CAUSAL NETWORKS

CAUSAL DISCOVERY FROM OBSERVATIONAL DATA

Fabio Stella

Department of Informatics, Systems and Communication,

University of Milan-Bicocca

Viale Sarca 336, 2016 Milan, ITALY

e-mail: fabio.stella@unimib.it

Twitter: FaSt@FabioAStella

Previous lectures assumed that the causal graph is given. What if we don't know the graph? Can we learn it? We will refer to this problem as structure identification or structure learning. In this lecture we show how the structure of the causal network can be learnt from observational data.

In particular, the lecture presents the following:

- Constraint-based algorithms
- The PC algorithm
- Semi-parametric causal discovery
- Additional topics

PART I

CONSTRAINT-BASED METHODS

THE FUNDAMENTAL PROBLEM OF CAUSAL INFERENCE

We cannot observe both $Y_i(1)$ and $Y_i(0)$, therefore we cannot observe the causal effect

$$\tau_i \triangleq Y_i(1) - Y_i(0)$$

Another relevant task is that of **CAUSAL DISCOVERY TASK**, i.e., the problem of discovering the causal model which helps to **EXPLAIN** the causal effect of the treatment X on the outcome Y.

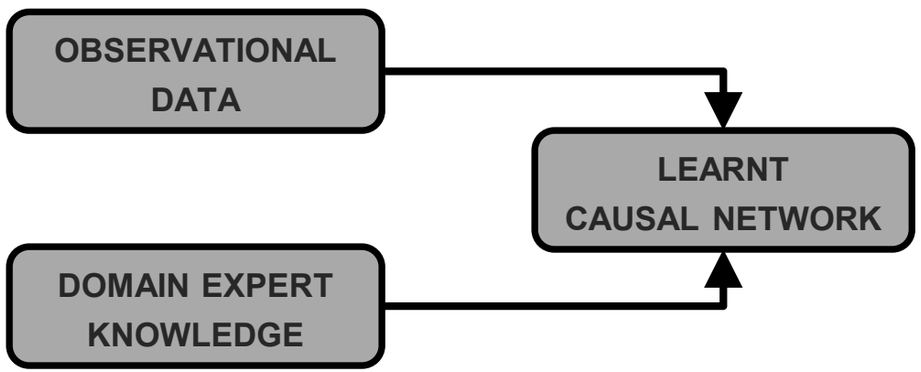


Figure 12.1

The **CAUSAL INFERENCE TASK** is an extremely relevant task and it consists of computing the causal effect τ_i of the treatment X on the outcome Y, no matter which is the causal model responsible for such a causal effect.

In this lecture we focus the attention to **CAUSAL NETWORKS**, as a valid tool to accomplish the **CAUSAL DISCOVERY TASK.**

More precisely we present the problem of **LEARNING A CAUSAL NETWORK** by fusion of **OBSERVATIONAL DATA** and **DOMAIN EXPERT KNOWLEDGE.**

Learning a **CAUSAL NETWORK** consists of learning

- Structure,
- Parameters.

Learning a **CAUSAL NETWORK** can be achieved by

- Constraint-based algorithms,
- Score-based algorithms,
- Hybrid algorithms.

However, in this lecture we focus the attention to **CONSTRAINT-BASED ALGORITHMS**.

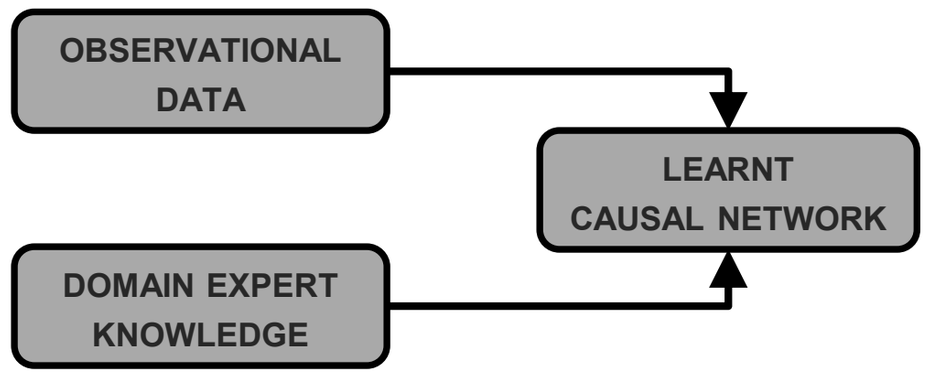


Figure 12.1

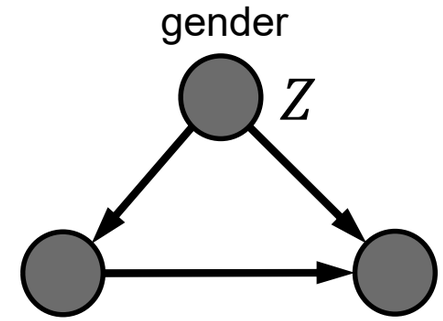


Figure 12.2

$$P(Z = 0) = 0.6$$

$$P(X = 0|Z = 0) = 0.5$$

$$P(X = 0|Z = 1) = 0.3$$

$$P(Y = 0|Z = 0, X = 0) = 0.6$$

$$P(Y = 0|Z = 1, X = 0) = 0.5$$

$$P(Y = 0|Z = 0, X = 1) = 0.1$$

$$P(Y = 0|Z = 1, X = 1) = 0.2$$

- We assume the **UNDERLYING PROCESS** follows a probability distribution P (the underlying probability distribution associated with DAG $\mathcal{G}$).
- Then, the **UNDERLYING PROCESS** can be adequately represented by sampling from P to obtain **OBSERVATIONAL DATA.**

The goal of the **CAUSAL DISCOVERY TASK** is to identify a model representation M of P.

To simplify the task, we assume the **PROBABILITY DISTRIBUTION** $\boldsymbol{P}$ to be a **DAG-FAITHFUL PROBABILITY DISTRIBUTION** with underlying DAG $\mathcal{G}$.

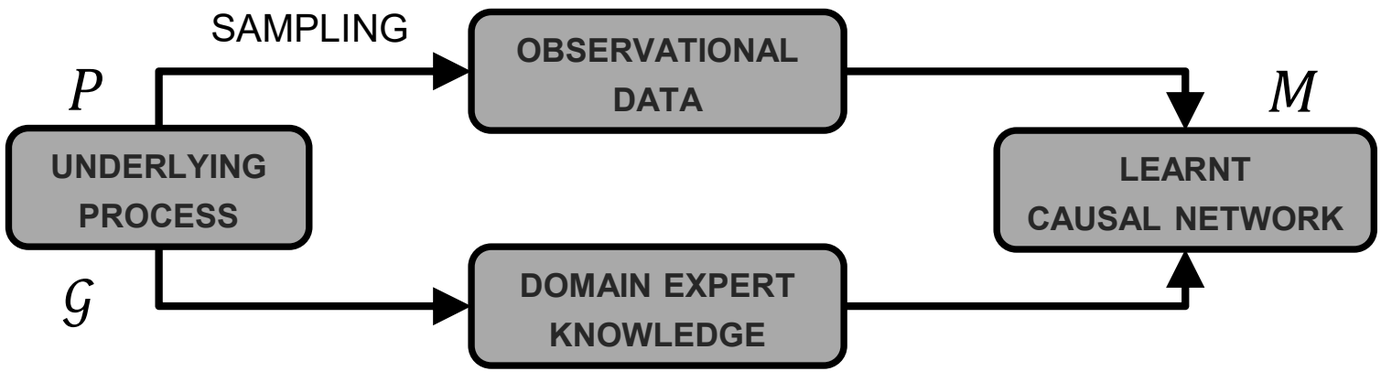


Figure 12.1

STABILITY – FAITHFULNESS

P is a stable (faithful) distribution if there exists a DAG $\mathcal{G}$ such that

$$\mathbf{X} \perp\!\!\!\perp_P \mathbf{Y} \mid \mathbf{Z} \Leftrightarrow \mathbf{X} \perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} \mid \mathbf{Z}$$

for any three sets of variables $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$.

We assume that the distribution P can be represented as a **CAUSAL NETWORK** (if P is not **DAG-FAITHFUL**, a causal network may still be an excellent approximation).

The **FAITHFULNESS** assumption says that the distribution P, induced by $\mathcal{G}$, satisfies no independence relations beyond those implied by $\mathcal{G}$.

A **CAUSAL NETWORK** is **FAITHFUL** iff for every d-connection (no d-separation) $\mathbf{Z}$ there is a corresponding conditional dependence, i.e.,

$$\mathbf{X} \not\perp\!\!\!\perp_{\mathcal{G}} \mathbf{Y} \mid \mathbf{Z} \Rightarrow \mathbf{X} \not\perp\!\!\!\perp_P \mathbf{Y} \mid \mathbf{Z}$$

FAITHFULNESS is a much less attractive assumption than the **MARKOV ASSUMPTION** because it is easy to think of counterexamples where

- two variables are independent in P,
- but there are unblocked paths between them in $\mathcal{G}$.

It is worthwhile to mention that many **CONSTRAINT-BASED METHODS** also assume that there are no unobserved confounders, which is known as **CAUSAL SUFFICIENCY**.

CAUSAL SUFFICIENCY

There are no unobserved confounders of any of the variables in the graph.

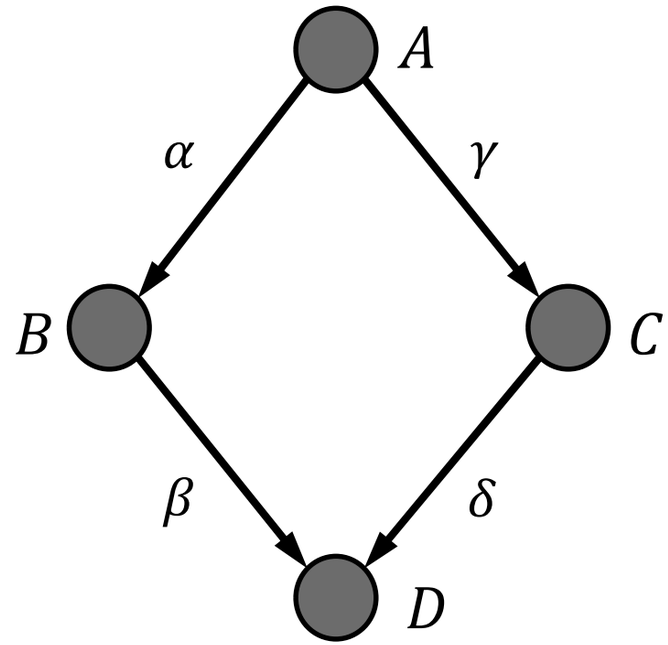


Figure 12.3

$$B \coloneqq \alpha A$$
$$C \coloneqq \gamma A$$
$$D \coloneqq \beta B + \delta C$$

$$D = \beta\alpha A + \delta\gamma A$$
$$D = (\beta\alpha + \delta\gamma)A$$

If $\beta\alpha = -\delta\gamma$

$A \longrightarrow B \longrightarrow D$
$A \longrightarrow C \longrightarrow D$

CANCEL OUT

$D = 0$

$A \perp\!\!\!\perp_P D$

$A \perp\!\!\!\perp_P D \mid \mathbf{Z} \Leftrightarrow A \perp\!\!\!\perp_{\mathcal{G}} D \mid \mathbf{Z}$

STABILITY − FAITHFULNESS

violates

$\mathbf{Z} = \{\varnothing\}$

$A \not\perp\!\!\!\perp_{\mathcal{G}} D$

$A \perp\!\!\!\perp_{\mathcal{G}} D$

$A \longrightarrow B \longrightarrow D$
$A \longrightarrow C \longrightarrow D$

there are no paths between A and D in the graph $\mathcal{G}$

cases generated by the underlying and unknown process

$$\mathfrak{D} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$$

independent and identically distributed data cases drawn at random from the probability distribution P

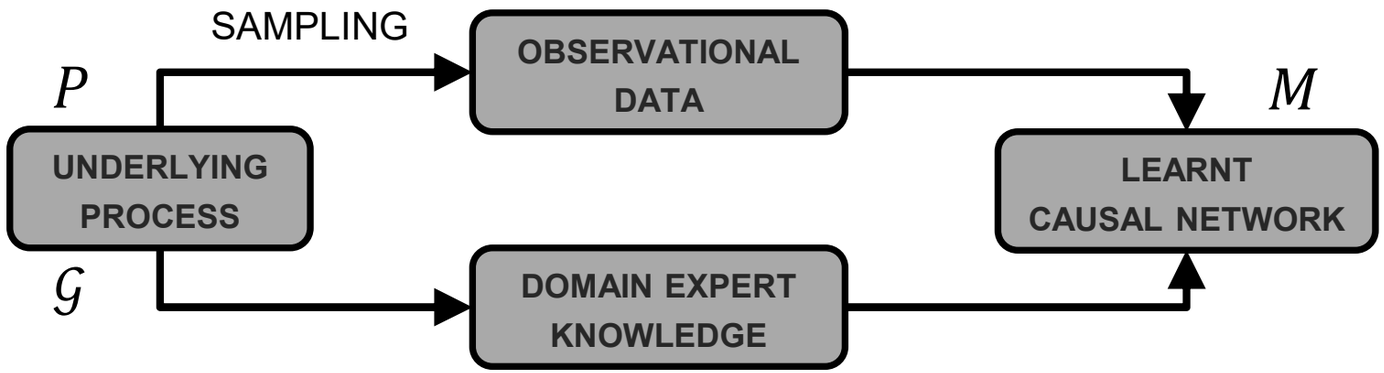


Figure 12.1

LEARNING A CAUSAL NETWORK is the **TASK** of identifying a DAG structure $\mathcal{G}$ and a set of conditional probability distributions with parameters Θ on the basis of

- $\mathfrak{D} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ and,
- possibly some domain expert background knowledge.

assignment of values to the n variables for the j^{th} data case

$$\boldsymbol{x}^j = \{x_1^j, x_2^j, \ldots, x_n^j\}$$

Some values in $\mathfrak{D}$ may be missing (N/A), but missing values are assumed to be:

- **MISSING AT RANDOM** (**MAR**) or,
- **MISSING COMPLETELY AT RANDOM** (**MCAR**). i.e., the missing data mechanism is uninformative and can be ignored.

A variable never observed is called a **HIDDEN** or a **LATENT VARIABLE**.

	X_1	X_2	...	X_n
$\boldsymbol{x}^1$	blue	yes	…	low
$\boldsymbol{x}^2$	green	no	…	low
$\boldsymbol{x}^3$	red	N/A	…	high
...	…	…	…	…

Table 12.1

Article

Hard and Soft EM in Bayesian Network Learning from Incomplete Data

Andrea Ruggieri [1,†], Francesco Stranieri [1,†], Fabio Stella [1] and Marco Scutari [2,*]

[1] Department of Informatics, Systems and Communication, Università degli Studi di Milano-Bicocca, 20126 Milano MI, Italy; a.ruggieri4@campus.unimib.it (A.R.); f.stranieri1@campus.unimib.it (F.S.); fabio.stella@unimib.it (F.S.)

[2] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), 6962 Viganello, Switzerland

* Correspondence: scutari@idsia.ch

† These authors contributed equally to this work.

Received: 18 November 2020; Accepted: 7 December 2020; Published: 9 Decemb

Abstract: Incomplete data are a common feature in many domains, fror applications. Bayesian networks (BNs) are often used in these domains because of their graphical and causal interpretations. BN parameter learning from incomplete data is usually implemented with the Expectation-Maximisation algorithm (EM), which computes the relevant sufficient statistics ("soft EM") using belief propagation. Similarly, the Structural Expectation-Maximisation algorithm (Structural EM) learns the network structure of the BN from those sufficient statistics using algorithms designed for complete data. However, practical implementations of parameter and structure learning

In the **CONSTRAINT-BASED APPROACH**, the DAG $\mathcal{G}$ of a causal network is considered as an encoding of a set of (conditional) dependence and independence relations $\mathcal{M}_{\mathcal{G}}$, which can be read off $\mathcal{G}$ using **D-SEPARATION**.

D-SEPARATION

A path p is blocked by a set of nodes $\mathbf{S}$ if and only if

1) p contains a chain of nodes $A \longrightarrow B \longrightarrow C$ or a fork $A \longleftarrow B \longrightarrow C$ such that the middle node B is in $\mathbf{S}$ (i.e., is conditioned on),
2) or p contains a collider $A \longrightarrow B \longleftarrow C$ such that the collision node B is not in $\mathbf{S}$, and no descendant of B is in $\mathbf{S}$.

If $\mathbf{S}$ blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on $\mathbf{S}$, and thus are independent conditional on $\mathbf{S}$.

- Structure learning is then the task of identifying a DAG structure that (best) encodes a set of (conditional) dependence and independence relations $\mathcal{M}_{\mathcal{G}}$.
- The set of (conditional) dependence and independence relations $\mathcal{M}_{\mathcal{G}}$ may, for instance, be derived from **OBSERVATIONAL DATA** by statistical tests.
- Based on $\mathfrak{D}$ alone, we can at most hope to identify an **EQUIVALENCE CLASS OF GRAPHS** encoding the (conditional) dependence and independence relations $\mathcal{M}_{\mathcal{G}}$ of the generating distribution P.

A **CONSTRAINT-BASED STRUCTURE LEARNING ALGORITHM** proceeds by determining the validity of independence relations of the form:

$I(X, Y|S_{XY})$ we check whether X is independent of Y given subset S_{XY}, where $X, Y \in \mathbf{X}$ and $S_{XY} \subseteq \mathbf{X}$

- The structure learning algorithm will work with any information source able to provide such information.
- We will consider the case where the validity of independence relations is determined by **STATISTICAL HYPOTHESIS TESTS OF INDEPENDENCE** based on a database of cases (**OBSERVATIONAL DATA**).

The size of the space of possible DAGs grows **SUPER-EXPONENTIALLY** with the number of nodes n in the graph.

The following recursive formula gives the number $f(n)$ of DAGs on n nodes:

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

n	#dags
1	1
2	3
3	25
4	543
5	29,281
6	3,781,503
7	$1.1\ 10^9$
8	$7.8\ 10^{11}$
9	$1.2\ 10^{15}$
10	$4.2\ 10^{18}$

Table 12.2

Under the conditions listed below:

1) The independence relationships have a perfect representation as a DAG (**ACYCLICITY AND FAITHFULNESS ASSUMPTIONS**).

2) No hidden (latent) variables are involved (**CAUSAL SUFFICIENCY ASSUMPTION**).

3) The database (**OBSERVATIONAL DATA**) consists of a set of independent and identically distributed cases.

4) The database (**OBSERVATIONAL DATA**) is infinitely large.

5) The statistical tests have no error.

a constraint-based structure learning algorithm discovers a DAG structure equivalent to the DAG structure of P.

- Two DAGs representing the same set of (conditional) dependence and independence relations are equivalent in the sense that they can capture the same set of probability distributions.
- The equivalence class of a DAG $\mathcal{G}$ is the set of DAGs with the same set of d-separation relations as $\mathcal{G}$.
- A PDAG — an acyclic, partially directed graph, i.e., an acyclic graph with some edges undirected (also known as a pattern or **ESSENTIAL GRAPH**) — can be used to represent the equivalence class of a set of DAG structures, i.e., a maximal set of DAGs with the same set of d-separation relations.

EQUIVALENT MODELS AND MARKOV EQUIVALENCE CLASS

Any two models $M_{\mathcal{G}'}$ and $M_{\mathcal{G}''}$ over the same set of variables, whose graphs $\mathcal{G}'$ and $\mathcal{G}''$, respectively, have the same skeleton $\mathcal{G}_S$ (i.e., undirected graph obtained by replacing directed edges with undirected edges) and the same v-structures, are **EQUIVALENT.**

Two graphs $\mathcal{G}'$ and $\mathcal{G}''$ are in the same **EQUIVALENCE CLASS**

- if they share a **COMMON SKELETON**—that is, if they possess the same edges, regardless of the direction of those edges—and
- if they share **COMMON V-STRUCTURES**, that is, colliders whose parents are not adjacent.

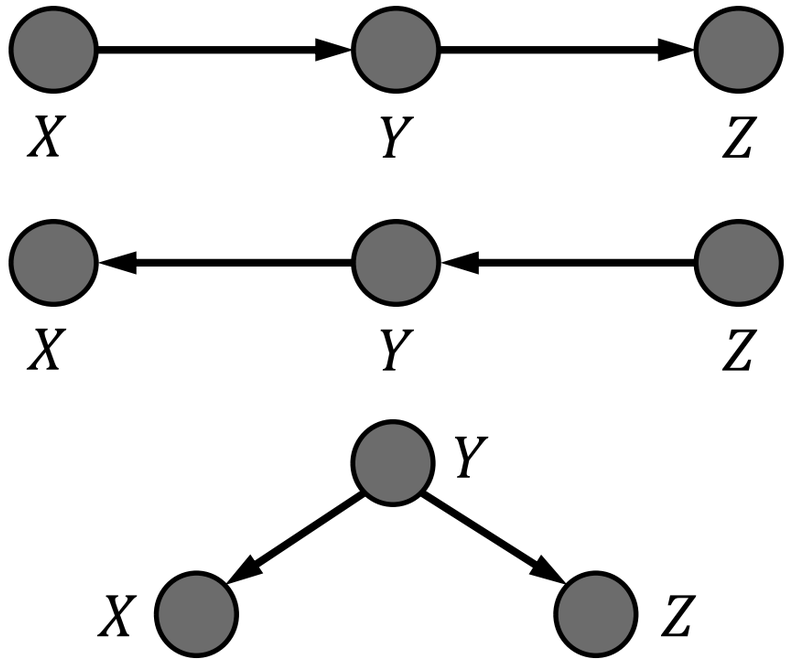


Figure 12.4

The three graphs in Figure 12.4 are **MARKOV EQUIVALENT.**

Hence, based on data $\mathfrak{D} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ alone we cannot distinguish between these three models, while we can distinguish them from the collider in Figure 12.6.

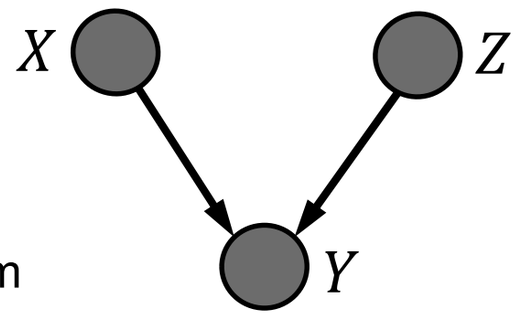


Figure 12.6

- common skeleton
- no v-structures

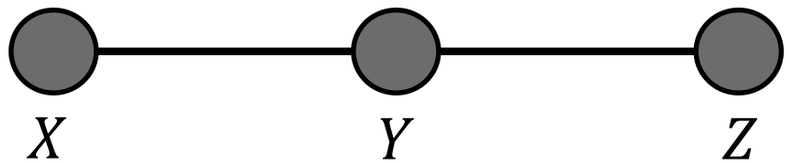


Figure 12.5

The graphs in Figure 12.4 correspond to the same set of independence/dependence assumptions $\mathcal{M}_{\mathcal{G}}$, i.e., we say that the three graphs are **MARKOV EQUIVALENT**.

Given a graph, we refer to its **MARKOV EQUIVALENCE CLASS** as the set of graphs that encode the same (conditional) independencies.

$X \perp\!\!\!\perp Z$ X is independent on Z

$X \not\perp\!\!\!\perp Z$ X is not independent on Z

$X \perp\!\!\!\perp Z \mid Y$ X is independent on Z given Y

$X \not\perp\!\!\!\perp Z \mid Y$ X is not independent on Z given Y

$\mathcal{M}_{\mathcal{G}}$

$X \not\perp\!\!\!\perp Y \quad X \not\perp\!\!\!\perp Z \quad Y \not\perp\!\!\!\perp Z$

$X \perp\!\!\!\perp Z \mid Y \qquad Z \perp\!\!\!\perp X \mid Y$

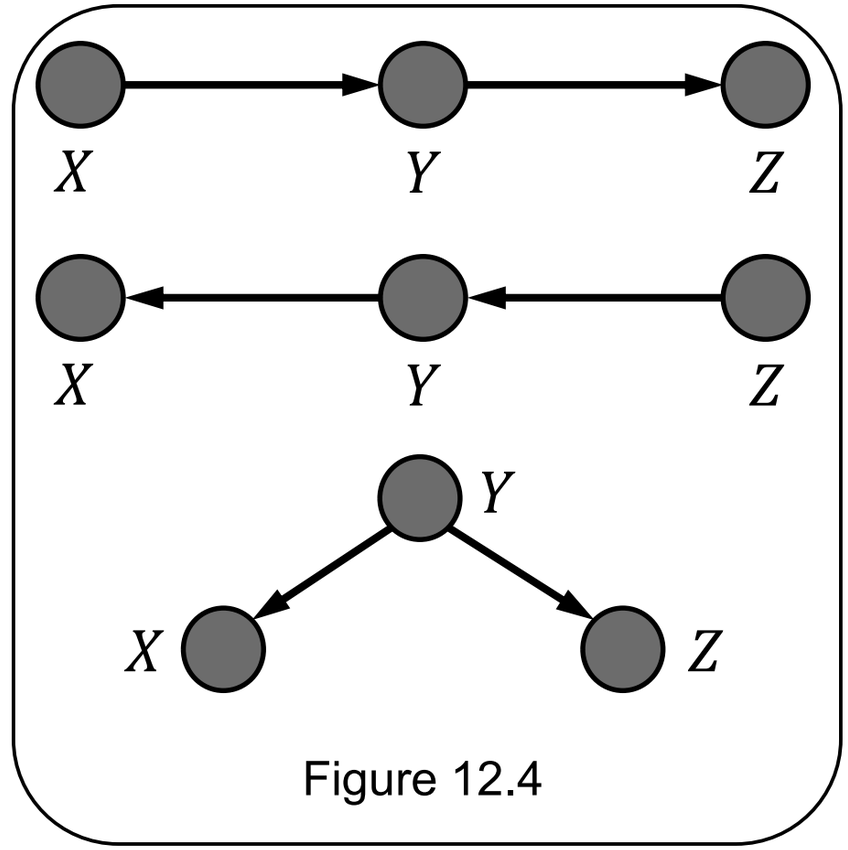
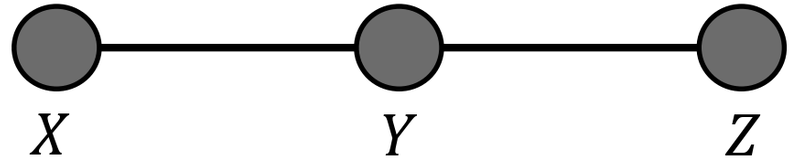


Figure 12.4

The three graphs can not be distinguished, when all we have is observational data!!!

The three graphs are **MARKOV EQUIVALENT**

- common skeleton
- no v-structures



EQUIVALENCE CLASS

An equivalence class is a maximal set of DAGs with the same set of independence properties $\mathcal{M}_{\mathcal{G}}$.

The three DAGs in Figure 12.7 all represent the same set of conditional independence and dependence relations.

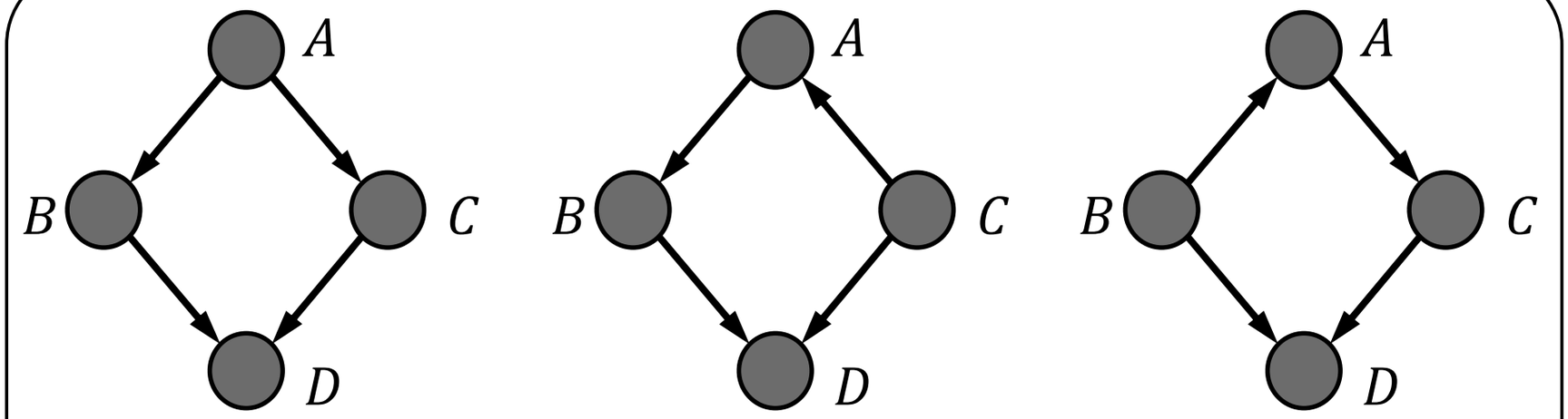


Figure 12.7

Three equivalent DAGs

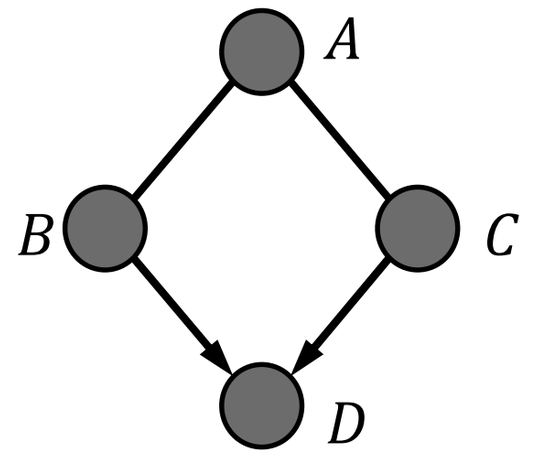


Figure 12.8

Equivalence class of the three DAGs to the left

A set of (conditional) dependence and independence relations $\mathcal{M}_{\mathcal{G}}$ may be generated by statistical tests on the **OBSERVATIONAL DATA.**

In each test, the hypothesis tested is that of independence between a pair of variables.

Let X and Y be a pair of variables for which we would like to determine dependence by **STATISTICAL HYPOTHESIS TESTING.**

We could:

- test for **MARGINAL INDEPENDENCE** and subsequently,
- test for **CONDITIONAL INDEPENDENCE** given subsets of other variables.

	X_1	X_2	...	X_n
$\boldsymbol{x}^1$	blue	yes	…	low
$\boldsymbol{x}^2$	green	no	…	low
$\boldsymbol{x}^3$	red	N/A	…	high
...	…	…	…	…

Table 12.1

A set of (conditional) dependence and independence relations $\mathcal{M}_{\mathcal{G}}$ may be generated by statistical tests on the **OBSERVATIONAL DATA.**

In each test, the hypothesis tested is that of independence between a pair of variables.

Let X and Y be a pair of variables for which we would like to determine dependence by **STATISTICAL HYPOTHESIS TESTING.**

We could:

- test for **MARGINAL INDEPENDENCE** and subsequently,
- test for **CONDITIONAL INDEPENDENCE** given subsets of other variables.

In the case of **MARGINAL INDEPENDENCE TESTING** between X and Y, the **HYPOTHESIS TO BE TESTED** is

$$H_0 : P(X,Y) = P(X)\,P(Y) \qquad X \perp\!\!\!\perp_P Y$$

$$H_1 : P(X,Y) \neq P(X)\,P(Y)$$

	X_1	X_2	...	X_n
$\boldsymbol{x}^1$	blue	yes	…	low
$\boldsymbol{x}^2$	green	no	…	low
$\boldsymbol{x}^3$	red	N/A	…	high
...	…	…	…	…

Table 12.1

A potential hypothesis test

Under H_0 the likelihood statistic

$$G^2 = 2\sum_{x,y} N_{xy}\, log\left(\frac{N_{xy}}{\mathbb{E}_{xy}}\right) \qquad \mathbb{E}_{xy} = \frac{N_x N_y}{N}$$

has an asymptotic χ^2 distribution with the appropriate degrees of freedom (df).

N_{xy}, number of cases where $X = x$ and $Y = y$

N_x, number of cases where $X = x$

N_y, number of cases where $Y = y$

A set of (conditional) dependence and independence relations $\mathcal{M}_{\mathcal{G}}$ may be generated by statistical tests on the **OBSERVATIONAL DATA.**

In each test, the hypothesis tested is that of independence between a pair of variables.

Let X and Y be a pair of variables for which we would like to determine dependence by **STATISTICAL HYPOTHESIS TESTING.**

We could:

- test for **MARGINAL INDEPENDENCE** and subsequently,
- test for **CONDITIONAL INDEPENDENCE** given subsets of other variables.

In the case of **CONDITIONAL INDEPENDENCE TESTING** between X and Y given subset S_{XY} the **HYPOTHESIS TO BE TESTED** is

$$H_0 : P(X, Y|S_{XY}) = P(X|S_{XY})\, P(Y|S_{XY}) \quad X \perp\!\!\!\perp_P Y \,|S_{XY}$$

$$H_1 : P(X, Y|S_{XY}) \neq P(X|S_{XY})\, P(Y|S_{XY})$$

	X_1	X_2	...	X_n
$\boldsymbol{x}^1$	blue	yes	…	low
$\boldsymbol{x}^2$	green	no	…	low
$\boldsymbol{x}^3$	red	N/A	…	high
...	…	…	…	…

Table 12.1

A potential hypothesis test

Under H_0 the likelihood statistic

$$G^2 = 2 \sum_{x,y,s} N_{xys}\, log\left(\frac{N_{xys}}{\mathbb{E}_{xys}}\right) \quad \mathbb{E}_{xys} = \frac{N_{xs} N_{ys}}{N_s}$$

has an asymptotic χ^2 distribution with the appropriate degrees of freedom (df).

$$df = (\|X\| - 1)(\|Y\| - 1) \prod_{s \in S_{XY}} \|Y\|$$

PART II

THE PC ALGORITHM

The **PC ALGORITHM** (Spirtes & Glymour 1991, Spirtes et al. 2000) is probably the most known **CONSTRAINT-BASED ALGORITHM** for learning the structure of a causal network.

The **MAIN STEPS OF THE PC ALGORITHM** are:

1) Test for (conditional) independence between each pair of variables represented in $\mathfrak{D} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ to derive $\mathcal{M}_{\mathfrak{D}}$, the set of conditional independence and dependence relations.
2) Identify the skeleton of the graph induced by $\mathcal{M}_{\mathfrak{D}}$.
3) Identify colliders.
4) Identify derived directions.

The **PC ALGORITHM** typically produces a **PDAG** (Partially DAG) representing an equivalence class as it emerges from hypothesis testing performed by using the available observational data $\mathfrak{D} = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$.

STEP 1: **TEST FOR (CONDITIONAL) INDEPENDENCE.** We try to determine the validity of the conditional independence statement

$$X \perp\!\!\!\perp_P Y \mid S_{XY}$$

$$H_0 : P(X, Y|S_{XY}) = P(X|S_{XY})P(Y|S_{XY})$$
$$H_1 : P(X, Y|S_{XY}) \neq P(X|S_{XY})P(Y|S_{XY})$$

The independence hypothesis H_0 is tested for **CONDITIONING SETS S_{XY}** of cardinality 0, 1, 2, 3, … in that order.

If the hypothesis H_0 cannot be rejected based on some preselected **SIGNIFICANCE LEVEL α**, then the search for an independence relation between X and Y is terminated.

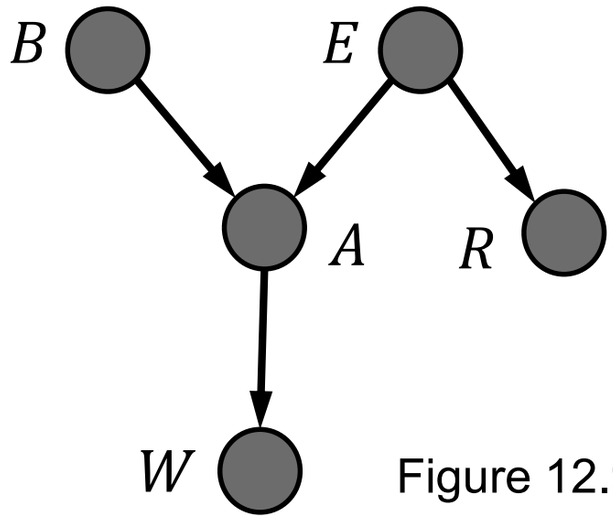


Figure 12.9

Assume you are given the database of cases (**OBSERVATIONAL DATA**) which has been generated from the model in Figure 12.9, and that you got the following (conditional) independence and dependence relations:

$$\mathcal{M}_{\mathcal{D}} \begin{cases} \mathcal{M}_{\perp\!\!\!\perp} = \{B \perp\!\!\!\perp E, B \perp\!\!\!\perp R, B \perp\!\!\!\perp W|A, A \perp\!\!\!\perp R|E, E \perp\!\!\!\perp W|A, R \perp\!\!\!\perp W|A\} & \leftarrow \text{independencies} \\ \mathcal{M}_{\not\perp\!\!\!\perp} = \{B \not\perp\!\!\!\perp A, B \not\perp\!\!\!\perp A|\{E\}, B \not\perp\!\!\!\perp A|\{R\}, B \not\perp\!\!\!\perp A|\{W\}, B \not\perp\!\!\!\perp A|\{E, R\}, \\ \quad B \not\perp\!\!\!\perp A|\{E, W\}, B \not\perp\!\!\!\perp A|\{R, W\}, B \not\perp\!\!\!\perp A|\{E, R, W\}, A \not\perp\!\!\!\perp E, \ldots \\ \quad A \not\perp\!\!\!\perp W, \ldots, E \not\perp\!\!\!\perp R, \ldots\}. & \leftarrow \text{dependencies} \end{cases}$$

STEP 2: **IDENTIFY THE SKELETON.** The skeleton of an acyclic, directed or partially directed graph $\mathcal{G}$ is the undirected graph $\mathcal{G}^u$ obtained from $\mathcal{G}$ by removing the direction on all directed edges.

The skeleton of the graph induced from $\mathcal{M}_{\mathfrak{D}}$ is constructed from the conditional dependence and independence statements of $\mathcal{M}_{\mathfrak{D}}$ generated by the statistical test in **STEP 1**.

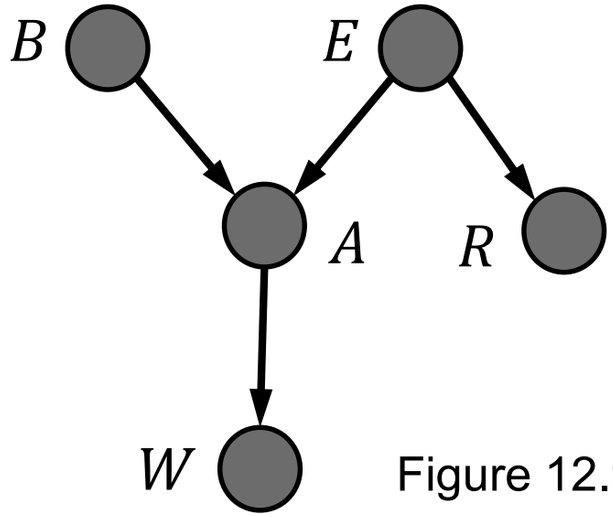


Figure 12.9

For each pair of variables X and Y where no independence statement

$$X \perp\!\!\!\perp_P Y \,|S_{XY}$$

exists, the undirected edge (X, Y) is created in the skeleton.

The graph of Figure 12.10 is a more intuitive and compact representation of the dependence and independence model than that of equations below.

$$\mathcal{M}_{\mathfrak{D}} \begin{cases} \mathcal{M}_{\perp\!\!\!\perp} = \{\mathrm{B} \perp\!\!\!\perp \mathrm{E}, \mathrm{B} \perp\!\!\!\perp \mathrm{R}, \mathrm{B} \perp\!\!\!\perp W|\mathrm{A}, \mathrm{A} \perp\!\!\!\perp \mathrm{R}|\mathrm{E}, \mathrm{E} \perp\!\!\!\perp W|\mathrm{A}, \mathrm{R} \perp\!\!\!\perp W|\mathrm{A}\} \\ \mathcal{M}_{\not\perp\!\!\!\perp} = \{\mathrm{B} \not\perp\!\!\!\perp \mathrm{A}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{R}\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}, \mathrm{R}\}, \\ \quad \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}, W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{R}, W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}, \mathrm{R}, W\}, \mathrm{A} \not\perp\!\!\!\perp \mathrm{E}, \ldots \\ \quad \mathrm{A} \not\perp\!\!\!\perp W, \ldots, \mathrm{E} \not\perp\!\!\!\perp \mathrm{R}, \ldots\}. \end{cases}$$

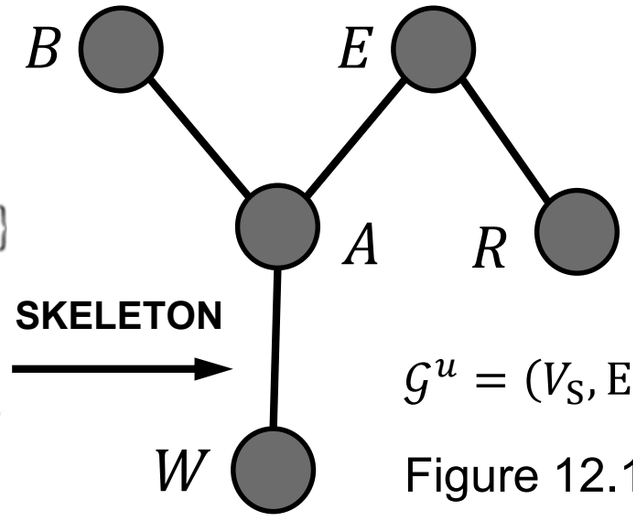


Figure 12.10

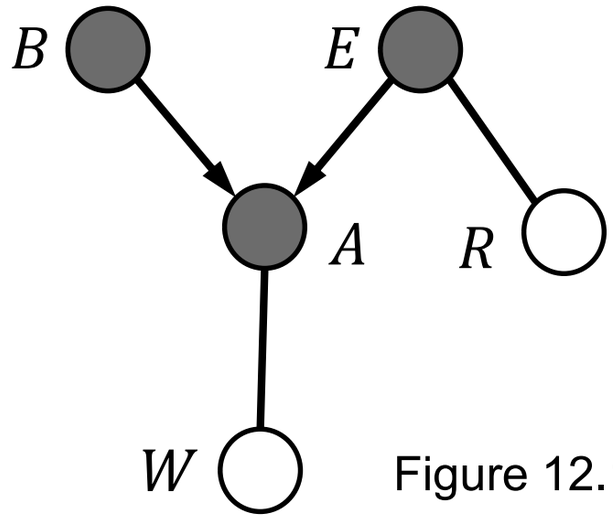
STEP 3: **IDENTIFY COLLIDERS**. Once the skeleton has been identified, colliders in the skeleton are identified.

Based on the skeleton, we search for subsets of variables $\{X, Y, Z\}$ such that X and Y are neighbors, Z and Y are neighbors while X and Z are not neighbors.

For each such subset a collider $X \rightarrow Y \leftarrow Z$ is created when $Y \notin S_{XZ}$ for any S_{XZ} satisfying

$$X \perp\!\!\!\perp_P Z \,|S_{XZ}$$

in $\mathcal{M}_{\mathfrak{D}}$.



$B \rightarrow A \leftarrow E$

Figure 12.11

$(B, A), (E, A) \in \mathrm{E_S}$

$(B, E) \notin \mathrm{E_S}$

$\boxed{\forall\, B \perp\!\!\!\perp_P E \,|S_{BE}}$ $S_{BE} = \{\emptyset\}$ $A \notin S_{BE}$

$$\mathcal{M}_{\mathfrak{D}} \begin{cases} \mathcal{M}_{\perp\!\!\!\perp} = \boxed{\mathrm{B} \perp\!\!\!\perp \mathrm{E}}\, \mathrm{B} \perp\!\!\!\perp \mathrm{R}, \mathrm{B} \perp\!\!\!\perp W|\mathrm{A}, \mathrm{A} \perp\!\!\!\perp \mathrm{R}|\mathrm{E}, \mathrm{E} \perp\!\!\!\perp W|\mathrm{A}, \mathrm{R} \perp\!\!\!\perp W|\mathrm{A}\} \\ \mathcal{M}_{\not\perp\!\!\!\perp} = \{\mathrm{B} \not\perp\!\!\!\perp \mathrm{A}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{R}\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}, \mathrm{R}\}, \\ \quad \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}, W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{R}, W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A}|\{\mathrm{E}, \mathrm{R}, W\}, \mathrm{A} \not\perp\!\!\!\perp \mathrm{E}, \ldots \\ \quad \mathrm{A} \not\perp\!\!\!\perp W, \ldots, \mathrm{E} \not\perp\!\!\!\perp \mathrm{R}, \ldots\}. \end{cases}$$

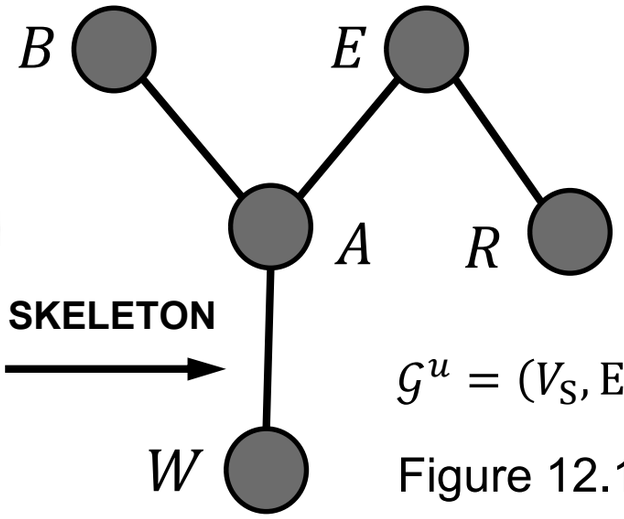


Figure 12.10

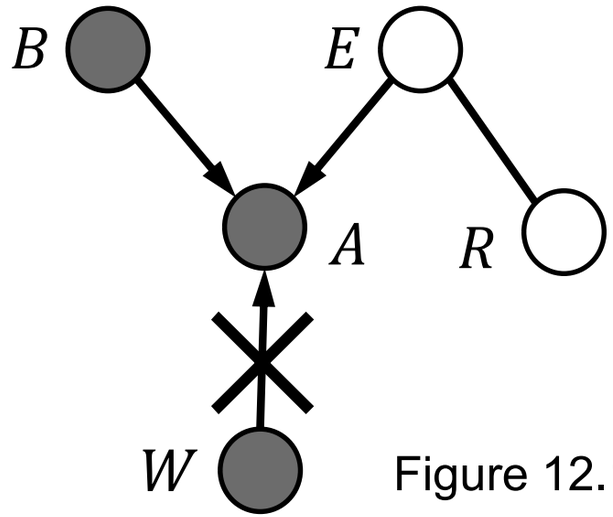
STEP 3: **IDENTIFY COLLIDERS**. Once the skeleton has been identified, colliders in the skeleton are identified.

Based on the skeleton, we search for subsets of variables $\{X, Y, Z\}$ such that X and Y are neighbors, Z and Y are neighbors while X and Z are not neighbors.

For each such subset a collider $X \rightarrow Y \leftarrow Z$ is created when $Y \notin S_{XZ}$ for any S_{XZ} satisfying

$$X \perp\!\!\!\perp_P Z \mid S_{XZ}$$

in $\mathcal{M}_{\mathfrak{D}}$.



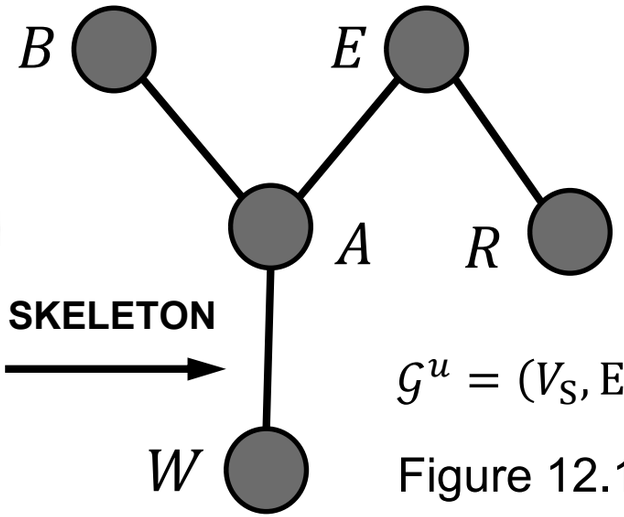
$$B \rightarrow A \leftarrow W$$

Figure 12.12

$$(B, A), (W, A) \in \mathrm{E_S} \qquad (B, W) \notin \mathrm{E_S}$$

$$\forall\, B \perp\!\!\!\perp_P W \mid S_{BW} \qquad S_{BW} = \{A\} \qquad A \in S_{BW}$$

$$\mathcal{M}_{\mathfrak{D}} \begin{cases} \mathcal{M}_{\perp\!\!\!\perp} = \{\mathrm{B} \perp\!\!\!\perp \mathrm{E}, \mathrm{B} \perp\!\!\!\perp \mathrm{R}, \boxed{\mathrm{B} \perp\!\!\!\perp W \mid \mathrm{A}}\ \mathrm{A} \perp\!\!\!\perp \mathrm{R} \mid \mathrm{E}, \mathrm{E} \perp\!\!\!\perp W \mid \mathrm{A}, \mathrm{R} \perp\!\!\!\perp W \mid \mathrm{A}\} \\ \mathcal{M}_{\not\perp\!\!\!\perp} = \{\mathrm{B} \not\perp\!\!\!\perp \mathrm{A}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A} \mid \{\mathrm{E}\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A} \mid \{\mathrm{R}\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A} \mid \{W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A} \mid \{\mathrm{E}, \mathrm{R}\}, \\ \quad \mathrm{B} \not\perp\!\!\!\perp \mathrm{A} \mid \{\mathrm{E}, W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A} \mid \{\mathrm{R}, W\}, \mathrm{B} \not\perp\!\!\!\perp \mathrm{A} \mid \{\mathrm{E}, \mathrm{R}, W\}, \mathrm{A} \not\perp\!\!\!\perp \mathrm{E}, \ldots \\ \quad \mathrm{A} \not\perp\!\!\!\perp W, \ldots, \mathrm{E} \not\perp\!\!\!\perp \mathrm{R}, \ldots\}. \end{cases}$$



$$\mathcal{G}^u = (V_{\mathrm{S}}, \mathrm{E_S})$$

Figure 12.10

STEP 4: **IDENTIFY DERIVED DIRECTIONS.** After identifying the skeleton and the colliders of $\mathcal{G}$, derived directions are identified.

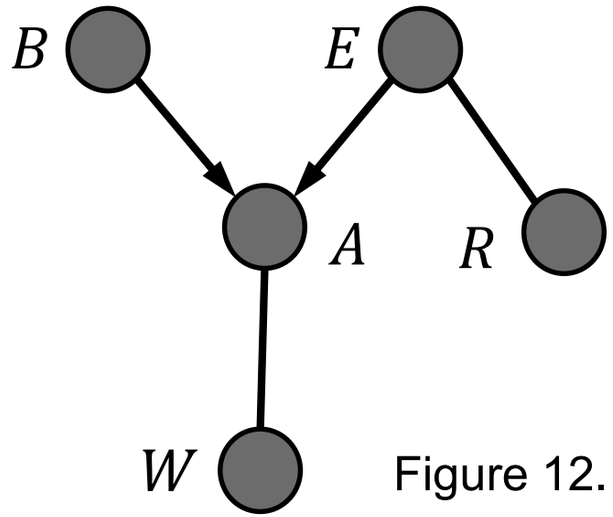


Figure 12.13

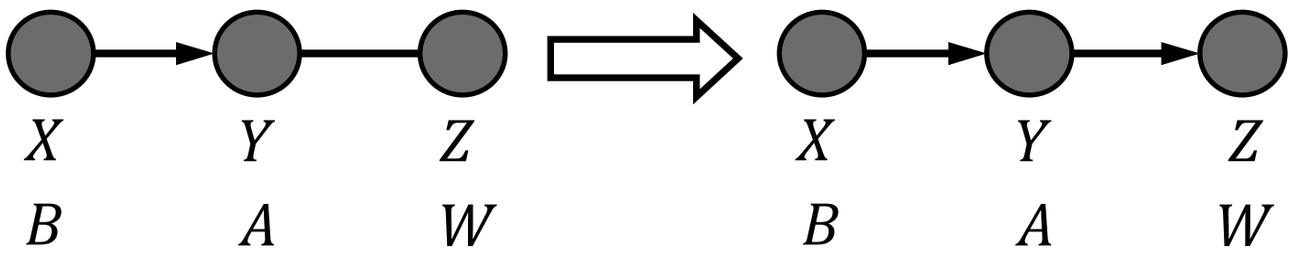
The direction of an edge is said to be derived when it is a logical consequence of (the lack of) previous actions (i.e., since the edge was not directed in a previous step and it should have been in order to have a certain direction, then the edge must be directed in the opposite direction).

Starting with any PDAG including all valid colliders, a **MAXIMALLY DIRECTED PDAG** can be obtained following four **NECESSARY AND SUFFICIENT RULES.**

That is, by repeated application of these four rules all edges common to the equivalence class of $\mathcal{G}$ are identified.

STEP 4: IDENTIFY DERIVED DIRECTIONS.

RULE 1



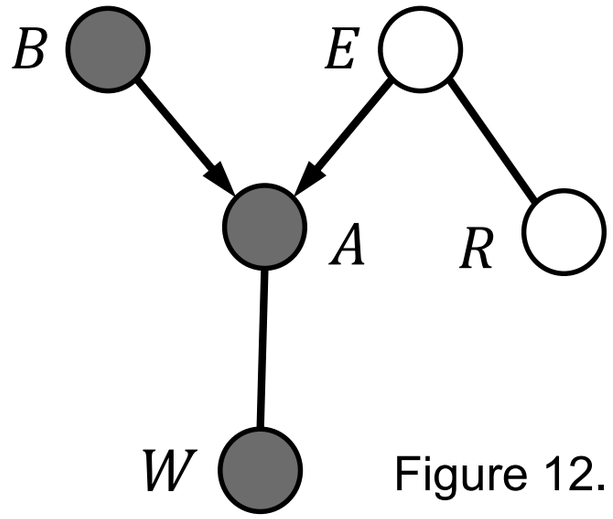


Figure 12.13

It follows from the fact that the collider

$$X \rightarrow Y \leftarrow Z$$

was not identified as a valid collider.

Since the edge between Y and Z is not part of the aforementioned collider, it must be directed from Y to Z.

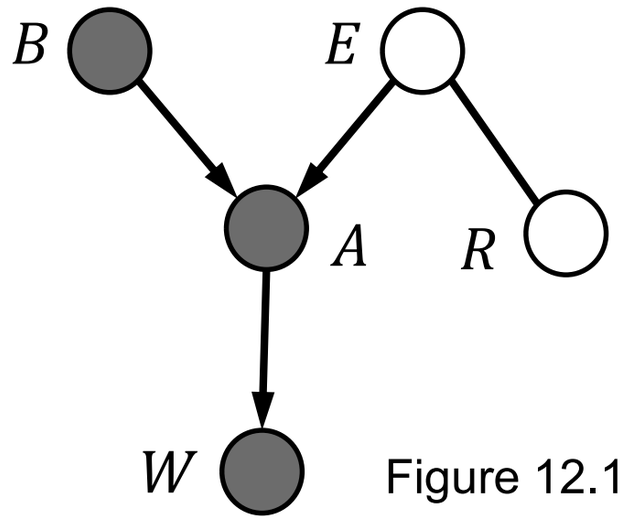
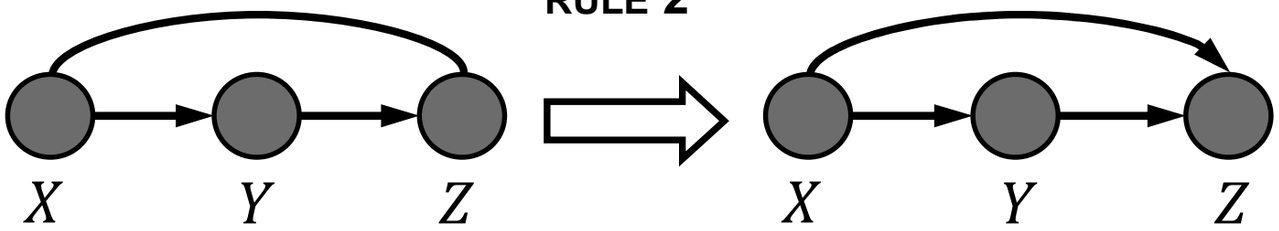


Figure 12.14

STEP 4: IDENTIFY DERIVED DIRECTIONS.

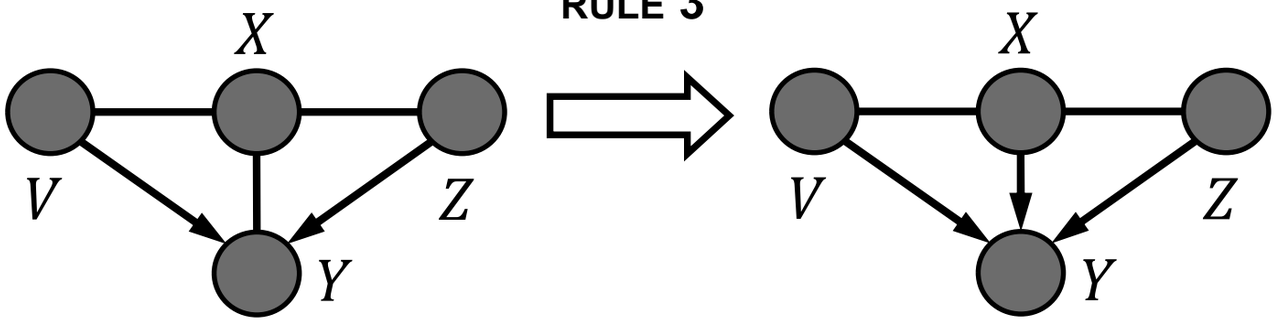
RULE 2



It follows from the fact that directing the edge between X and Z from Z to X will induce a directed cycle in the graph.

Thus, the edge must be directed from X to Z.

RULE 3



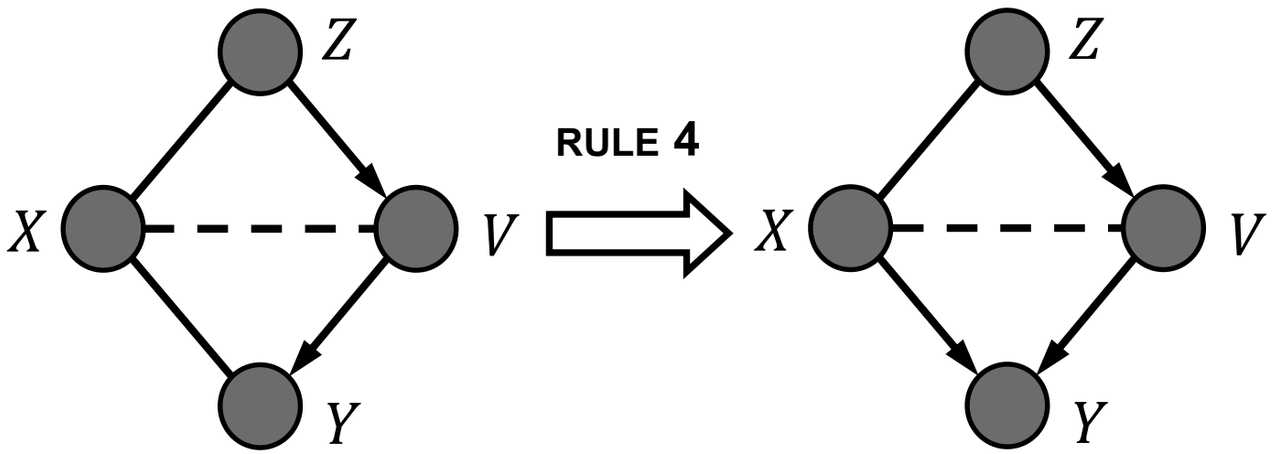
It follows from the fact that directing the edge between X and Y from Y to X will inevitably produce an additional collider

$$V \rightarrow X \leftarrow Z$$

or a directed cycle.

Hence, the edge must be directed from X to Y.

STEP 4: IDENTIFY DERIVED DIRECTIONS.



X and V are adjacent nodes, i.e., $X \rightarrow V$, $V \rightarrow X$ or $X - V$.

It follows from the fact that directing the edge between X and Y from Y to X will inevitably produce an additional collider

$$Y \rightarrow X \leftarrow Z$$

or a directed cycle.

Hence, the edge must be directed from X to Y.

The dashed lines used to illustrate the fourth rule indicate that X and V are connected by an edge (either directed or not, $X \rightarrow V$, $V \rightarrow X$ or $X - V$).

The fourth rule is not necessary if the orientation of the initial PDAG is limited to containing colliders only.

The initial PDAG may contain non-colliders when expert knowledge on edge directions are included in the graph.

STEP 4: IDENTIFY DERIVED DIRECTIONS.

As neither the collider

$$B \rightarrow A \leftarrow W$$

nor the collider

$$E \rightarrow A \leftarrow W$$

was identified as a collider of $\mathcal{G}$, the edge between A and W must be directed from A to W (Figure 12.13). (Application of **RULE 1**)

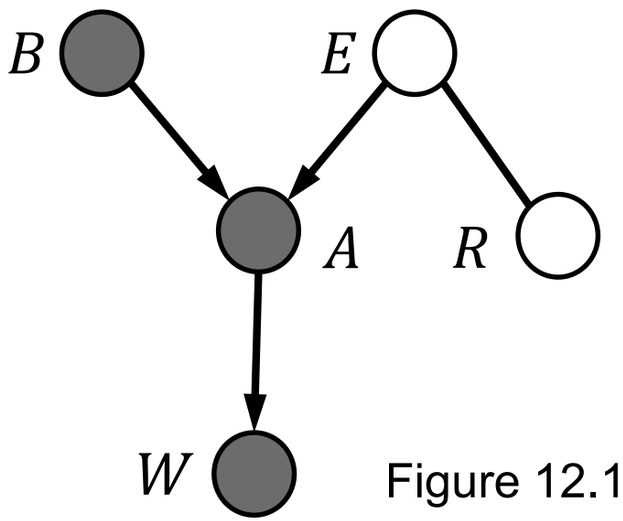


Figure 12.13

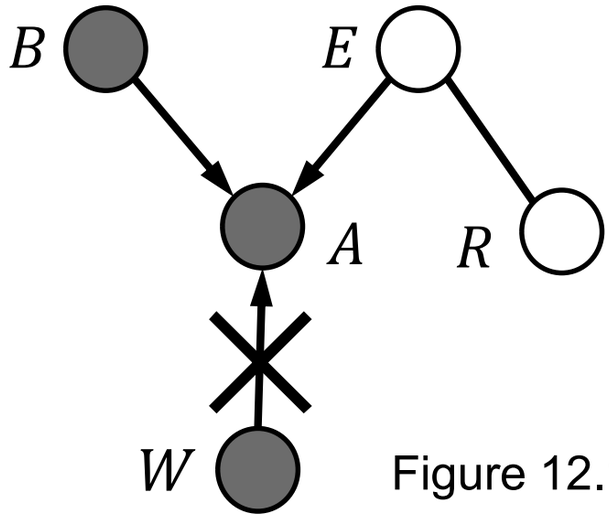


Figure 12.12

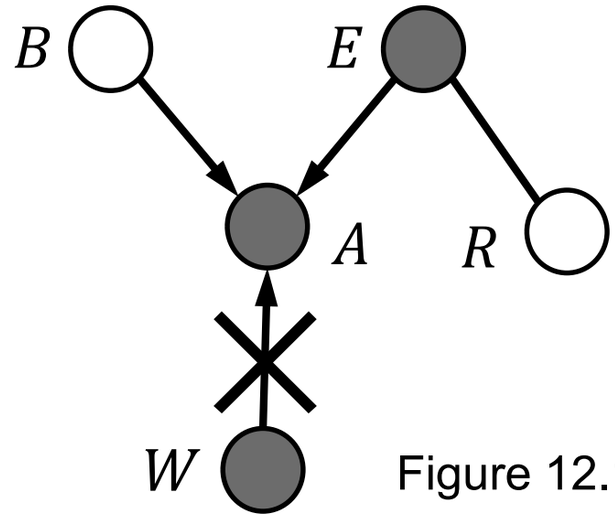


Figure 12.15

STEP 4: IDENTIFY DERIVED DIRECTIONS.

As neither the collider

$$B \rightarrow A \leftarrow W$$

nor the collider

$$E \rightarrow A \leftarrow W$$

was identified as a collider of $\mathcal{G}$, the edge between A and W must be directed from A to W (Figure 12.14). (Application of **RULE 1**)

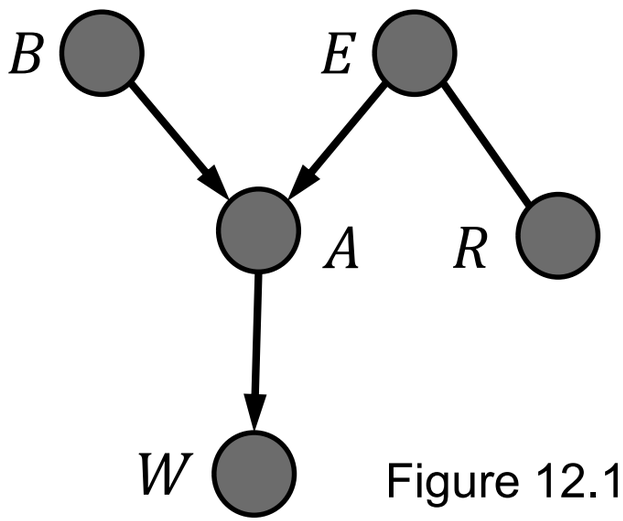


Figure 12.14

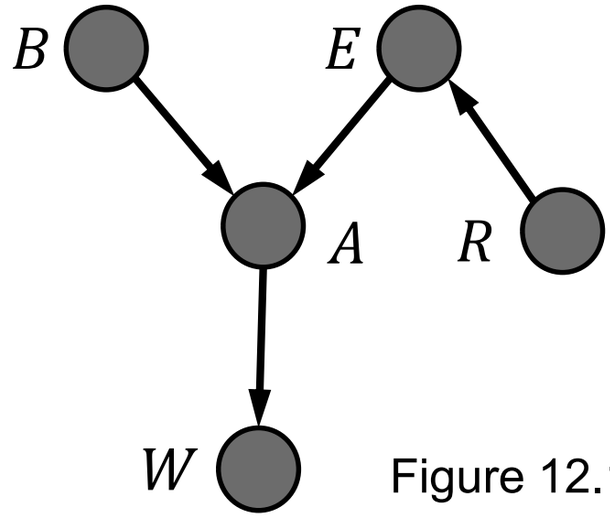


Figure 12.16

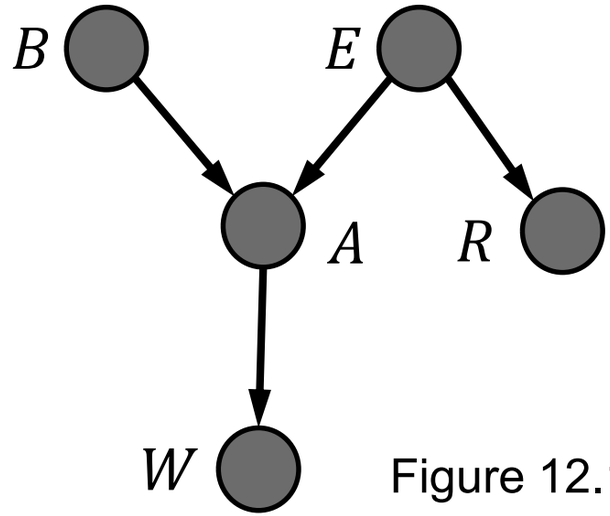


Figure 12.17

Figure 12.16 and Figure 12.17 show the **EQUIVALENCE CLASS** of $\mathcal{M}_{\mathfrak{D}}$.

The **EQUIVALENCE CLASS** of $\mathcal{M}_{\mathfrak{D}}$ contains two DAGs differing only with respect to the orientation of the edge between E and R.

- The four rules are necessary and sufficient for achieving maximal orientation (up to equivalence) of the PDAG returned by the PC algorithm.
- We use these four rules repeatedly until no edge can be given an orientation.
- Notice that the result of closing edge directions under rules from 1 to 4 is not necessarily a DAG.
- If the graph is not a DAG, then expert knowledge may be appropriate in order to direct an edge.
- Once an edge has been directed by use of expert knowledge derived directions should be identified.
- This process may be repeated until a DAG structure is obtained.
- Experience shows that most edges are directed using **RULE 1**, and that **RULE 3** is only rarely used.

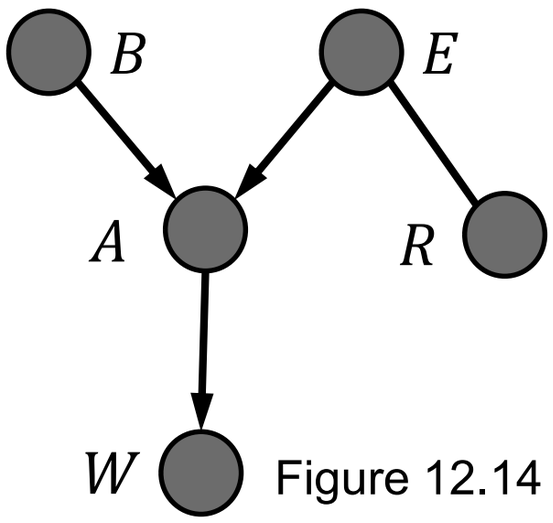


Figure 12.14

possible completions of the PDAG

The two possible completions induce the same set of conditional independence and dependence relations.

When only the database of cases is available, we can not select which is the true DAG, i.e., the one corresponding to the underlying process.

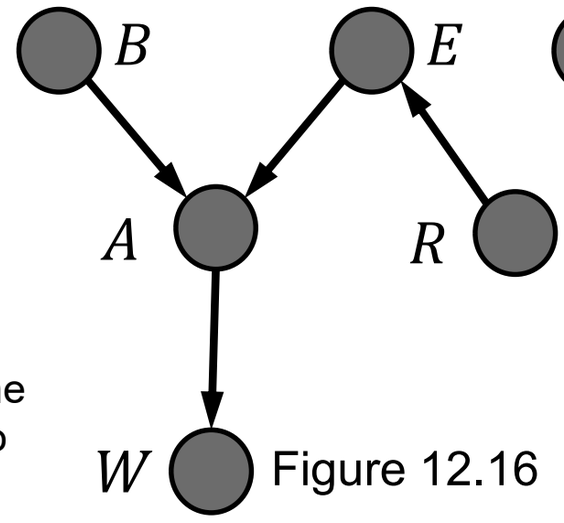


Figure 12.16

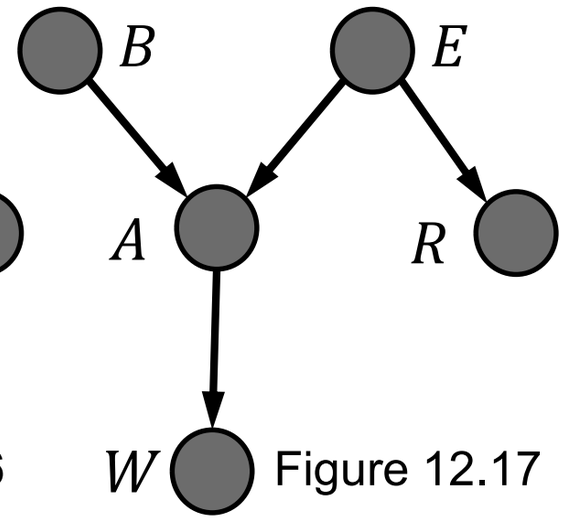


Figure 12.17

- When a unique DAG can not be obtained from the database of cases we can ask help to the domain expert.
- If we can obtain a unique DAG, then we move on to learn the parameters of the causal network.

- There are **OTHER CONSTRAINT-BASED ALGORITHMS** that allow us to drop various assumptions.
- The FCI (**FAST CAUSAL INFERENCE**) algorithm works without assuming causal sufficiency.
- The **CCD** algorithm works without assuming acyclicity.
- **CONSTRAINT-BASED ALGORITHMS** suffer from the fact that conditional independence tests are hard, and it can sometimes require a lot of data to get accurate test results.

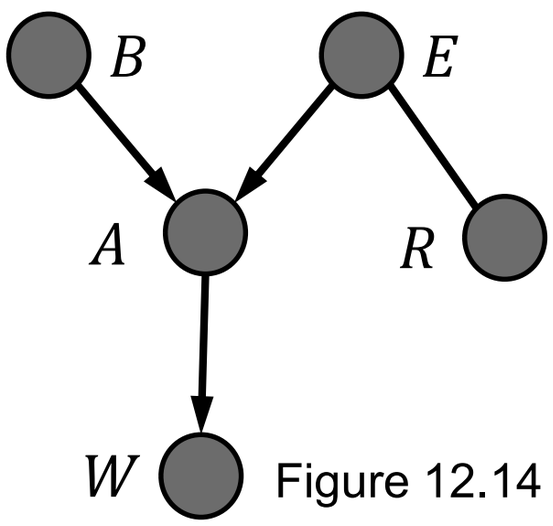


Figure 12.14

possible completions of the PDAG

The two possible completions induce the same set of conditional independence and dependence relations.

When only the database of cases is available, we can not select which is the true DAG, i.e., the one corresponding to the underlying process.

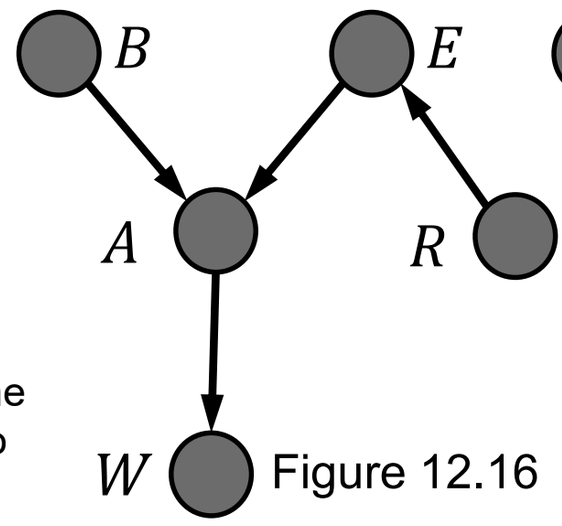


Figure 12.16

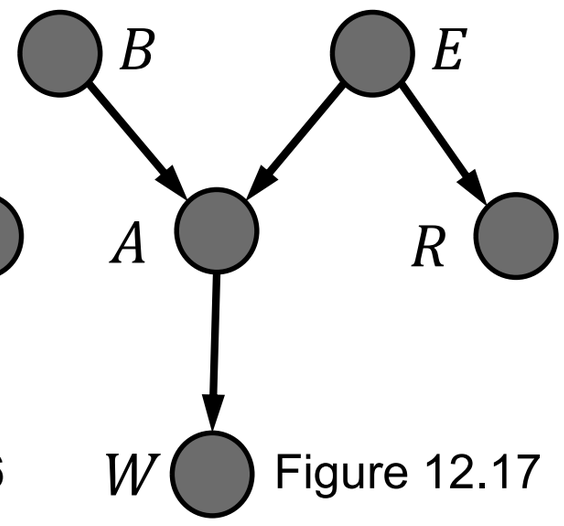


Figure 12.17

Progress in Artificial Intelligence
https://doi.org/10.1007/s13748-019-00194-y

REVIEW

A survey on Bayesian network structure learning from data

Mauro Scanagatta[1] · Antonio Salmerón[2] · Fabio Stella[3]

Received: 14 March 2019 / Accepted: 20 May 2019

Abstract
A necessary step in the development of artificial intelligence is to enable a machine to *represent* how the world works, building an internal structure from data. This structure should hold a good trade-off between expressive power and querying efficiency. Bayesian networks have proven to be an effective and versatile tool for the task at hand. They have been applied to modeling knowledge in a variety of fields, ranging from bioinformatics to law, from image processing to economic risk analysis. A crucial aspect is learning the dependency graph of a Bayesian network from data. This task, called *structure learning*, is NP-hard and is the subject of intense, cutting-edge research. In short, it can be thought of as choosing one graph over the many candidates, grounding our reasoning over a collection of samples of the distribution generating the data. The number of possible graphs increases very quickly at the increase in the number of variables. Searching in this space, and selecting a graph over the others, becomes quickly burdensome. In this survey, we review the most relevant structure learning algorithms that have been proposed in the literature. We classify them according to the approach they follow for solving the problem and we also show alternatives for handling missing data and continuous variable. An extensive review of existing software tools is also given.

Keywords Machine learning · Statistics · Bayesian network · Structure learning