenatives also has $Z \to W$. But



is.

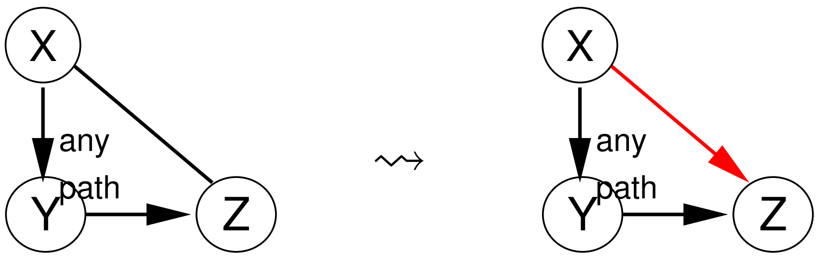
Computing DAG patterns

So, to compute the DAG pattern that represents the equivalence class of a given DAG,

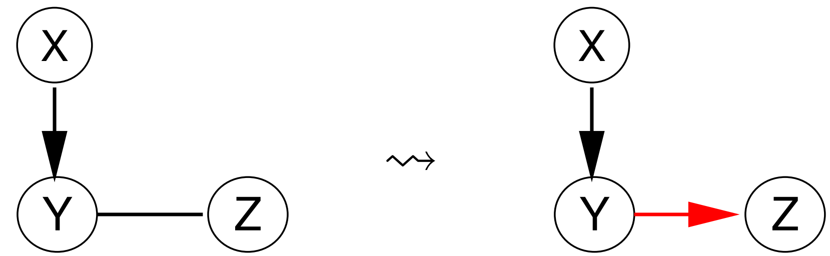
1. start with the skeleton plus all uncoupled head-to-head-meetings,
2. add entailed edges successively (saturating).

Saturating DAG patterns

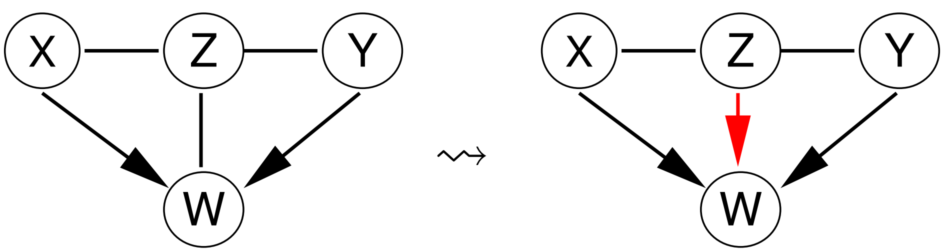
rule 1:



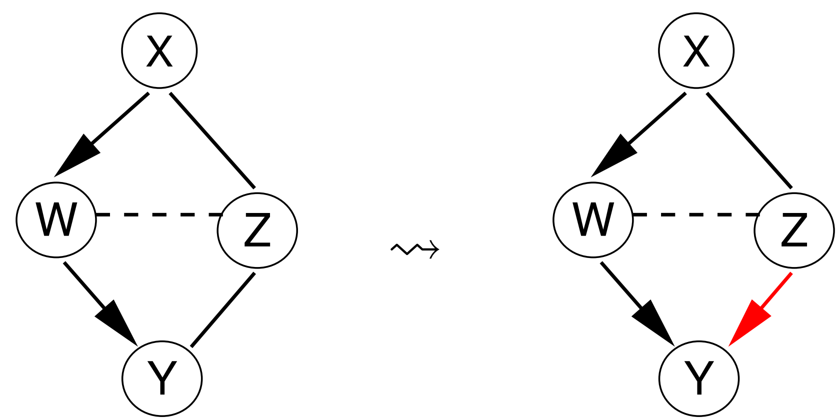
rule 2:



rule 3:



rule 4:



Dashed link can be $W \rightarrow Z$, $W \leftarrow Z$, or $W{-}Z$
(so rule 4 is actually a compact notation for 3 rules).

Computing DAG patterns

saturate(graph pattern $G = (V, E)$) :
apply rules 1–4 to G until no more rule matches
return G

dag-pattern(graph $G = (V, E)$) :
$H := (V, F)$ with $F := \{\{x, y\} \mid (x, y) \in E\}$
for $X \rightarrow Z \leftarrow Y$ uncoupled head-to-head-meeting in G **do**
    orient $X \rightarrow Z \leftarrow Y$ in H
od
saturate(H)
return H

Figure 7: Algorithm for computing the DAG pattern of the Markov-equivalence class of a given DAG.

Lemma 6. *For a given graph G, algorithm 7 computes correctly the DAG pattern that represents its Markov-equivalence class.*

Furthermore, here, even the rule set 1–3 will do and is non-redundant.

See [Mee95] for a proof.

Computing DAG patterns

What follows, is an alternative algorithm for computing DAG patterns that represent the Markov-equivalence class of a given DAG.

Toplogical edge ordering

Definition 8. Let $G := (V, E)$ be a directed graph.
A bijective map

$$\tau : \{1, \ldots, |E|\} \to E$$

is called an **ordering of the edges of** G.

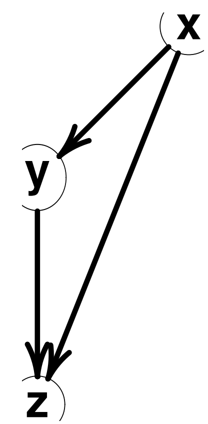
An edge ordering τ is called **topological edge ordering** if

(i) numbers increase on all paths, i.e.

$$\tau^{-1}(x, y) < \tau^{-1}(y, z)$$

for paths $x \to y \to z$ and

(ii) shortcuts have larger numbers, i.e. for x, y, z with



it is

$$\tau^{-1}(x, y) < \tau^{-1}(y, z) \overset{!}{<} \tau^{-1}(x, z)$$

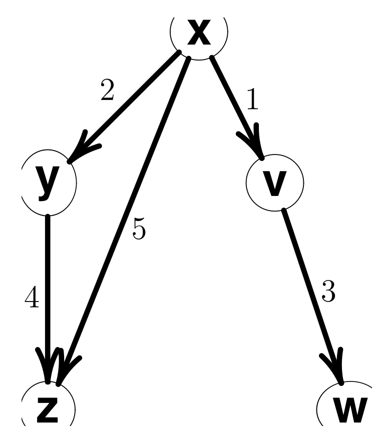


Figure 8: Example for a topological edge ordering.

Toplogical edge ordering

topological-edge-ordering$(G = (V, E))$:
$\sigma :=$ *topological-ordering*(G)
$E' := E$
for $i = 1, \ldots, |E|$ **do**
    Let $(v, w) \in E'$ with $\sigma^{-1}(w)$ minimal and then with $\sigma^{-1}(v)$ maximal
    $\tau(i) := (v, w)$
    $E' := E' \setminus \{(v, w)\}$
od
return τ

Figure 9: Algorithm for computing a topological edge ordering of a DAG.

dag-pattern$(G = (V, E))$:
$\tau :=$ *topological-edge-ordering*(G)
$E_{\text{irr}} := \emptyset$
$E_{\text{rev}} := \emptyset$
$E_{\text{rest}} := E$
while $E_{\text{rest}} \neq \emptyset$ **do**
    Let $(y, z) \in E_{\text{rest}}$ with $\tau^{-1}(y, z)$ minimal
    [label $\text{pa}(z)$:]
    if $\exists (x, y) \in E_{\text{irr}}$ with $(x, z) \notin E$
        $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \mid x' \in \text{pa}(z)\}$
    else
        $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \mid (x', y) \in E_{\text{irr}}\}$
        if $\exists (x, z) \in E$ with $x \notin \{y\} \cup \text{pa}(y)$
            $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \mid (x', z) \in E_{\text{rest}}\}$
        else
            $E_{\text{rev}} := E_{\text{rev}} \cup \{(x', z) \mid (x', z) \in E_{\text{rest}}\}$
        fi
    fi
    $E_{\text{rest}} := E \setminus E_{\text{irr}} \setminus E_{\text{rev}}$
od
return $\bar{G} := (V, E_{\text{irr}} \cup \{\{v, w\} | (v, w) \in E_{\text{rev}}\})$

Figure 10: Algorithm for computing the DAG pattern representing the markov equivalence class of a DAG G. [Chi95]

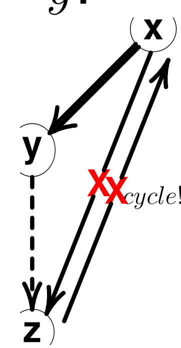
A simple but important lemma

Lemma 7 ([Chi95])**.** *Let* G *be a DAG and* x, y, z *three vertices of* G *that are pairwise adjacent.*
If any two of the connecting edges are reversible, then the third one is also.

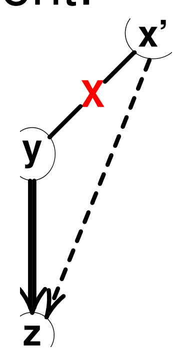
line 10

1 dag-pattern$(G = (V, E))$:
2 $\tau :=$ *topological-edge-ordering*(G)
3 $E_{\text{irr}} := \emptyset$
4 $E_{\text{rev}} := \emptyset$
5 $E_{\text{rest}} := E$
6 **while** $E_{\text{rest}} \neq \emptyset$ **do**
7 Let $(y, z) \in E_{\text{rest}}$ with $\tau^{-1}(y, z)$ minimal
8 [label $\text{pa}(z)$:]
9 **if** $\exists (x, y) \in E_{\text{irr}}$ with $(x, z) \notin E$
10 $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \,|\, x' \in \text{pa}(z)\}$
11 **else**
12 $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \,|\, (x', y) \in E_{\text{irr}}\}$
13 **if** $\exists (x, z) \in E$ with $x \notin \{y\} \cup \text{pa}(y)$
14 $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \,|\, (x', z) \in E_{\text{rest}}\}$
15 **else**
16 $E_{\text{rev}} := E_{\text{rev}} \cup \{(x', z) \,|\, (x', z) \in E_{\text{rest}}\}$
17 **fi**
18 **fi**
19 $E_{\text{rest}} := E \setminus E_{\text{irr}} \setminus E_{\text{rev}}$
20 **od**
21 **return** $\bar{G} := (V, E_{\text{irr}} \cup \{\{v, w\} | (v, w) \in E_{\text{rev}}\})$

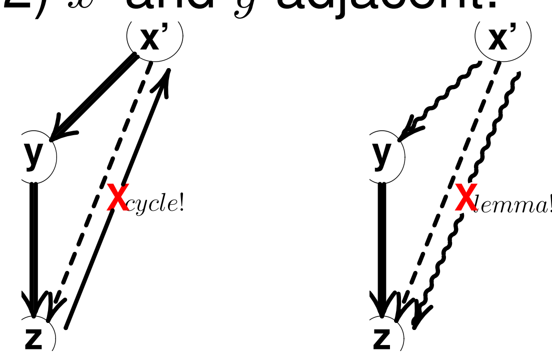
a) $x' = y$:



b) $x' \neq y$: case 1) x' and y not adjacent:



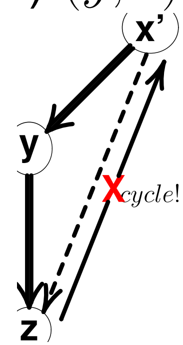
case 2) x' and y adjacent:



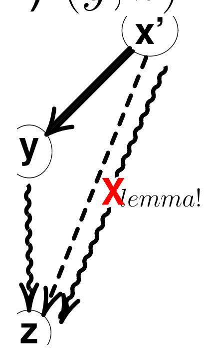
line 12

1 dag-pattern$(G = (V, E))$:
2 $\tau :=$ *topological-edge-ordering*(G)
3 $E_{\text{irr}} := \emptyset$
4 $E_{\text{rev}} := \emptyset$
5 $E_{\text{rest}} := E$
6 **while** $E_{\text{rest}} \neq \emptyset$ **do**
7   Let $(y, z) \in E_{\text{rest}}$ with $\tau^{-1}(y, z)$ minimal
8   [label pa(z) :]
9   **if** $\exists (x, y) \in E_{\text{irr}}$ with $(x, z) \notin E$
10       $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \,|\, x' \in \text{pa}(z)\}$
11   **else**
12         $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \,|\, (x', y) \in E_{\text{irr}}\}$
13   **if** $\exists (x, z) \in E$ with $x \notin \{y\} \cup \text{pa}(y)$
14           $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \,|\, (x', z) \in E_{\text{rest}}\}$
15   **else**
16             $E_{\text{rev}} := E_{\text{rev}} \cup \{(x', z) \,|\, (x', z) \in E_{\text{rest}}\}$
17   **fi**
18   **fi**
19     $E_{\text{rest}} := E \setminus E_{\text{irr}} \setminus E_{\text{rev}}$
20 **od**
21 **return** $\bar{G} := (V, E_{\text{irr}} \cup \{\{v, w\} | (v, w) \in E_{\text{rev}}\})$

case 1) (y, z) is irreversible:



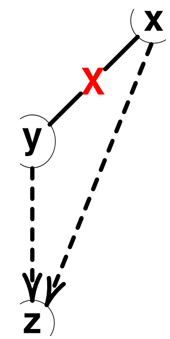
case 2) (y, z) is reversible:



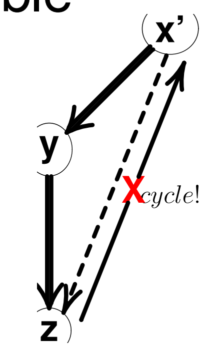
line 14

1 dag-pattern$(G = (V, E))$:
2 $\tau :=$ *topological-edge-ordering*(G)
3 $E_{\text{irr}} := \emptyset$
4 $E_{\text{rev}} := \emptyset$
5 $E_{\text{rest}} := E$
6 **while** $E_{\text{rest}} \neq \emptyset$ **do**
7 Let $(y, z) \in E_{\text{rest}}$ with $\tau^{-1}(y, z)$ minimal
8 [label $\text{pa}(z)$:]
9 **if** $\exists (x, y) \in E_{\text{irr}}$ with $(x, z) \notin E$
10 $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \mid x' \in \text{pa}(z)\}$
11 **else**
12 $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \mid (x', y) \in E_{\text{irr}}\}$
13 **if** $\exists (x, z) \in E$ with $x \notin \{y\} \cup \text{pa}(y)$
14 $E_{\text{irr}} := E_{\text{irr}} \cup \{(x', z) \mid (x', z) \in E_{\text{rest}}\}$
15 **else**
16 $E_{\text{rev}} := E_{\text{rev}} \cup \{(x', z) \mid (x', z) \in E_{\text{rest}}\}$
17 **fi**
18 **fi**
19 $E_{\text{rest}} := E \setminus E_{\text{irr}} \setminus E_{\text{rev}}$
20 **od**
21 **return** $\bar{G} := (V, E_{\text{irr}} \cup \{\{v, w\} | (v, w) \in E_{\text{rev}}\})$

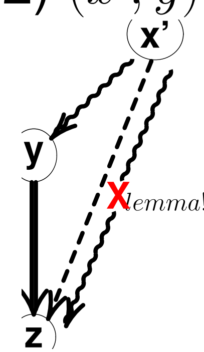
a) $x' = x$:



b) $x' \neq x$: case 1) (x', y) irreversible



case 2) (x', y) is reversible:



Summary (1/2)

- (Conditional) probabilistic independence in estimated JPDs has to be checked by means of a **statistical test** (e.g., χ^2, G^2).
- For those tests, a **test statistics** (χ^2) is computed and its probability under the assumption of independence is computed.
  1. If this is too small, the independency assumption is rejected and dependency assumed.
  2. If this exceeds a given lower bound, the independency assumption cannot be rejected and independency assumed.

Summary (2/2)

- Some DAGs encode the same independency relation (**Markov equivalence**).
- A Markov equivalence class can be represented by a **DAG pattern**.
  (but not all DAG patterns represent a Markov equivalence class!)
- For a given DAG, its DAG pattern can be computed by
  1. start from the undirected skeleton,
  2. add all directions of **uncoupled head-to-head meetings**,
  3. **saturate infered directions** (using 3 rules).

1. Checking Probalistic Independencies

2. Markov Equivalence and DAG patterns

3. PC Algorithm

References

[CGH97] Enrique Castillo, José Manuel Gutiérrez, and Ali S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer, New York, 1997.

[Chi95] D. Chickering. A transformational characterization of equivalent bayesian network structures. In Philippe Besnard and Steve Hanks, editors, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 87–98. Morgan Kaufmann, 1995.

[Mee95] C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence,* Montreal, QU, pages 403–418. Morgan Kaufmann, August 1995.

[PGV90] J. Pearl, D. Geiger, and T. S. Verma. The logic of influence diagrams. In R. M. Oliver and J. Q. Smith, editors, *Influence Diagrams, Belief Networks and Decision Analysis*. Wiley, Sussex, 1990. (a shorter version originally appeared in Kybernetica, Vol. 25, No. 2, 1989).

[SGS00] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2 edition, 2000.