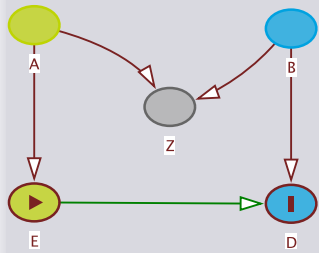


Lecture 1

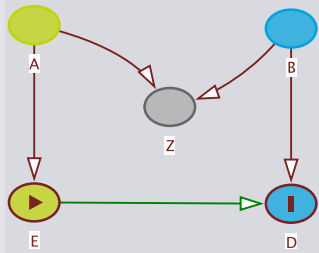
Introduction to Bayesian Networks

Course Bayesian Networks, September 3rd, 2018

Johannes Textor
Institute for Computing and Information Sciences

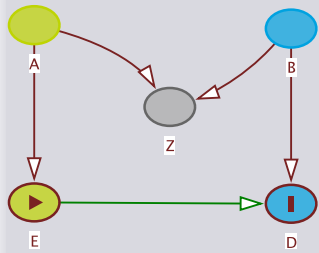
Learning Objectives 1

1. Get to know each other!
2. Know what a Bayesian network looks like.
3. Understand why Bayesian networks are needed.
4. Understand what a Bayesian network represents.



Lecture Outline 1

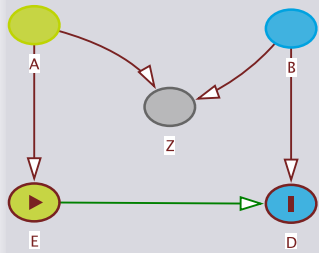
1. Hello, World!
2. Why We Need Bayesian Networks: A Paradox
3. Optimal Bayes
4. Naive Bayes
5. Bayesian Networks



Introducing Ourselves

- Johannes Textor – assistant professor at Tumor Immunology, RUMC
  johannes.textor@radboudumc.nl – or through Brightspace
- Perry Groot – postdoc at iCIS, student advisor Computing Sciences & Information Sciences
  Perry.Groot@science.ru.nl
- Ankur Ankan – fellow Master student, author of the pgmpy package
  a.ankan@cs.ru.nl

We are all active researchers in the field of probabilistic graphical models and their applications.

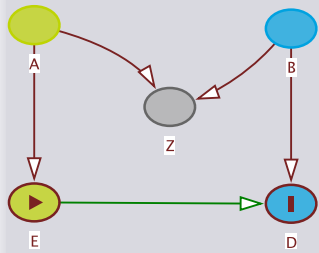


Introducing the Course

- **Lectures**
  Beware: lectures are interactive and you will be asked questiond!
- **Exercises**
  There is one planned exercise session per week.
- **Assignment I**
  Implement a Bayesian network for a real-world domain.
- **Assignment II**
  Weigh in on a current scientific issue in the field.

Assessment

Exam	50%
Assignment I	25%
Assignment II	25%

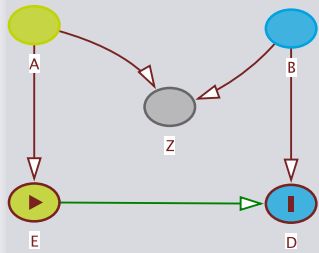


Introducing the Resources

In brief, everything is on Brightspace:
All course materials (slides, assignments, literature, . . .).
It's also the official communication channel between you and us.
And it's where you hand in your assignments.
At `brightspace.ru.nl`

Literature

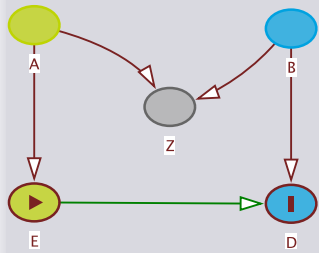
Korb & Nicholson: Bayesian Artificial Intelligence
Pearl, Glymour & Jewell: Causal Inference in Statistics: A Primer
Pearl & Mackenzie: The Book of Why



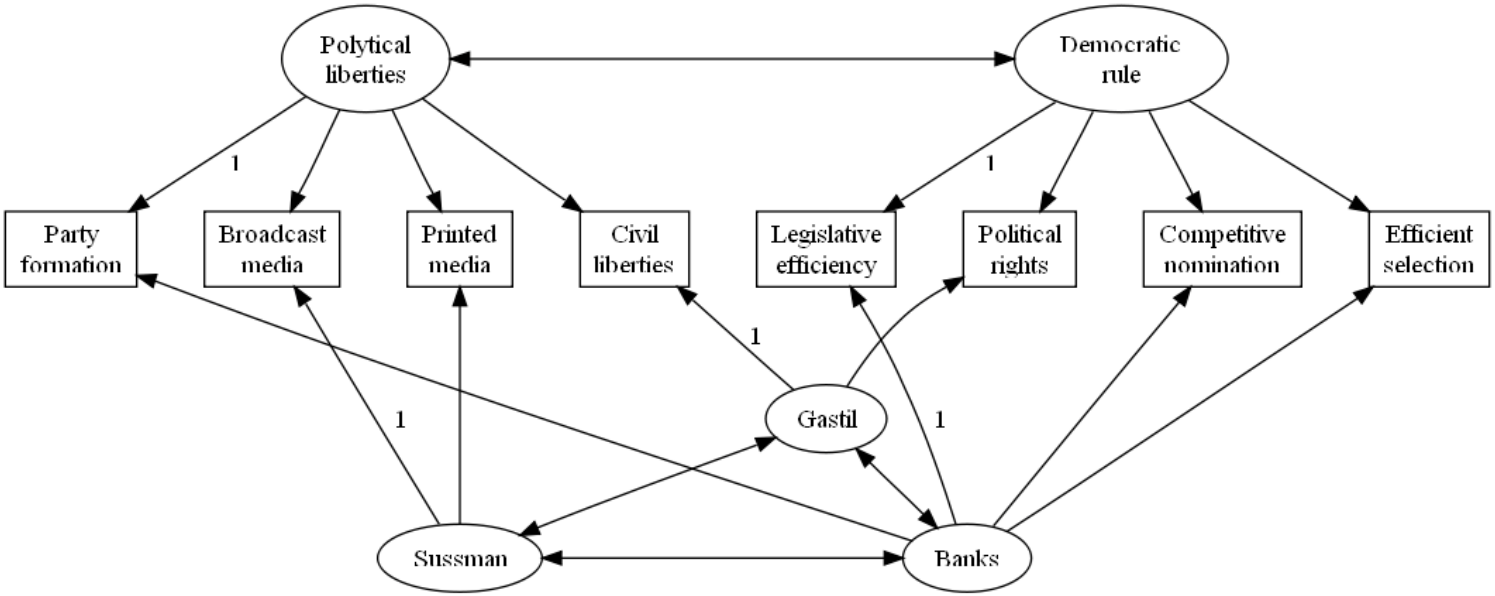
Bayesian Networks

- Bayesian networks are models of reality.
- Bayesian networks are models of knowledge.
- Bayesian networks are models of probability distributions.

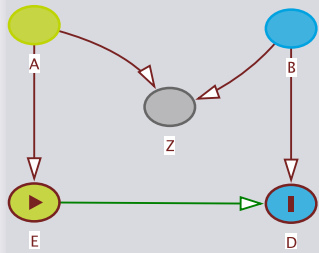
Visually, Bayesian networks are graphs consisting of nodes and edges.



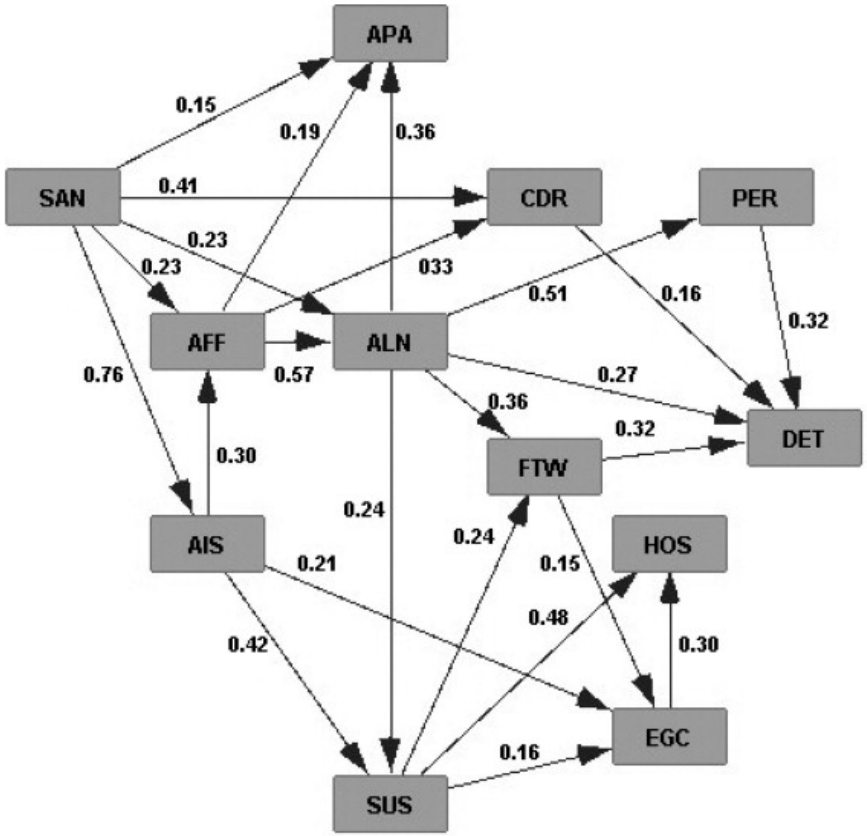
What Constitutes a Liberal Democracy?



Bollen, American Journal of Political Science 37(4):1207–30, 1993

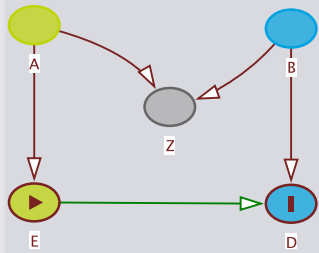


How Does Schizophrenic Disease Unfold?

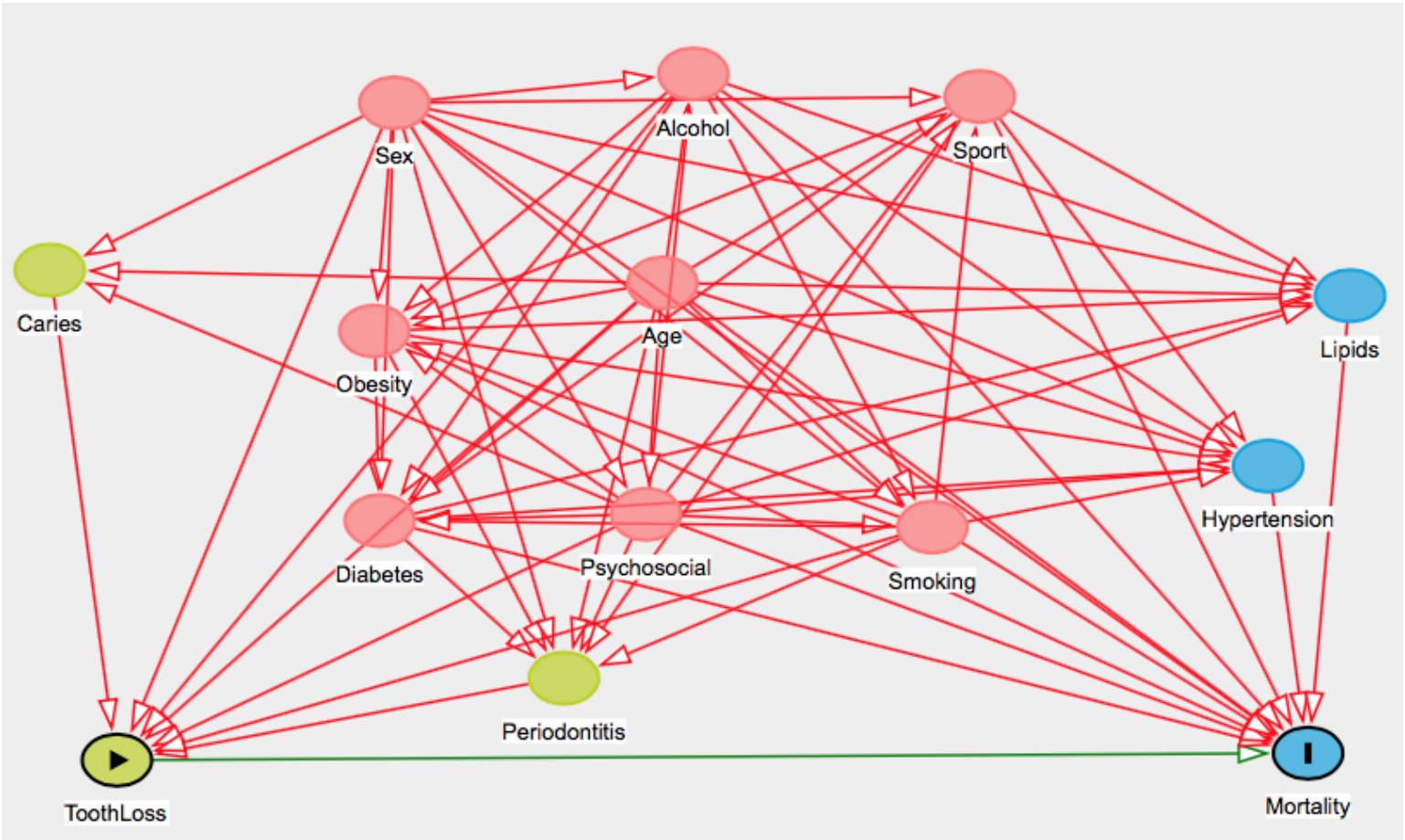


SAN: Social Anxiety;
AIS: Active Isolation;
AFF: Affective Flattening;
SUS: Suspiciousness;
EGC: Egocentrism;
FTW: Living in a Fantasy World;
ALN: Alienation;
APA: Apathy;
HOS: Hostility;
CDR: Cognitive Derailment;
PER: Perceptual Aberrations;
DET: Delusional Thinking.

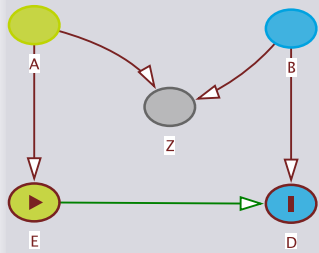
van Kampen, European Psychiatry 29(7):437–48, 2014



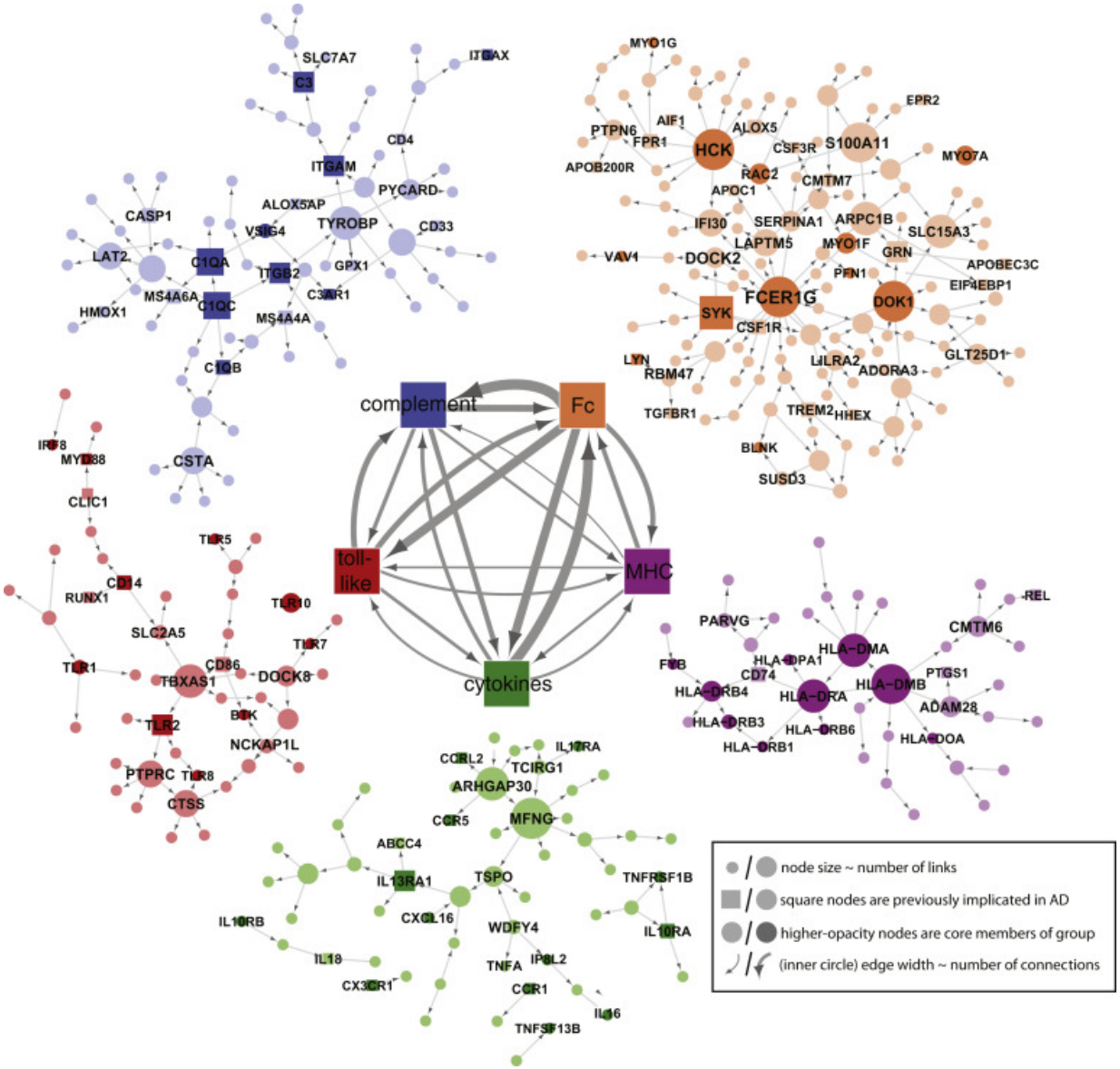
Is There a Benefit of Replaced Teeth?



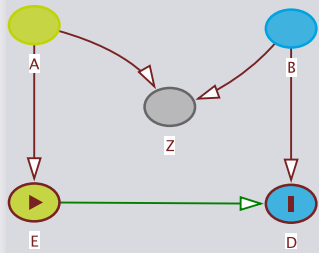
Polzer et al, Clinical Oral Investigations 16(2):333–351, 2012



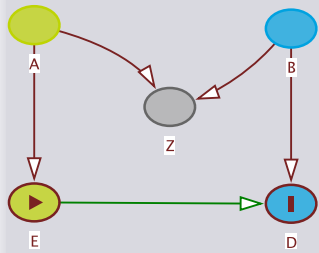
Which Genes Are Relevant in Alzheimer's Disease?



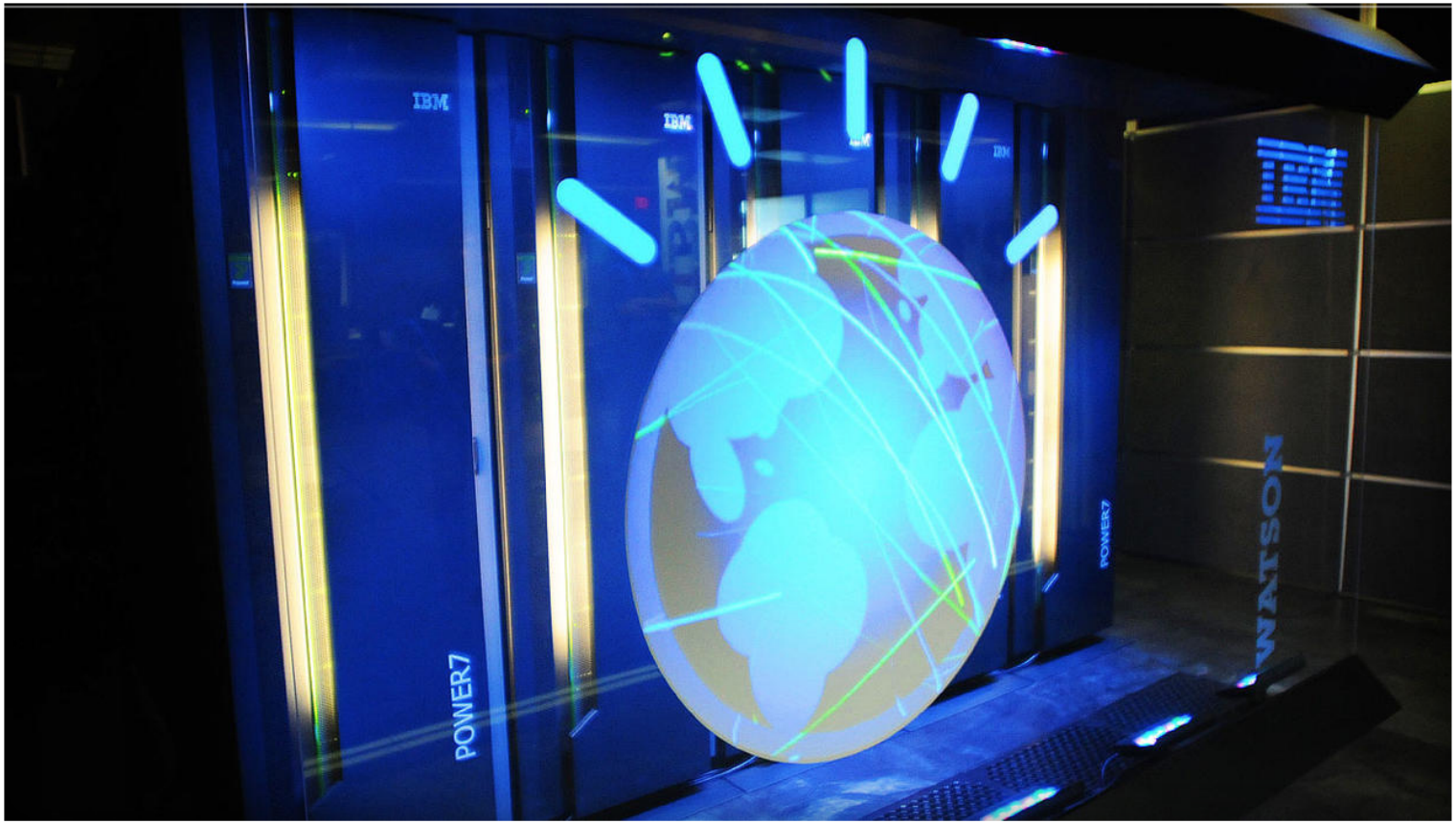
Zhang et al, Cell 153(3):702–720, 2013



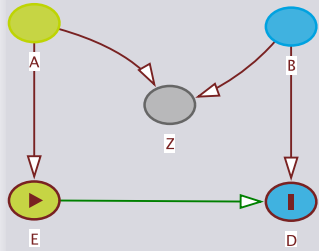
Why should I care about Bayesian networks?



Supercomputer Watson takes on cancer care with Memorial Sloan-Kettering



Caption: This Jan. 13, 2011 photo provided by IBM shows the IBM computer system known as Watson at IBM's T.J. Watson research center in Yorktown Heights, N.Y. Watson, best known for handily defeating the world's best "Jeopardy!" players on TV earlier this year, is on a diet of medical textbooks and journals for health care. IBM says Watson, with its ability to understand plain language, can digest questions about a person's symptoms and medical history and quickly suggest diagnoses and treatments. (AP Photo/IBM) / **AP**



EDITOR'S PICK | 209,668 views | Feb 19, 2017, 03:48pm

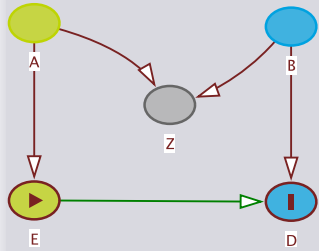
MD Anderson Benches IBM Watson In Setback For Artificial Intelligence In Medicine

Matthew Herper Forbes Staff
Pharma & Healthcare
I cover science and medicine, and believe this is biology's century.

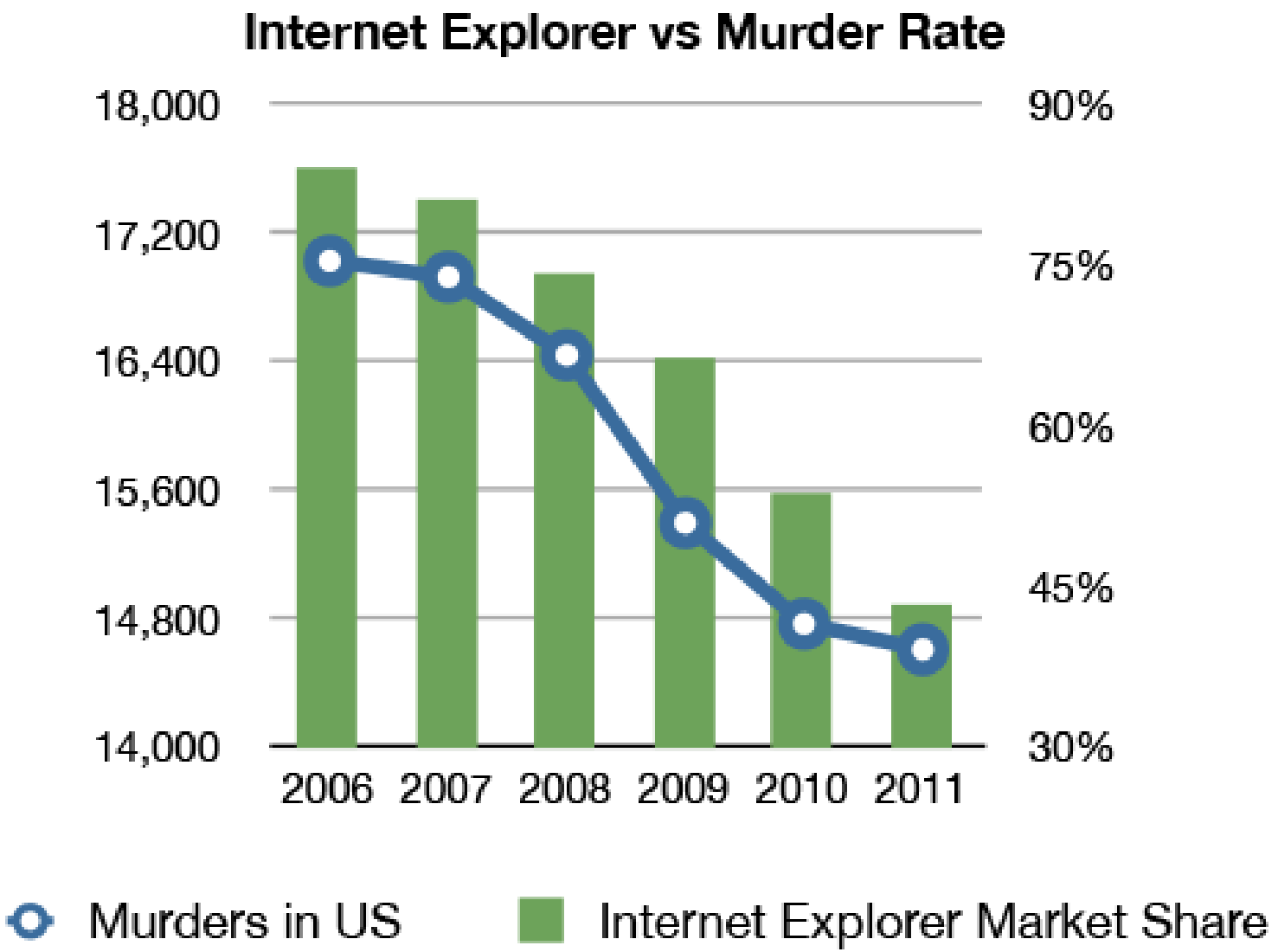
EXCLUSIVE

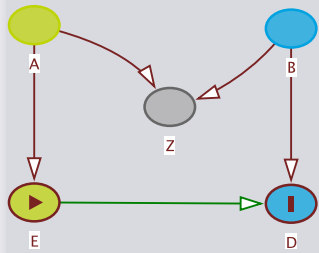
IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show

By CASEY ROSS @caseymross *and* IKE SWETLITZ @ikeswetlitz / JULY 25, 2018

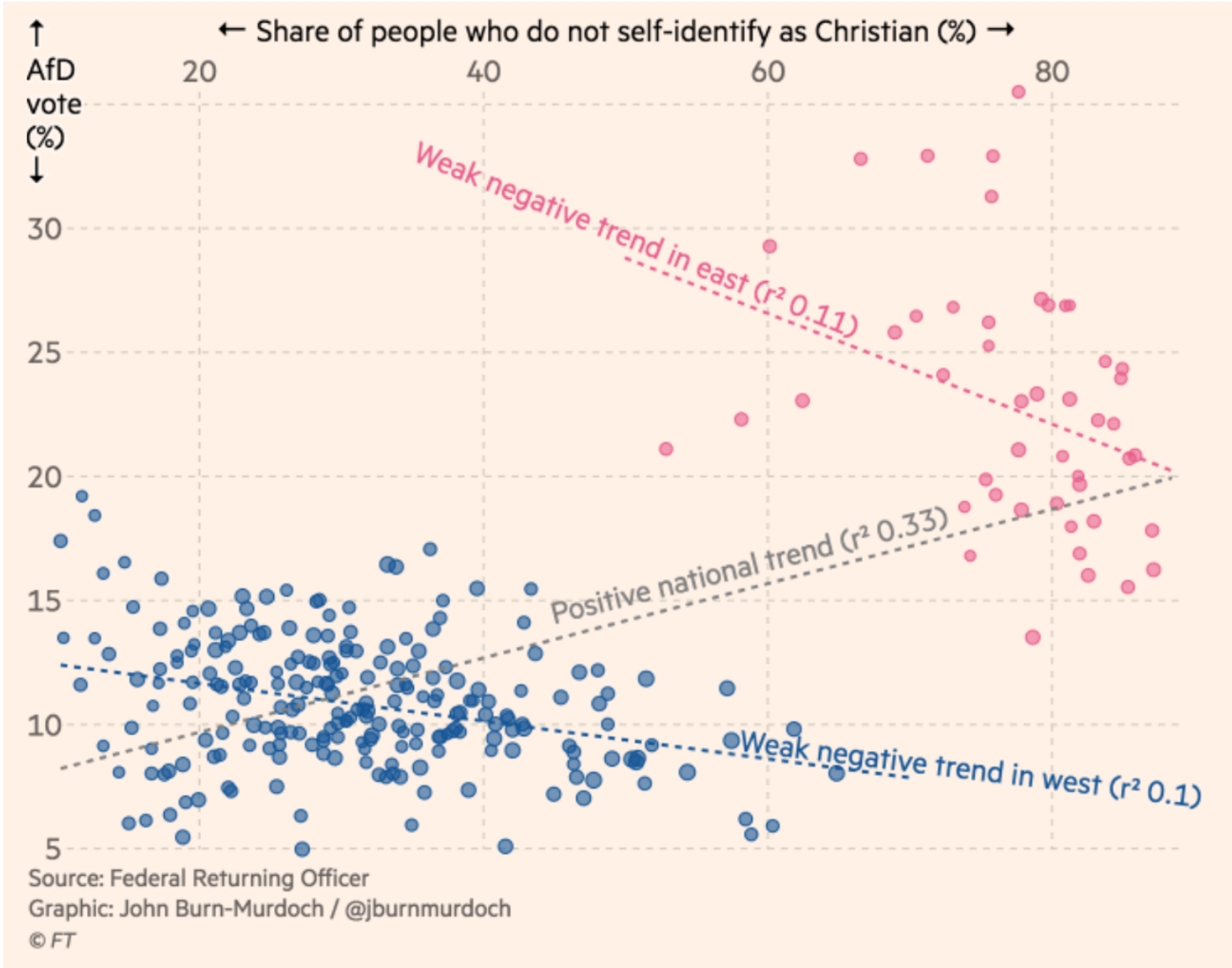


Correlation Does Not Imply Causation

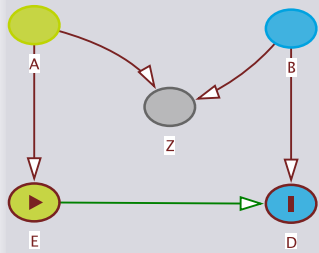




Correlation Patterns Can Be Complex



Johannes Textor

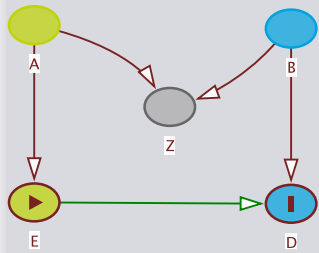


Explanations for Correlations

We observe:

- Across Germany, Christians are more likely to vote right-wing.
- In both east and west, Christians are less likely to vote right-wing.

Our intuition tells us that the nationwide correlation is “driven by” differences between east and west, and is therefore not “real”. But how come we know that?



Simpson's Paradox

Suppose a new treatement for a disease is tested in a trial with the following results:

	Cured	Not Cured
Treated	20	20
Not Treated	16	24

$P(C = 1 \mid T = 1) = 0.5$

$P(C = 1 \mid T = 0) = 0.4$

Now the investigator wants to know whether the treatment is more effective in men or women, and gets the following results:

Males	Cured	Not Cured
Treated	18	12
Not Treated	7	3

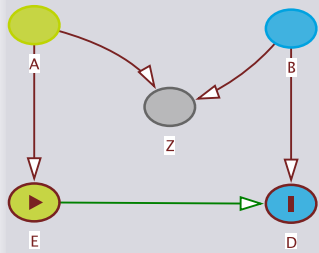
$P(C = 1 \mid T = 1, S = m) = 0.6$
$P(C = 1 \mid T = 0, S = m) = 0.7$

Females	Cured	Not Cured
Treated	2	8
Not Treated	9	21

$P(C = 1 \mid T = 1, S = f) = 0.2$
$P(C = 1 \mid T = 0, S = f) = 0.3$

Do we give the treatment or not?

Judea Pearl:
Simpson's Paradox: An Anatomy
http://bayes.cs.ucla.edu/R264.pdf



Simpson's Paradox

Suppose a new treatement for a disease is tested in a trial with the following results:

	Cured	Not Cured
Treated	20	20
Not Treated	16	24

$P(C = 1 \mid T = 1) = 0.5$

$P(C = 1 \mid T = 0) = 0.4$

The investigator knows that treatment affects blood pressure, and measures it after treatment. She gets the following results:

High BP	Cured	Not Cured
Treated	18	12
Not Treated	7	3

$P(C = 1 \mid T = 1, S = m) = 0.6$

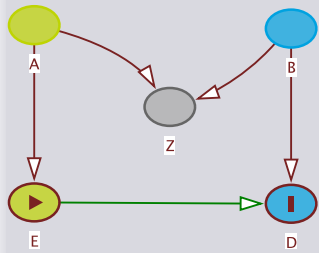
$P(C = 1 \mid T = 0, S = m) = 0.7$

Low BP	Cured	Not Cured
Treated	2	8
Not Treated	9	21

$P(C = 1 \mid T = 1, S = f) = 0.2$

$P(C = 1 \mid T = 0, S = f) = 0.3$

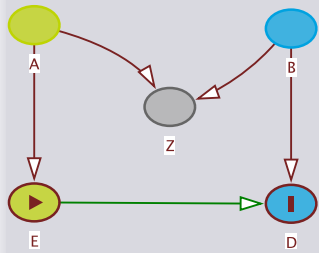
Do we give the treatment or not?



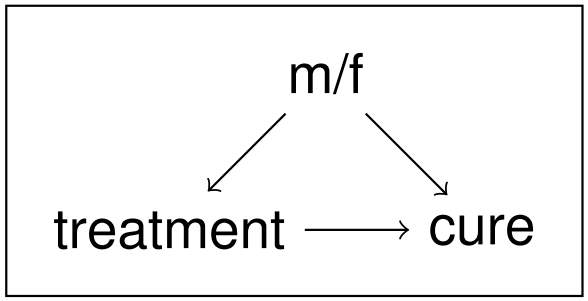
5-Minute Exercise

Exercise

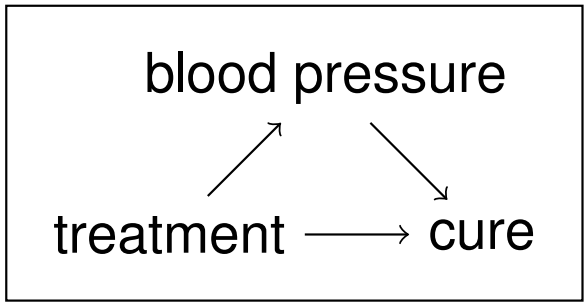
We have just seen two examples of Simpson's paradox. The data were exactly the same, just the labels were different. Does this mean that the answer – do we give the treatment or not – must be the same in both cases?



A Paradox Resolved

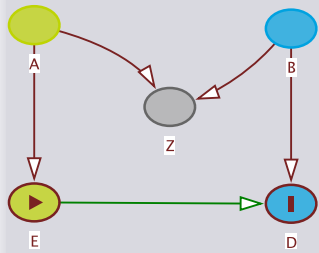


Gender is a confounder of the effect of treatment on cure. Its influence should be removed by conditioning on it. Therefore, given our data, we would not give the treatment.



Blood pressure is a mediator of the relationship between treatment and cure. If we were to condition on it, this might obscure or reverse the effect of treatment on cure. Therefore, given our data, we would give the treatment.

You are smarter than your data. Data do not understand causes and effect; humans do.
– Judea Pearl, "Book of Why"

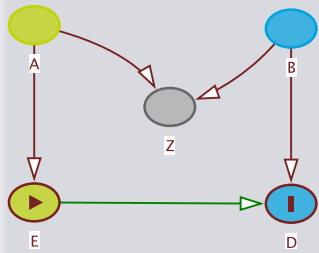


Causal Inference

- Statistics is the art of describing data.
- Causal inference is the art of deriving causal insight from data:
  - Policy recommendations
  - Intervention targets
  - Explanation
- Causal inference can never be unambiguously performed based on data alone. We always require extra information or assumptions.

No causes in – no causes out

(Nancy Cartwright, "Hunting Causes and Using Them")



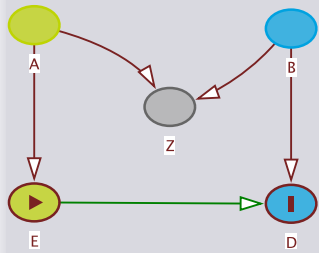
Summary

- Bayesian networks can be used as classifiers, and it is perfectly fine to do that.
- But the real power of Bayesian networks lies in their ability to act as causal models.
- Bayesian networks fuse knowledge about how the world works with observations of the world.

Promises of this course

My goal is to provide you the oppurtunity to learn:

- That data science is more than just fancy algorithms.
- That answering real-world questions often requires substantial domain knowledge.
- How you can combine your knowledge about the world with your data to make useful predictions.



5-Minute Exercise

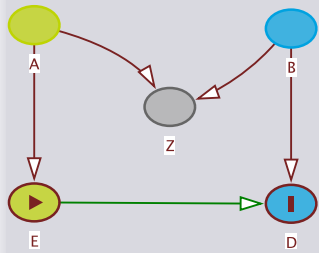
Exercise

What is your assessment of this paragraph?

Nuts and bolts of machine learning

With machine learning, given enough data and a rule-discovery algorithm, a computer has the ability to determine all known physical laws (and potentially those that are currently unknown) without human input. In traditional computational approaches, the computer is little more than a calculator, employing a hard-coded algorithm provided by a human expert. By contrast, machine-learning approaches learn the rules that underlie a dataset by assessing a portion of that data and building a model to make predictions. We consider the basic steps involved in the construction of a model, as illustrated in Fig. 1; this constitutes a blueprint of the generic workflow that is required for the successful application of machine learning in a materials-discovery process.

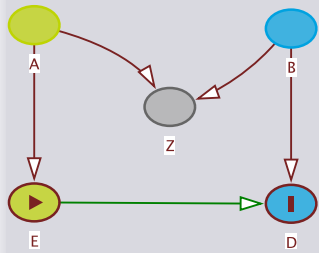
Butler et. al.: Machine learning for molecular and materials science. Nature 559, 547–555 (2018).



Example: Diagnosing Lung Cancer

A patient walks into a doctor's office with breathing problems. The doctor wishes to diagnose whether the patient might have lung cancer. The doctor knows that the following variables are relevant for her diagnosis:

Variable	Domain
Recent visit to asia	{yes, no}
X-ray shows shadow on lung	{yes, no}
Pollution exposure	{weak, medium, strong}
Patient smokes	{yes, no}
Patient has Bronchitis	{yes, no}
Patient has tuberculosis	{yes, no}



Some Notation: Probabilities

We consider (discrete) random variables (e.g. X, Y) that can take on certain values (e.g. $X \in \{x_1, x_2\}$; $Y \in \{y_1, y_2, y_3\}$).

Symbol	Meaning
$P(X = x_1)$	Probability of the event $X = x_1$
$P(x)$	Probability density function of X
$P(x, y)$	Joint probability density of X and Y
$P(x \mid y)$	Conditional probability density of X given Y

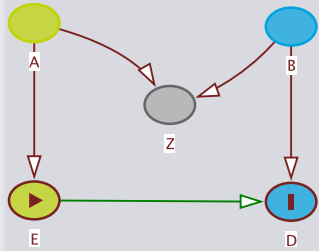
Important identities:

$$\sum_a P(a) = \sum_a P(a \mid b) = 1$$

$$P(a) = \sum_b P(a, b)$$

$$P(a, b) = P(a \mid b)\, P(b)$$

$$P(a, b, c) = P(a \mid b, c)\, P(b \mid c)\, P(c)$$

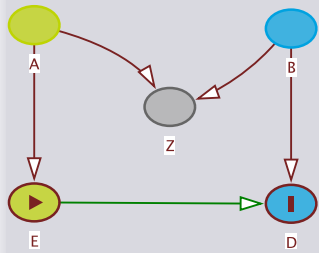


5-Minute Exercise

Exercise

Prove Bayes' theorem:

$$P(h \mid e) = \frac{P(e \mid h)P(h)}{P(e)}$$

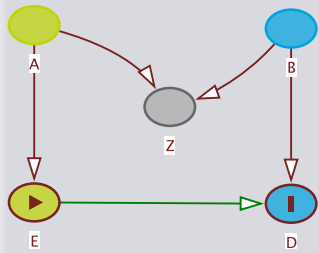


Formalizing the Classification Task

Given the evidence e, choose "best" hypothesis h from the hypothesis space $\mathcal{H}$.
If we read "best" as "most likely", then this means

$$\begin{aligned} & \operatorname{argmax}_{h\in\mathcal{H}} P(h \mid e) \\ \text{(Bayes' rule)} \quad = \quad & \operatorname{argmax}_{h\in\mathcal{H}} \frac{P(e \mid h)P(h)}{P(e)} \\ = \quad & \operatorname{argmax}_{h\in\mathcal{H}} P(e \mid h)P(h) \\ \text{(by def'n)} \quad = \quad & \operatorname{argmax}_{h\in\mathcal{H}} P(e,h)\ . \end{aligned}$$

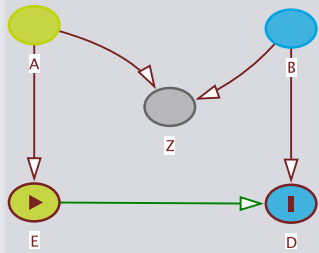
All we need to perform this classification is the joint probability distribution $P(e,h)$.



Joint Probability Distributions

One way to represent a (discrete) joint probability distribution is by using a table.

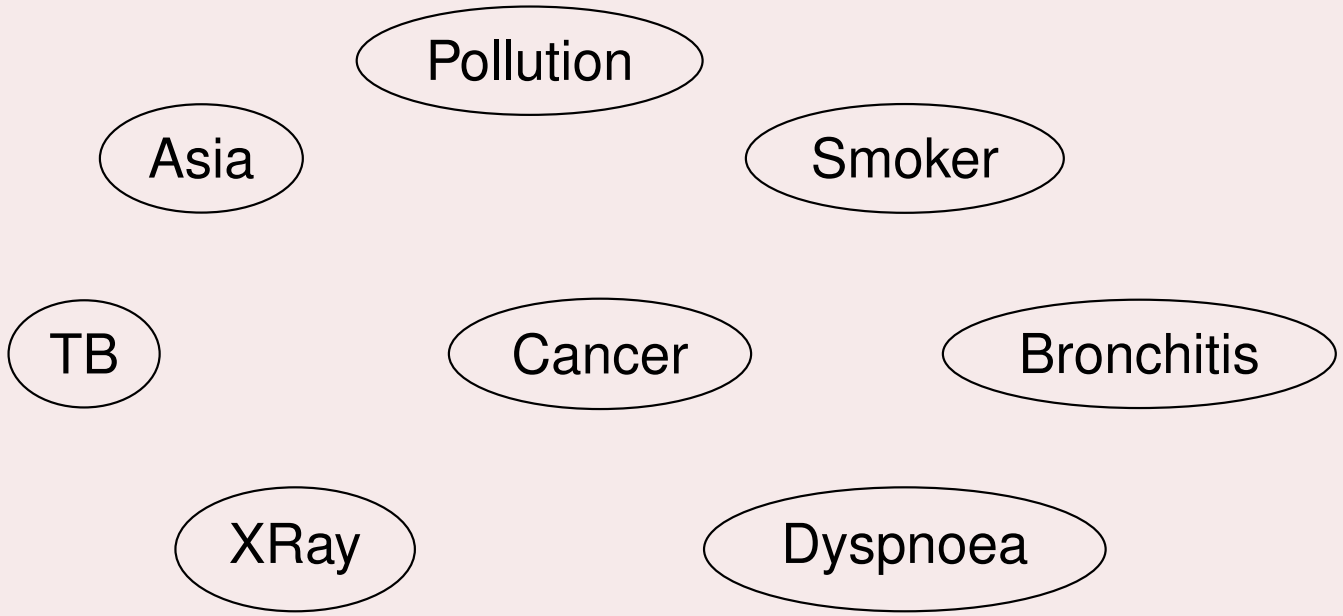
asia	poll'n	smoke	TB	bronch	xRay	dyspn	cancer	P
yes	low	no	no	yes	no	no	no	.01
no	low	no	no	yes	no	no	no	.012
yes	med	no	no	yes	no	no	no	.011
no	med	no	no	yes	no	no	no	.009
yes	high	no	no	yes	no	no	no	.02
no	high	no	no	yes	no	no	no	.015
				⋮				⋮



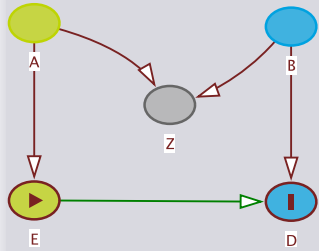
5-Minute Exercise

Exercise

How many parameters (=table rows) are needed to describe the joint probability distribution of the trip to asia example?



Reminder: “pollution” is ternary, others are binary.



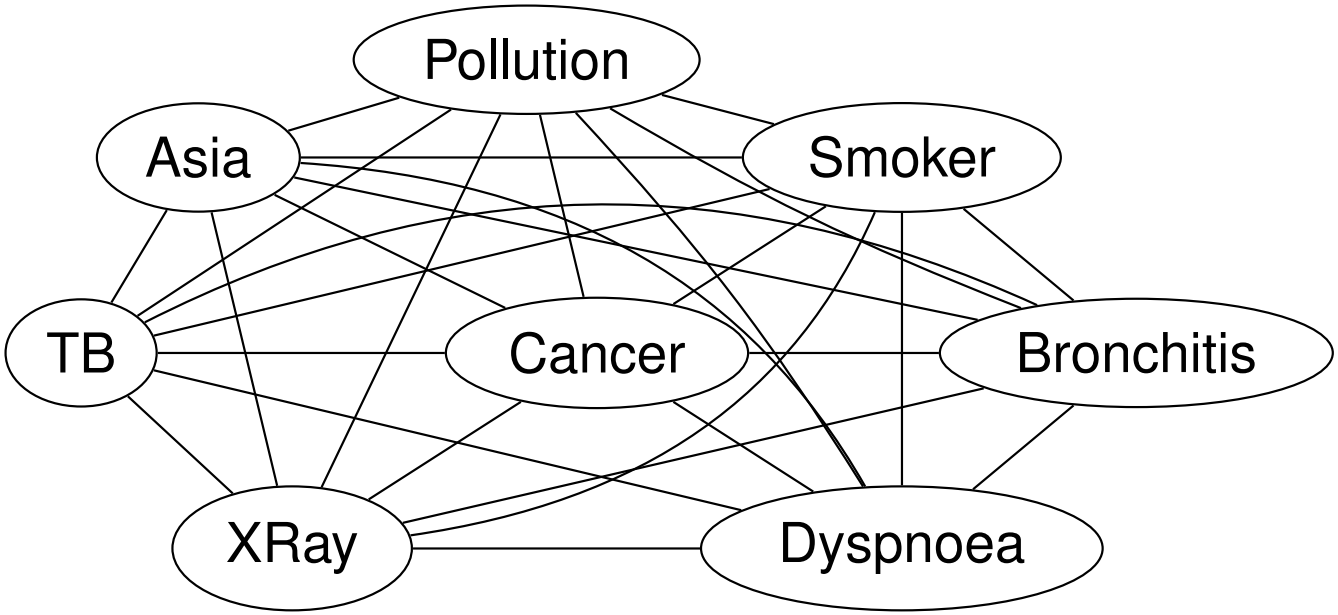
Optimal Bayes

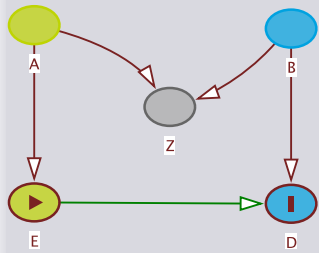
Application of Bayes' theorem leads to the maximum-a-posteriori (MAP)-classifier

$$\operatorname{argmax}_{h \in \mathcal{H}} P(e, h) \ .$$

In some sense (the 0-1-loss function), this is an optimal classifier – it can't be outperformed by any other classifier.

But: the number of parameters of this classifier is exponential in the number of variables – we allow that everything is linked to everything else.





A Simplistic Classifier

By specifying a probability for every combination of evidence data, we take an extreme approach: everything depends on everything.

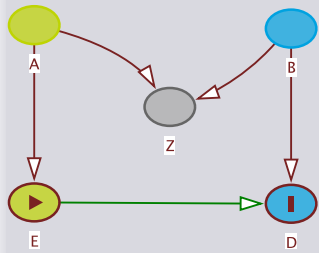
The other extreme approach is to assume that everything is (conditionally) independent. That is, for n evidence variables $E_1, \ldots, E_n$, we assume that

$$P(e_1, \ldots, e_n \mid h) = \prod_i P(e_i \mid h) \, .$$

This allows us to rewrite the MAP classifier as

$$\text{argmax}_{h \in \mathcal{H}} = P(h) \prod_i P(e_i \mid h) \, ,$$

which is called the naive Bayes classifier.



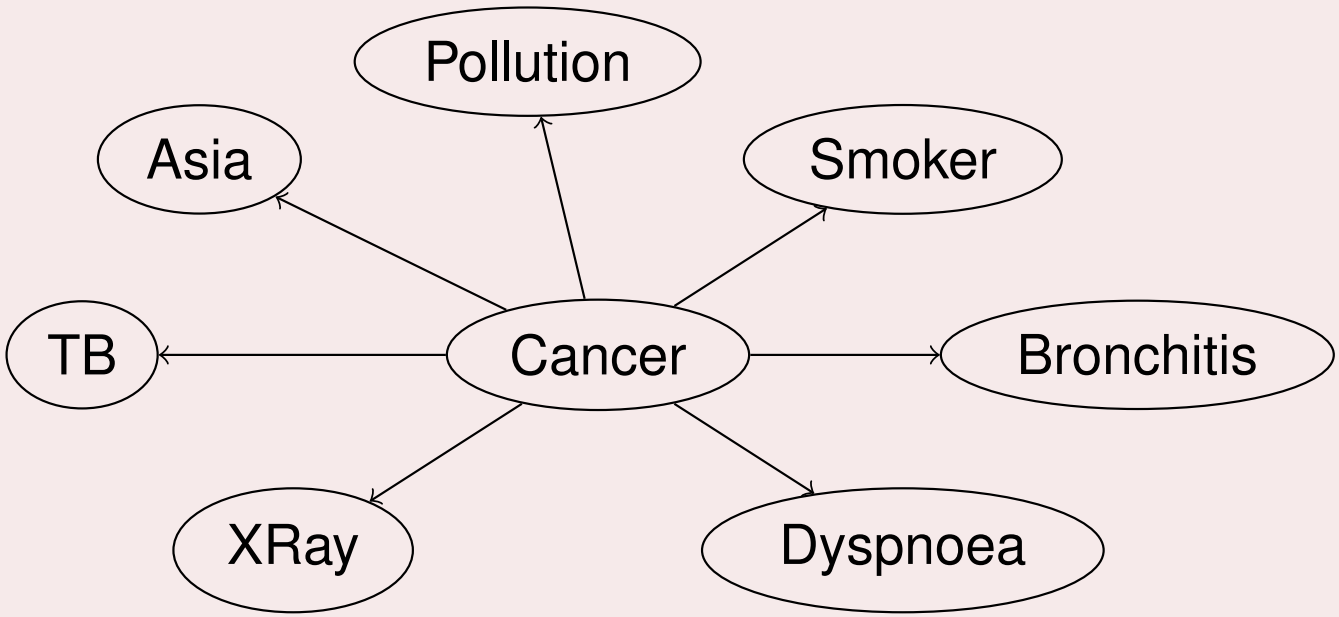
5-Minute Exercise

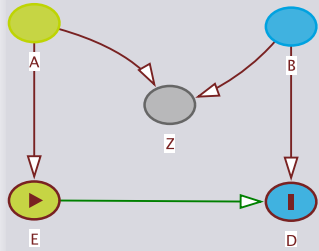
Exercise

How many parameters do we need for the naive Bayes classifier

$$\text{argmax}_{h \in \mathcal{H}} P(h) \prod_i P(e_i \mid h)$$

with our example?





Naive Bayes

In practice, we take the logarithm to arrive at

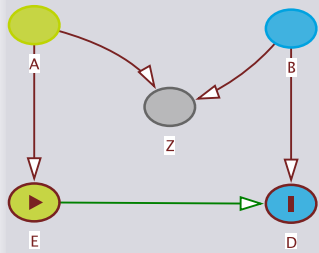
$$\operatorname{argmax}_{h \in \mathcal{H}} \log(P(h)) + \sum_i \log(P(e_i \mid h)) \,,$$

which avoids numerical underflow. The probabilities $P(e_i \mid h)$ can be estimated from training data by simple counting.

Advantages of the naive Bayes classifier

- Easy to train
- Easy to implement
- Easy to evaluate
- Often quite hard to beat!

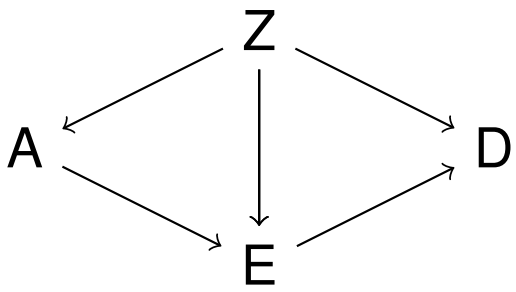
However, the assumption of complete (conditional) independence of all evidence is often too extreme.



Graphs and DAGs

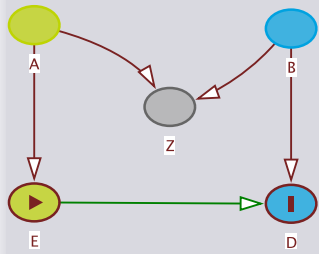
Bayesian networks interpolate between optimal and naive Bayes. They specify the dependencies between the involved variables in terms of a directed acyclic graph (DAG).

- A graph consists of nodes (vertices) and edges (arrows).



- We describe node relations using kinship terminology.
  - Z is a parent of E, D.
  - D is a child of Z, E.
  - D is a descendant of A.
  - A is an ancestor of D.

By convention, each node is a "trivial" ancestor and descendant of itself.
A DAG is a graph in which there is no directed path from a node to itself (cycle; e.g., $X \to Y \to Z \to X$).

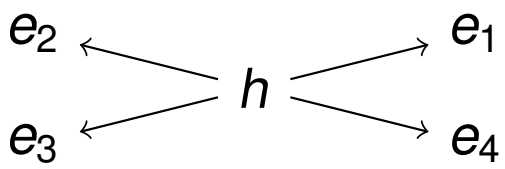


Bayesian Networks

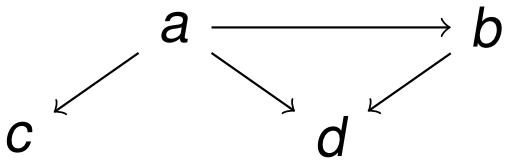
Definition

Given a set of variables $V = X_1, \ldots, X_n$, their probability density $P(x_1, \ldots, x_n)$, and a DAG $G = (V, E)$ (whose nodes are the variables), let $\mathrm{pa}_G(x_i)$ denote the set of all parents of X_i in G. Then P is said to factorize according to G if

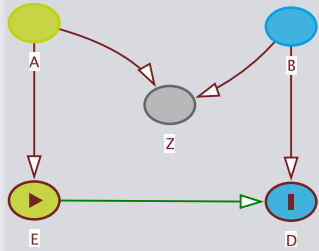
$$P(x_1, \ldots, x_n) = \prod_i P(x_i \mid \mathrm{pa}_G(x_i)) \, .$$



$$P(e_1, e_2, e_3, e_4, h) = P(h)P(e_1 \mid h)P(e_2 \mid h)P(e_3 \mid h)P(e_4 \mid h)$$



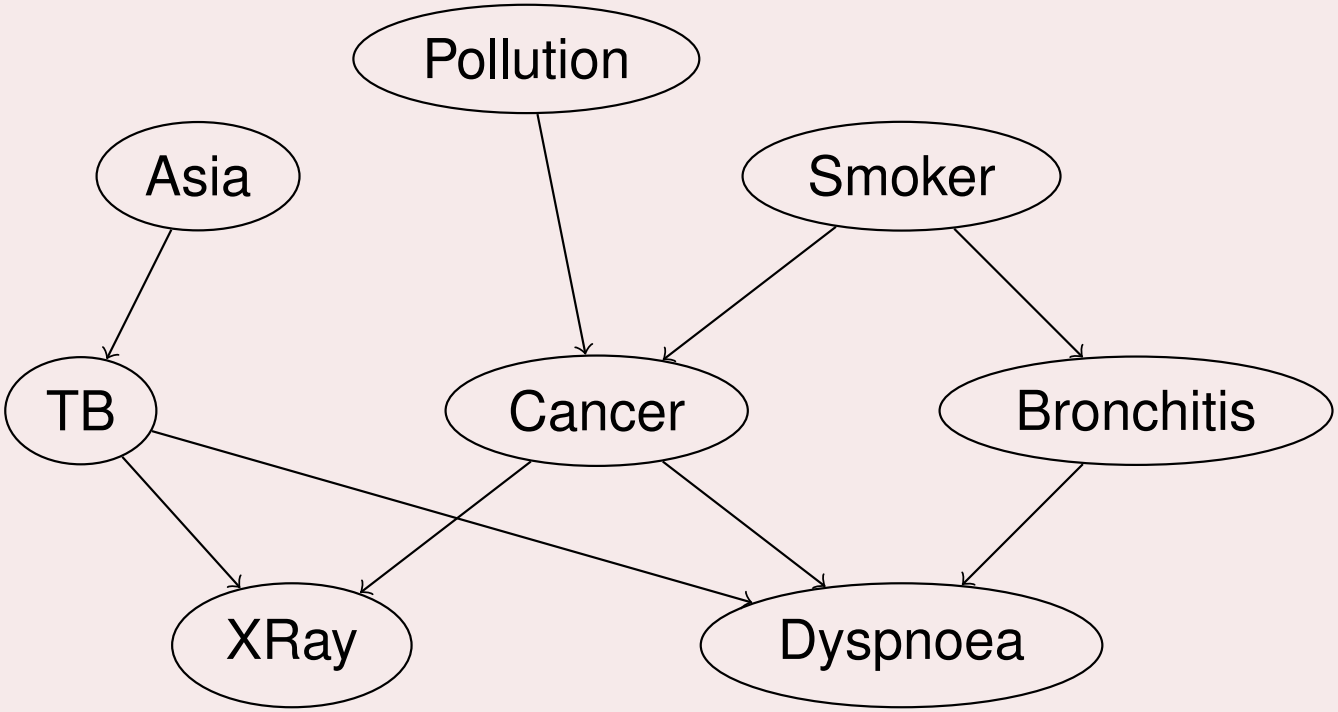
$$P(a, b, c, d) = P(a)P(b \mid a)P(c \mid a)P(d \mid a, b)$$

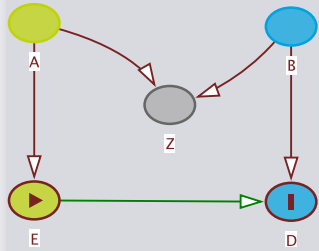


5-Minute Exercise

Exercise

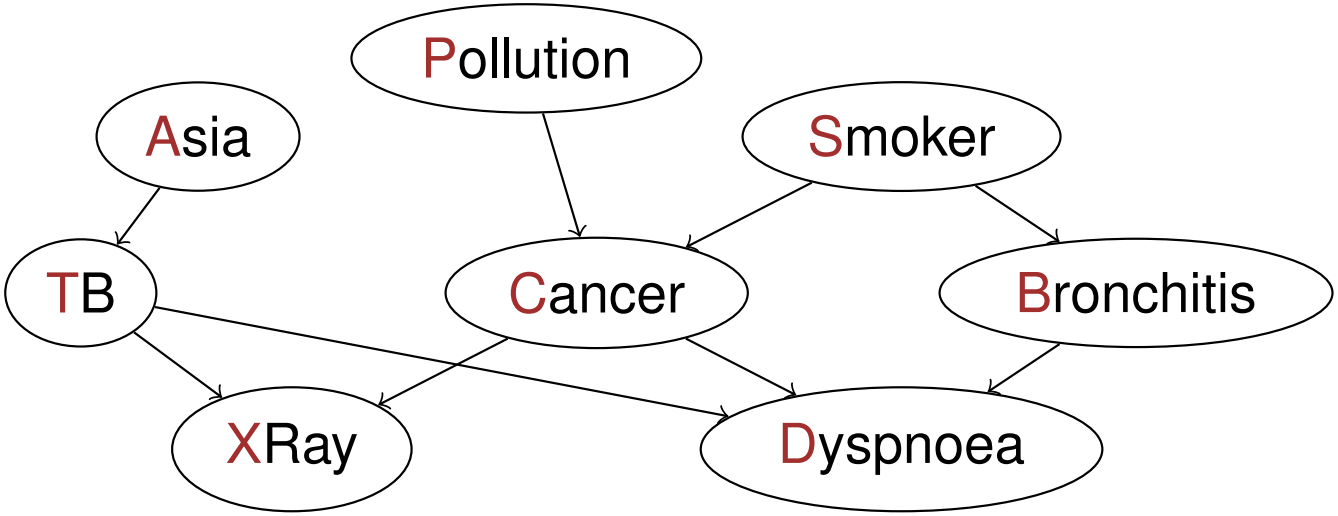
How many parameters do we need to represent this Bayesian network?





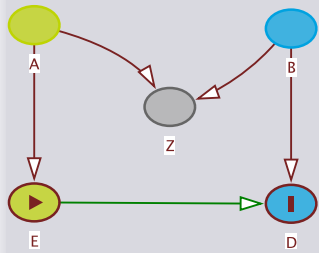
Querying Marginal Probabilities

Having a Bayesian network built, we can now perform inference – we can ask various types of queries. For example, suppose we are interested in the marginal probability $P(\text{Cancer})$.



Which variables will we need to determine $P(\text{Cancer})$? First, recall that

$$P(c) = \sum_{x,d,b,s,p,a,t} P(a)P(p)P(s)P(t \mid a)P(c \mid p,s) \\ P(b \mid s)P(x \mid t,c)P(d \mid t,c,b) \; .$$



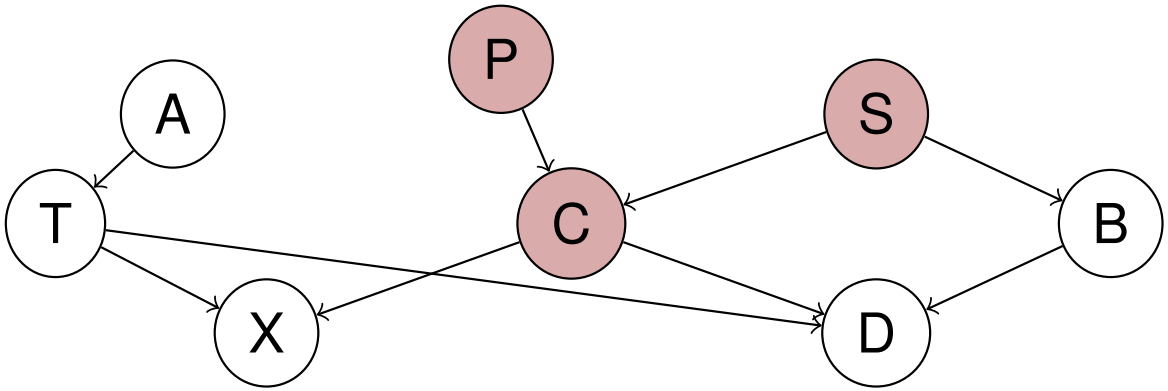
The Ancestral Network

Let's first get rid of the variable X. This turns out to be easy:

$$\sum_X P(a)P(p)P(s)P(t \mid a)P(c \mid p,s)P(b \mid s)P(x \mid t,c)P(d \mid t,c)$$

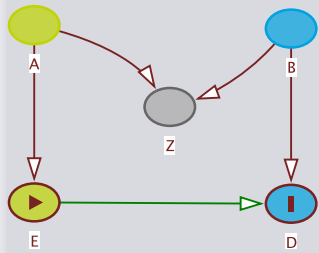
$$= P(a)P(p)P(s)P(t \mid a)P(c \mid p,s)P(b \mid s)P(d \mid t,c) \; .$$

In fact, by going "bottom-up" through the network, we can get rid of all variables that are not ancestors of C. In each step, we'll remove a term where the variable occurs only before the |.



In this way (see lecture notes for details), we get that

$$P(c) = \sum_{s,p} P(p)P(s)P(c \mid p,s) \; .$$



A General Inference Strategy

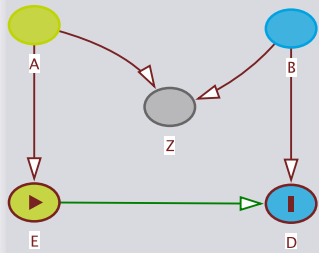
This suggests the following general strategy to answer queries of type $P(a)$ and $P(a \mid b)$:

- Convert conditional to marginal probabilities using the identity

$$P(a \mid b) = \frac{P(a, b)}{P(b)} .$$

- For each marginal probability, determine the ancestral network and write down its factorization.
- Compute the required probabilities from the factorization.

Although this works, it's not exactly an efficient strategy. We'll return to this point later.

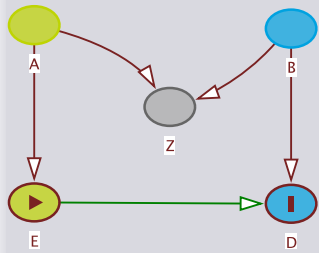


Bayesian Networks and Causality

The formal definition of Bayesian networks obscures the fact that they are intuitively thought of as causal diagrams, in which arrows point from causes to their effect.

This intuition can also itself be formalized – we will return to this topic at the end of this lecture.

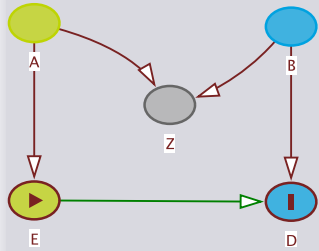
For now, it suffices to say that thinking of what causes what is one way to construct a Bayesian network for a given problem domain.

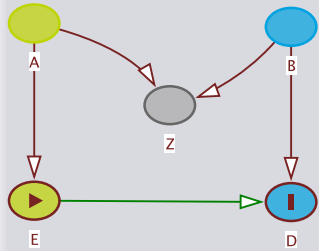


Summary

1. Bayesian networks are graphical models of joint probability distributions.
2. Joint probability distributions can be used for classification and many other inference tasks.
3. There is no better classifier than optimal Bayes.
4. Naive Bayes is worse, but often pretty good.
5. Bayesian networks are the complex middle ground between naive and optimal Bayes.

Kevin B. Korb and Ann E. Nicholson
Bayesian artificial intelligence. Chapters 1 and 2
Chapman & Hall, 2004





Lecture 2

Conditional Independence and *d*-Separation

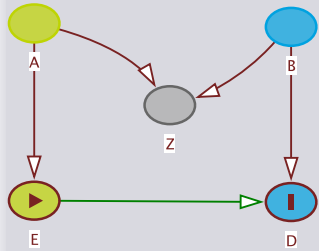
Course Bayesian Networks, September 17th, 2018

Johannes Textor

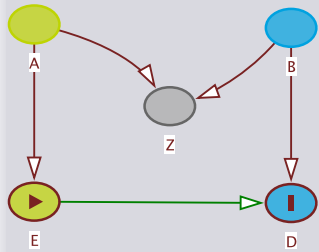
Institute for Computing and Information Sciences

Learning Objectives 2

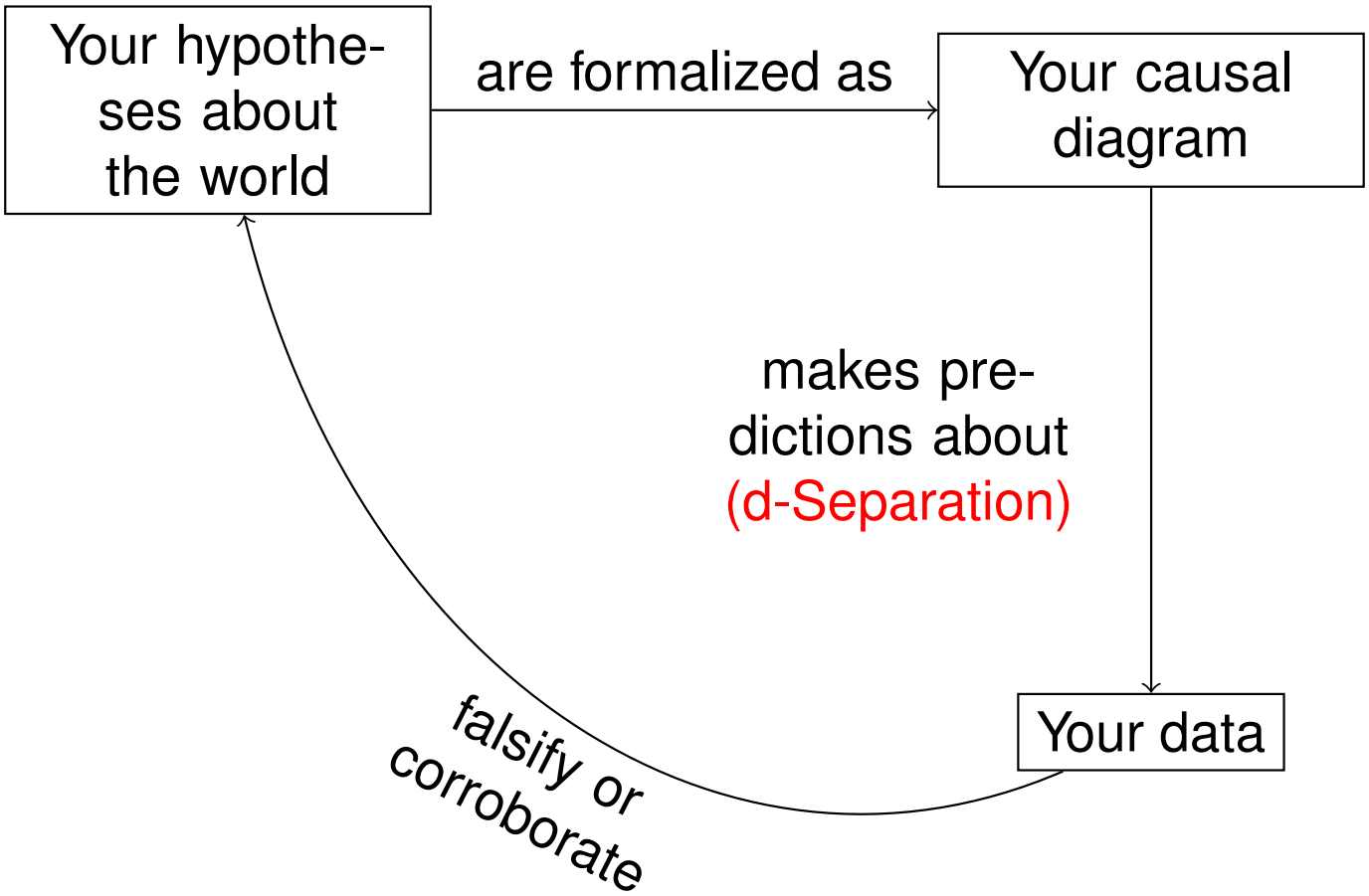
1. Understand the concept of conditional independence.
2. Understand and be able to apply the *d*-separation criterion.
3. Be able to test Bayesian network structure against categorical data.

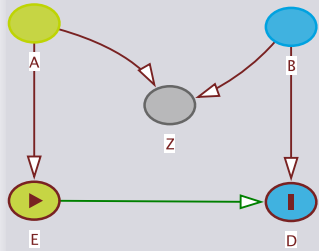


Lecture Outline 2



The Model Building Cycle for Bayesian Networks



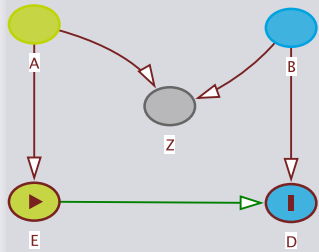


Motivation

- Bayes nets are models of variable relationships in a certain domain.
- Sparse Bayes net models encode certain assumptions about these relationships.
- Incorrect assumptions may lead to incorrect inferences.
- Once a Bayes net is constructed, we can test some of the assumptions it encodes against data.

No free lunch!

Model testing never guarantees a correct model! It can only refute, but never prove it.



Conditional Independence

Two variables X, Y are called independent if

$$P(x, y) = P(x)P(y) .$$

Two variables X, Y are called conditionally independent given a set of variables $\mathbf{Z}$ if

$$P(x, y \mid z) = P(x \mid z)P(y \mid z) .$$

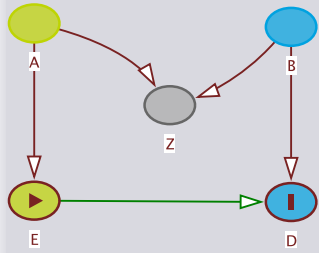
Equivalently, X and Y are called independent given $\mathbf{Z}$ if

$$P(x \mid y, z) = P(x \mid z) .$$

Interpretation: Once we know $\mathbf{Z}$, Y provides us no additional information about X.

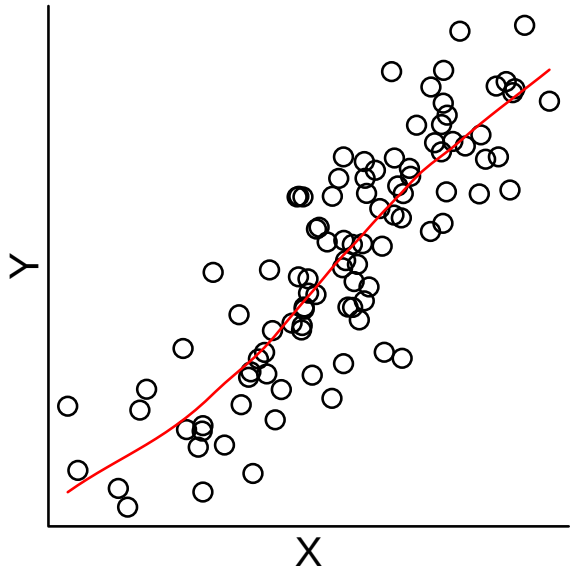
Notation

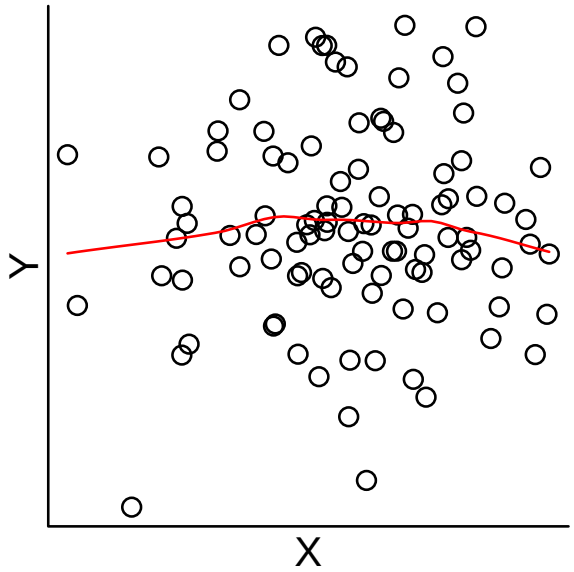
- $X \perp\!\!\!\perp Y$ means: X and Y are independent.
- $X \perp\!\!\!\perp Y \mid Z$ means: X and Y are independent given Z.



Consistency

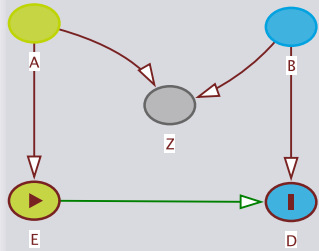
We call a probability density P consistent with G if P factorizes according to G.





$X \longleftarrow Z \longrightarrow Y$ $\qquad$ $X \longrightarrow Z \longleftarrow Y$

It appears that consistency is intimately linked to statistical dependencies.

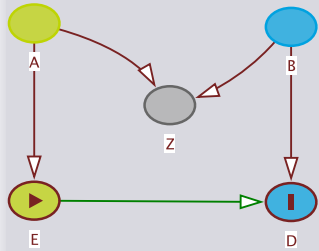


Example 1:

The factorization is:

$$P(x, y) = P(y|x)P(x)$$

This holds for every probability density. Therefore, this model does not imply any restrictions on P.

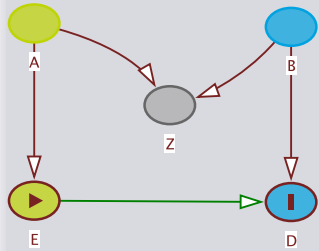


Example 2:

The factorization is:

$$P(x, y) = P(x)P(y)$$

This is the definition of statistical independence of X and Y. Therefore, this model implies that X and Y are independent ($X \perp\!\!\!\perp Y$) in P.



Example: The Fork Model

$$X \longleftarrow M \longrightarrow Y$$

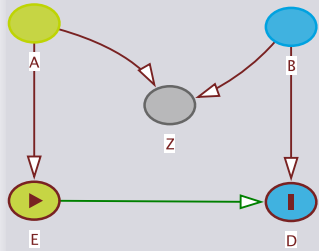
The factorization is:

$$P(x, m, y) = P(m)P(x \mid m)P(y \mid m)$$

Therefore,

$$P(x, y \mid m) = \frac{P(x, y, m)}{P(m)} = P(x \mid m)P(y \mid m)$$

This is the definition of conditional independence of X and Y given M. Therefore, this model implies that $X \perp\!\!\!\perp Y \mid M$.



Example: The Mediation Model

$$X \longrightarrow M \longrightarrow Y$$

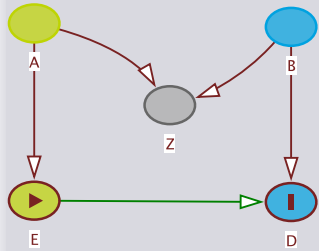
The factorization is:

$$P(x, m, y) = P(x)P(m \mid x)P(y \mid m) = P(x \mid m)P(y \mid m)P(m)$$

Therefore,

$$P(x, y \mid m) = \frac{P(x, y, m)}{P(m)} = P(x \mid m)P(y \mid m)$$

This is the definition of conditional independence of X and Y given M. Therefore, this model implies that $X \perp\!\!\!\perp Y \mid M$.

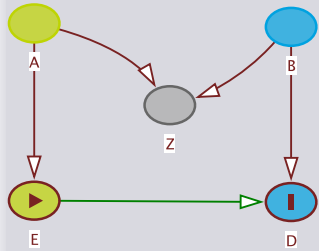


5-Minute Exercise

Exercise

The Bayes net $X \longrightarrow M \longrightarrow Y$ "claims" that X and Y are conditionally independent given $\{M\}$. Fill in the missing probabilities into the table below such that this claim is violated.

M	X	Y	P
0	0	0	
0	0	1	
0	1	0	
0	1	1	1/8
1	0	0	1/8
1	0	1	1/8
1	1	0	1/8
1	1	1	1/8



Example: The Collider

$$X \longrightarrow M \longleftarrow Y$$

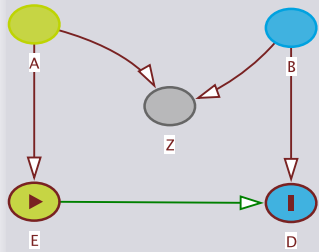
The factorization is:

$$P(x, m, y) = P(x)P(y)P(m \mid x, y)$$

Therefore,

$$\begin{aligned} P(x, y) &= \sum_m P(x, y, m) \\ &= \sum_m P(x)P(y)P(m \mid x, y) \\ &= \sum_m P(x)P(y)P(m \mid x, y) \\ &= P(x)P(y) \end{aligned}$$

This is the definition of statistical independence of X and Y. Therefore, this model implies that X and Y are independent ($X \perp\!\!\!\perp Y$) in P.



d-Separation

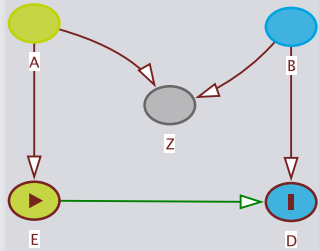
In our examples, we derived a conditional independence by re-arranging the factorization. This always works, but it is tedious.
d-separation is a graphical criterion that allows us to derive all conditional independencies from the Bayes net graph.

Logic of the *d*-separation criterion

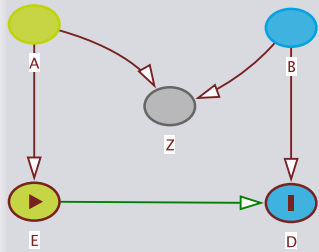
Let G be a DAG and let P be a probability density that factorizes according to G.

- If X and Y are "*d*-separated" by $\mathbf{Z}$, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ is guaranteed to hold.
- If X and Y are not "*d*-separated" by $\mathbf{Z}$, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ may or may not hold.

In other words, *d*-separation is a sufficient, but not a necessary criterion.



Paths in Bayesian Networks



Definition

A path in a Bayes net is a sequence of variables in which each adjacent pair is connected by an edge.

Note that this differs from the classic graph-theoretical concept of a path: we allow to move against arrow directions.
$X \rightarrow M \leftarrow Y$ is a path in the Bayes-net sense, but not in the classic sense.
By convention, we consider only those paths that contain each variable at most once.

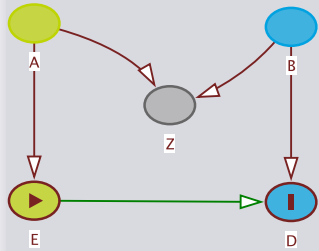
The 3-Variable Case

Let us consider the simplest four Bayes nets with two unconnected variables.

Network	Name	Independence implication	
		Unconditional	Conditional
$X \rightarrow M \rightarrow Y$	chain	none	$X \perp Y \mid \{M\}$
$X \leftarrow M \leftarrow Y$	inverse chain	none	$X \perp Y \mid \{M\}$
$X \leftarrow M \rightarrow Y$	fork	none	$X \perp Y \mid \{M\}$
$X \rightarrow M \leftarrow Y$	collider	$X \perp Y$	none

For a path $\pi = (X, M, Y)$, we define:

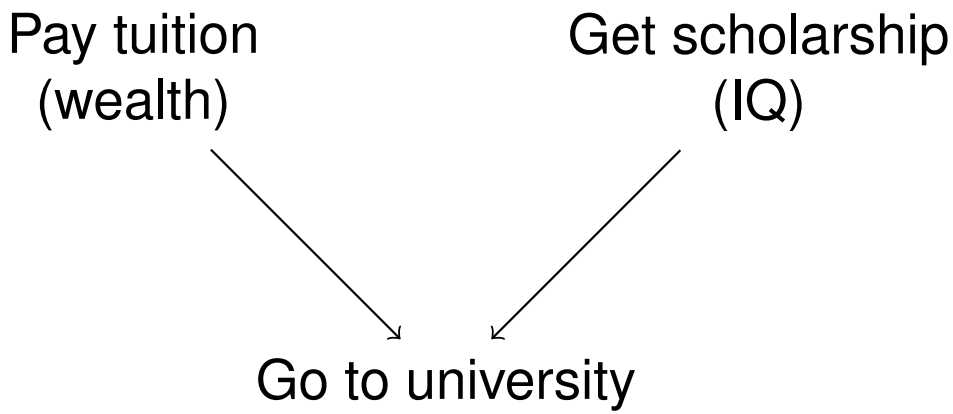
1. If π is a collider $X \rightarrow M \leftarrow Y$:
   - $\{\}$ *d*-separates X and Y
   - $\{M\}$ *d*-connects X and Y
2. If π is not a collider:
   - $\{\}$ *d*-connects X and Y
   - $\{M\}$ *d*-separates X and Y

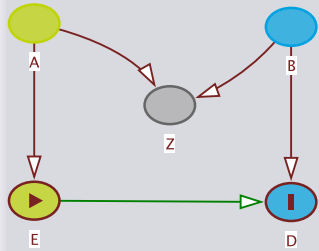


Interpretation of the Collider Case

Example

There is no striking correlation between IQ and wealth in the general population. But surveying students on a campus of a private elite university, one might well find a striking inverse correlation – smarter students tend to be poorer. Why could this happen?



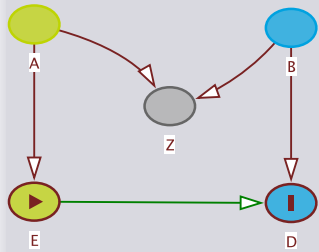


d-Separation for Paths

Consider a Bayes net that consists of a single path $\pi = (X_1, X_2, \ldots, X_n)$, and a set $\mathbf{Z} \subseteq \{X_2, \ldots, X_{n-1}\}$. We say that $\mathbf{Z}$ *d*-separates X_1 and X_n if:

- π contains a collider $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, where $X_i \notin \mathbf{Z}$;
- π contains a non-collider
  - $X_{i-1} \rightarrow X_i \rightarrow X_{i+1}$,
  - $X_{i-1} \leftarrow X_i \rightarrow X_{i+1}$, or
  - $X_{i-1} \leftarrow X_i \leftarrow X_{i+1}$,

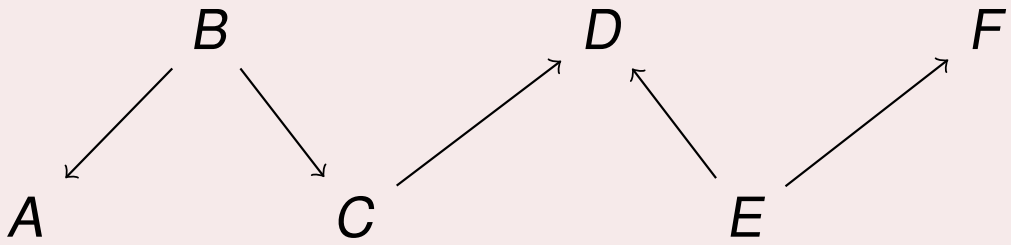
  where $X_i \in \mathbf{Z}$.



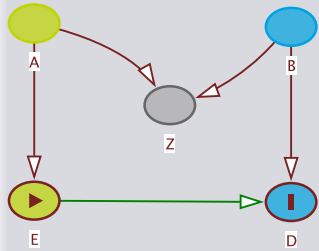
5-Minute Exercise

Exercise

Take the Bayes net below.



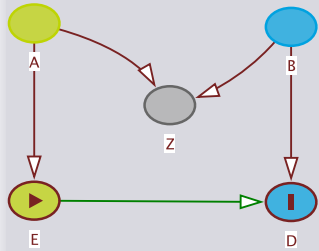
- Give two sets **Z** that *d*-separate *A* and *F*.
- Give a set **Z** containing *D* that *d*-separates *A* and *F*.



The dagitty package

I am going to show how to work with structural causal models in R using the package ‘dagitty‘. The latest version of this package can be downloaded as follows:

```
install.packages("devtools") #if you haven't already
library(devtools)
install_github("jtextor/dagitty/r")
```

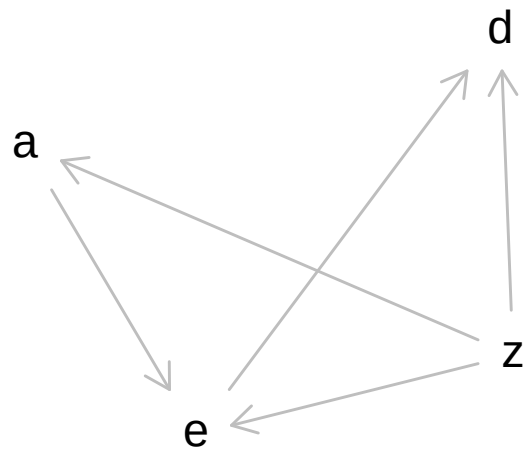



Defining graphs in R

We define graphs using a textual syntax based on the program "graphviz". A simple command exists to plot graphs.

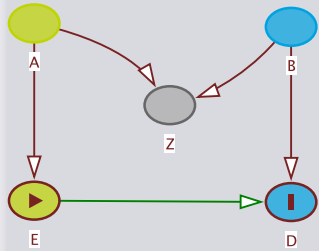
```
g <- dagitty( "dag {
a -> e
e -> d
z -> a
z -> e
z -> d
}" )
```

```
plot(graphLayout(g))
```



Or, more briefly:

```
g <- dagitty( "dag{{a->e->d}<-z}" )
```

Defining graphs in R

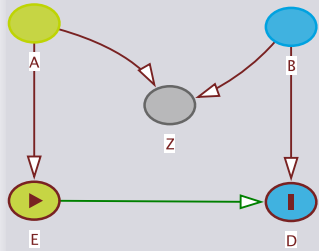
The function `graphLayout` generates automatic layouts. This works OK, but of course you can also specify the layout yourself.

```
g <- dagitty( 'dag {
a [pos="0,0"]
e [pos="1,0"]
d [pos="2,0"]
z [pos="0,1"]
z->{a->e->d}}' )
```

```
g <- dagitty( 'dag{
z->{a->e->d}}' )

coordinates(g) <- list(
X=c(a=0,b=1,c=2,z=1),
Y=c(a=1,b=1,c=1,z=0))
```

Or build it in the dagitty.net GUI, copy/paste the syntax or download it.

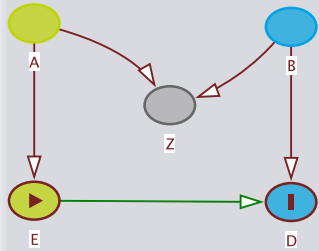
```
g <- downloadGraph("dagitty.net/m331")
```

Path inspection in R

```
library(dagitty)
g <- dagitty("dag{ X -> R -> S -> T <- U <- V -> Y }")
```

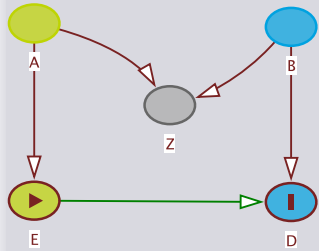

```
paths(g,"X","Y")

## $paths
## [1] "X -> R -> S -> T <- U <- V -> Y"
##
## $open
## [1] FALSE
```

Path inspection in R

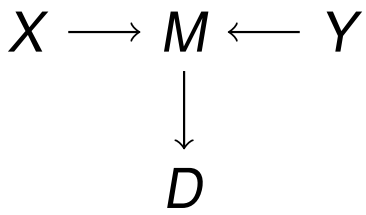
```
g <- dagitty("dag{ X -> R -> S -> T <- U <- V -> Y }")
```


```
paths(g,"X","Y","T")

## $paths
## [1] "X -> R -> S -> T <- U <- V -> Y"
##
## $open
## [1] TRUE
```

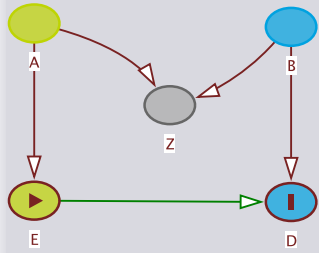



d-Separation for Paths in Graphs

An additional rule is required for colliders that appear in larger graphs. In this Bayes net, we know that conditioning on M can render X and Y dependent.

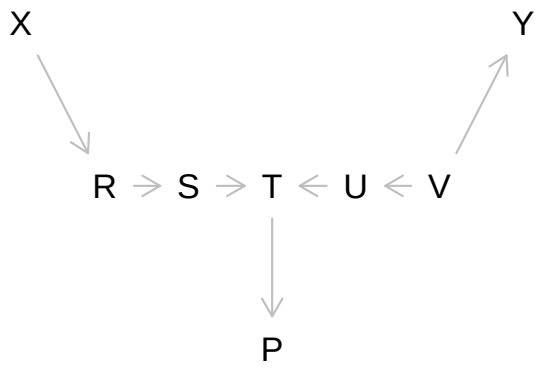


But: conditioning on D can also render X and Y dependent. To see this, consider that D could contain similar information about X and Y as M itself does. Then conditioning on D or M would have quite similar effects.

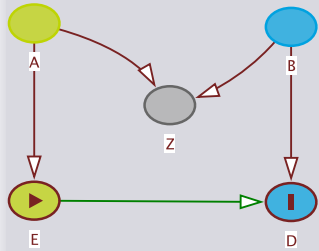


Path inspection in R

```
g <- dagitty("dag{ X -> R -> S -> T <- U <- V -> Y
T -> P }")
```




```
paths(g,"X","Y","P")

## $paths
## [1] "X -> R -> S -> T <- U <- V -> Y"
##
## $open
## [1] TRUE
```

The Full *d*-Separation Criterion

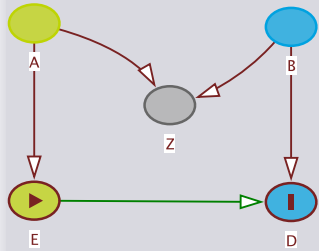
Consider a Bayes net G with variables $\mathbf{V} = \{V_1, \ldots, V_n\}$. We say that $\mathbf{Z} \subseteq \mathbf{V} \setminus \{V_i, V_j\}$ *d*-separates V_i and V_j in G if for every path $\pi = (V_i, V_{k_1}, \ldots, V_{k_n}, V_j), n \geq 0$,

- π contains a collider $X_{i-1} \rightarrow X_i \leftarrow X_{i+1}$, such that X_i is not an ancestor of any node in $\mathbf{Z}$; or
- π contains a non-collider
  - $X_{i-1} \rightarrow X_i \rightarrow X_{i+1}$,
  - $X_{i-1} \leftarrow X_i \rightarrow X_{i+1}$, or
  - $X_{i-1} \leftarrow X_i \leftarrow X_{i+1}$,

  where $X_i \in \mathbf{Z}$.

Theorem (Verma & Pearl, 1984)

If $\mathbf{Z}$ d-separates X and Y in a DAG G, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ in every probability density P that factorizes according to G.



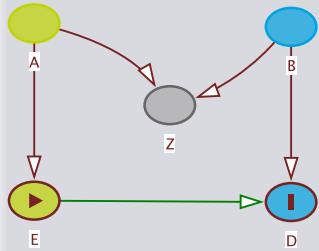
d-Separation For Sets

We can extend the definition of *d*-separation from single variables X and Y to sets $\mathbf{X}$ and $\mathbf{Y}$.

Definition

Consider a Bayes net G with variables $\mathbf{V} = \{V_1, \ldots, V_n\}$. Given three pairwise disjoint sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$, we say that $\mathbf{Z}$ *d*-separates $\mathbf{X}$ and $\mathbf{Y}$ if for all $X \in \mathbf{X}$, $Y \in \mathbf{Y}$, $\mathbf{Z}$ *d*-separates X and Y.

(In short, this works exactly like you would expect.)



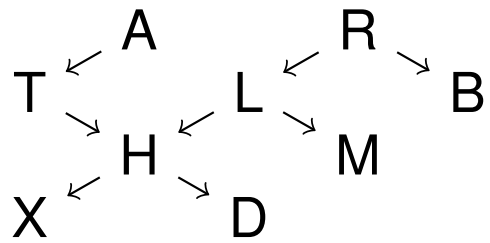
The Ancestral Graph

The *d* in *d*-separation stands for “directed”. But surprisingly, the somewhat intricate *d*-separation criterion can be quite elegantly reduced to standard separation in undirected graphs (à la max-flow-min-cut). We need two definitions for this.

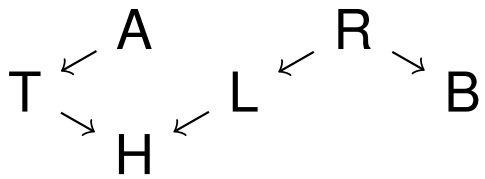
Definition

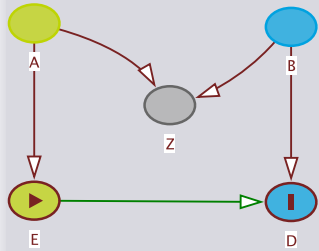
Given a DAG $G = (\mathbf{V}, \mathbf{E})$ and a variable subset $\mathbf{Z} \subseteq \mathbf{V}$, the ancestral graph $G_a(\mathbf{Z})$ is obtained as follows: Delete all variables except those that are ancestors of any variable in $\mathbf{Z}$.

$G =$



$G_a(\{H, B\}) =$





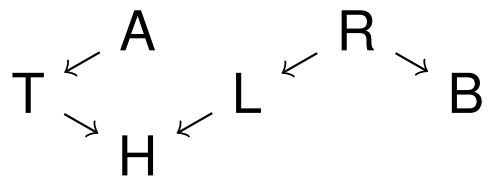
The Moral Graph

Definition

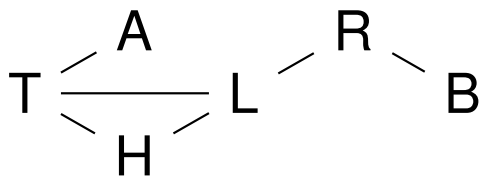
Given a DAG D, the moral graph G^m is the undirected graph created as follows:

1. Connect all variables that have common children with an undirected edge.
2. Replace all directed by undirected edges between the same nodes.

$G =$

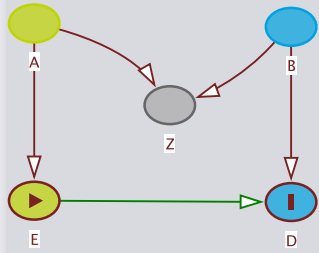


$G^m =$



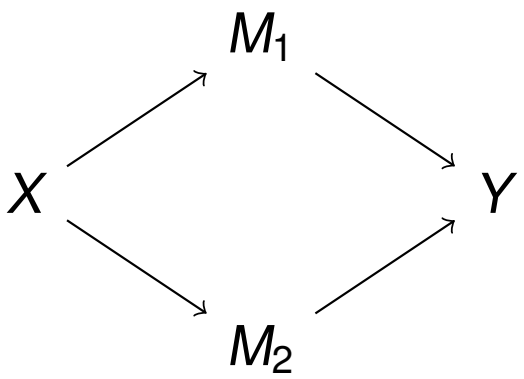
Theorem (Lauritzen & Spiegelhalter)

A set $\mathbf{Z}$ *d-separates* X *and* Y *in the DAG* G *if and only if* $\mathbf{Z}$ *separates* X *and* Y *in the ancestor moral graph* $(G_a(\{X, Y\} \cup \mathbf{Z}))^m$.



Discovering Model Misspecifications

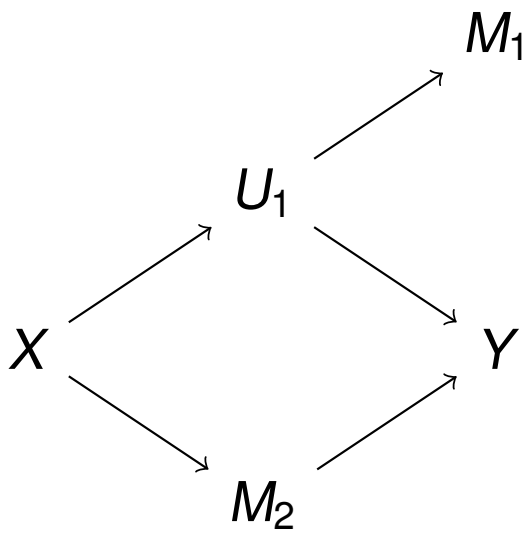
Failed *d*-separation implications inform us about errors in the model structure. In the example below, our Bayes net fails to take into account an unobserved variable U_1. How could we detect such a mistake?



assumed model

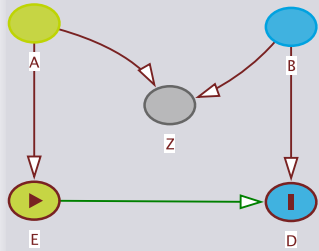
$M_1 \perp\!\!\!\perp M_2 \mid X$

$X \perp\!\!\!\perp Y \mid \{M_1, M_2\}$



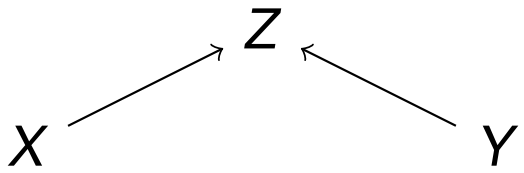
true model

$M_1 \perp\!\!\!\perp M_2 \mid X$



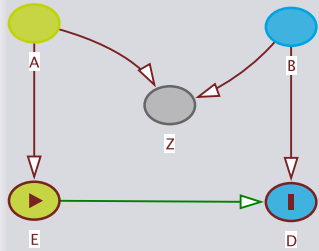
Example 1: Testing a Collider Model

Let us simulate data in R that follows the following collider structure:




```r
set.seed(123)
# Number of samples to be generated
n <- 10000
# X and Y are simple coin tosses
# P(X=1)=P(Y=1)=0.5
X <- rbinom(n,1,p=.5)
Y <- rbinom(n,1,p=.5)
# Z is a variable that depends
# on both X and Y.
# P(Z=1) = e^(X+Y)/(e^(X+Y)+1)
Z <- rbinom(n,1,
  p=exp(X+Y)/(exp(X+Y)+1))
```

```r
# continued
as.data.frame(
  table(X,Y,Z))

##    X Y Z Freq
## 1  0 0 0 1288
## 2  1 0 0  697
## 3  0 1 0  687
## 4  1 1 0  272
## 5  0 0 1 1270
## 6  1 0 1 1843
## 7  0 1 1 1812
## 8  1 1 1 2131
```

Example 1: Testing a Collider Model

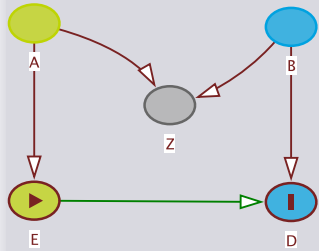
Our collider model $X \rightarrow Z \leftarrow Y$ has a single implication: $X \perp\!\!\!\perp Y$.
We test this implication using a standard chi-square test:

$$\chi^2 = \sum_{x,y} \frac{(O_{x,y} - E_{x,y})^2}{E_{x,y}}$$

where:

$O_{x,y}$ = number of observations where $X = x, Y = y$
$E_{x,y}$ = expected number of observations where $X = x, Y = y$ when X and Y are independent.

When X and Y are independent, χ^2 follows a chi-square distribution with $(n_X - 1)(n_Y - 1)$ degrees of freedom, where n_X is the number of possible values for X.
For binary X and Y, df=1.



Example 1: Performing the Chi-Square Test

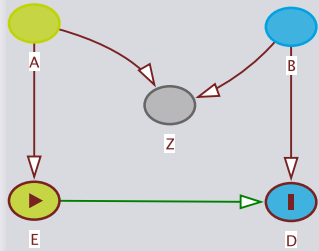
```
# Tabulate observed frequencies
Oi <- table(X,Y)
print( Oi )

##    Y
## X      0    1
##   0 2558 2499
##   1 2540 2403

# Calculate expected frequencies
Ei <- outer(rowSums(Oi),
        colSums(Oi),"*")/n
print(Ei)

##          0        1
## 0 2578.059 2478.941
## 1 2519.941 2423.059
```

```
# Compute the p-value of the
# chi-square test
residuals <- (Oi-Ei)^2 / Ei
chisq <- sum(residuals)
df <- (nrow(residuals)-1)*
        (ncol(residuals)-1)
1-pchisq(chisq, df)

## [1] 0.4222347
```

Or, just do everything in a single line of code:

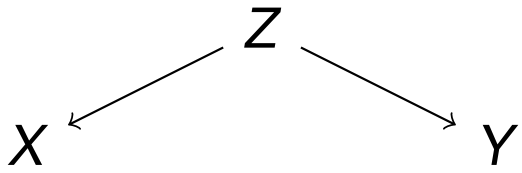
```
chisq.test(X,Y,
   correct=FALSE)$p.value

## [1] 0.4222347
```

The resulting p-value does not provide strong evidence against independence of X and Y. In other words, our data do not provide strong evidence against our model.



Example 2: Testing a Fork Model

Now let us simulate data that follows the fork structure:

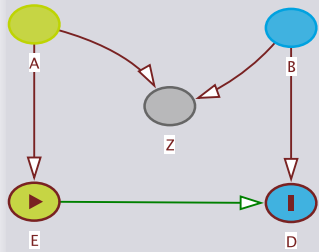



```
set.seed(123)
n <- 10000
Z <- rbinom(n,1,p=.5)
X <- rbinom(n,1,p=exp(Z)/(exp(Z)+1))
Y <- rbinom(n,1,p=exp(Z)/(exp(Z)+1))
```

The fork model does not imply $X \perp\!\!\!\perp Y$, because X and Y are d-connected. So let us see what will happen if we test $X \perp\!\!\!\perp Y$:

```
chisq.test(X,Y)$p.value

## [1] 1.350596e-09
```

This result provides evidence that our data have not been generated by a collider or another model where $X \perp\!\!\!\perp Y$ – which we know is indeed not the case.



Example 2: Testing a Fork Model

Our fork model $X \leftarrow Z \rightarrow Y$ has a conditional independence implication: $X \perp\!\!\!\perp Y \mid Z$.

General strategy to test conditional independence given Z

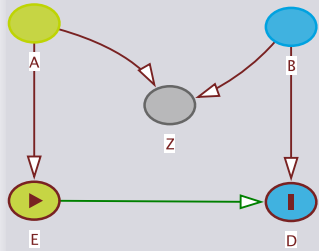
Perform separate independence tests for each value of Z and combine the results.

Let us first perform the separate tests:

```
T.Z0<-chisq.test(X[Z==0],Y[Z==0])
T.Z0$p.value

## [1] 0.9704022
```

```
T.Z1<-chisq.test(X[Z==1],Y[Z==1])
T.Z1$p.value

## [1] 0.8076361
```

Each test on its own does not provide strong evidence against $X \perp\!\!\!\perp Y \mid Z$.



Example 2: A Single Conditional Independence Test

Making decisions based on multiple p-values is not easy. It is more convenient to combine the results for the different levels of Z into a single test, especially when Z has many levels.

For chi-square tests, when $T_0 \sim \chi^2(a)$ and $T_1 \sim \chi^2(b)$ are two chi-square distributed variables with a and b degrees of freedom, then

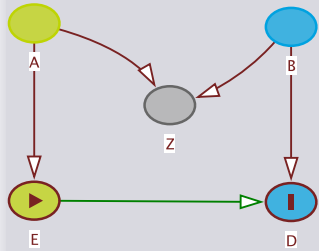
$$T_1 + T_2 \sim \chi^2(a + b)$$

So we can combine the results as follows:

```
chisq.combined <- T.Z0$statistic + T.Z1$statistic
df.combined <- T.Z0$parameter + T.Z1$parameter
1-pchisq(chisq.combined,df.combined)
```

```
## [1] 0.9701265
```

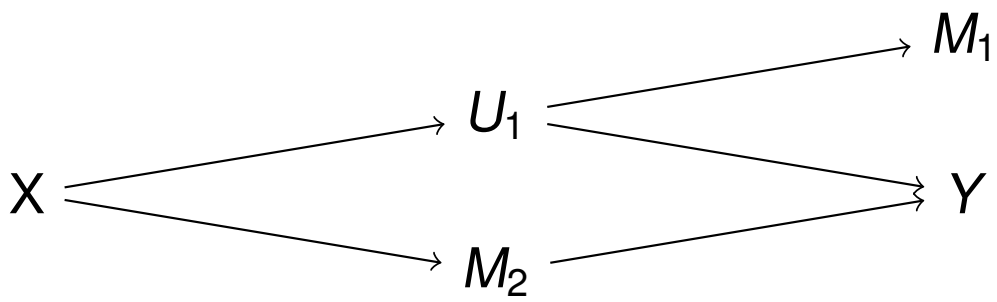
There is no strong evidence agaist $X \perp\!\!\!\perp Y \mid Z$.

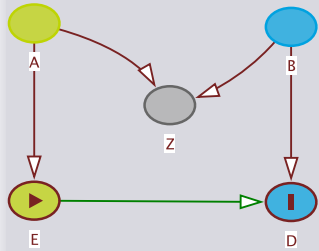


A More Complex Example

Let us generate some binary data that follows a more complex "true" Bayes net structure.

```
set.seed(123)
# A shorthand function that converts log-odds to probabilities.
odds2p <- function(o) exp(o)/(exp(o)+1)
n <- 10000
# X does not depend on anything.
X <- rbinom(n,1,.5)
# X=1 increases the odds of U1=1 and M2=1.
U1 <- rbinom(n,1,odds2p(4*X-2))
M2 <- rbinom(n,1,odds2p(4*X-2))
# U1=1 increases the odds of M1=1.
M1 <- rbinom(n,1,odds2p(2*U1-1))
# U1=1 and M2=1 both increase the odds of Y=1.
Y <- rbinom(n,1,odds2p(2*U1+2*M2))
```



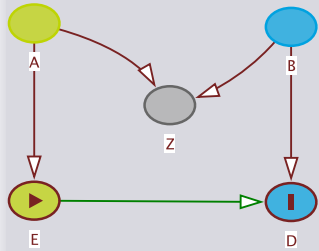


Testing the Assumed Model (I)

Let us start with the first implied independence: $M_1 \perp\!\!\!\perp M_2 \mid X$.

```
chisq <- 0 ; df <- 0
for( x in unique(X) ){
        tst <- chisq.test( M1[X==x], M2[X==x] )
        chisq <- chisq + tst$statistic
        df <- df + tst$parameter
}
1-pchisq(chisq,df)

## X-squared
## 0.4191195
```

There is no strong evidence against dependence here.

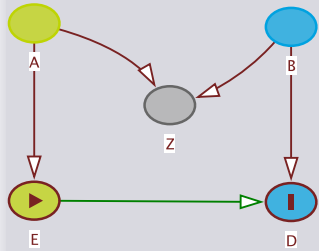


Testing the Assumed Model (II)

Now let us use the same approach to test the second implied independence: $X \perp\!\!\!\perp Y \mid \{M1, M2\}$.

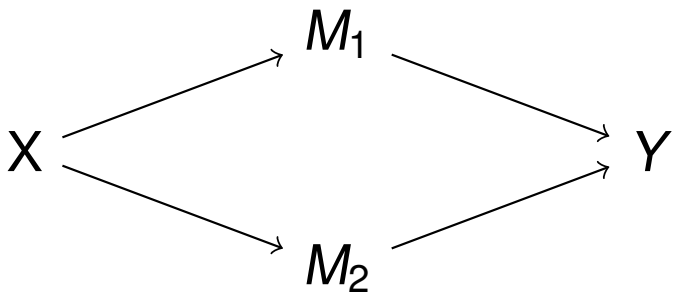
```
chisq <- 0 ; df <- 0
for( m1 in unique(M1) ){
        for( m2 in unique(M2) ){
                tst <- chisq.test( X[M1==m1&M2==m2],
                        Y[M1==m1&M2==m2] )
                chisq <- chisq + tst$statistic
                df <- df + tst$parameter

        }
}
chisq

## X-squared
##  201.3673

1-pchisq(chisq,df)

## X-squared
##         0
```

The data strongly contradict $X \perp\!\!\!\perp Y \mid \{M1, M2\}$.

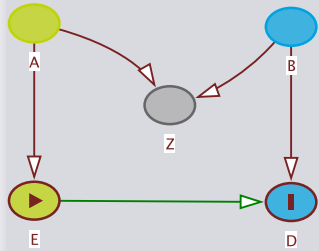


Summary of Test Results

- We assumed the following Bayes net:

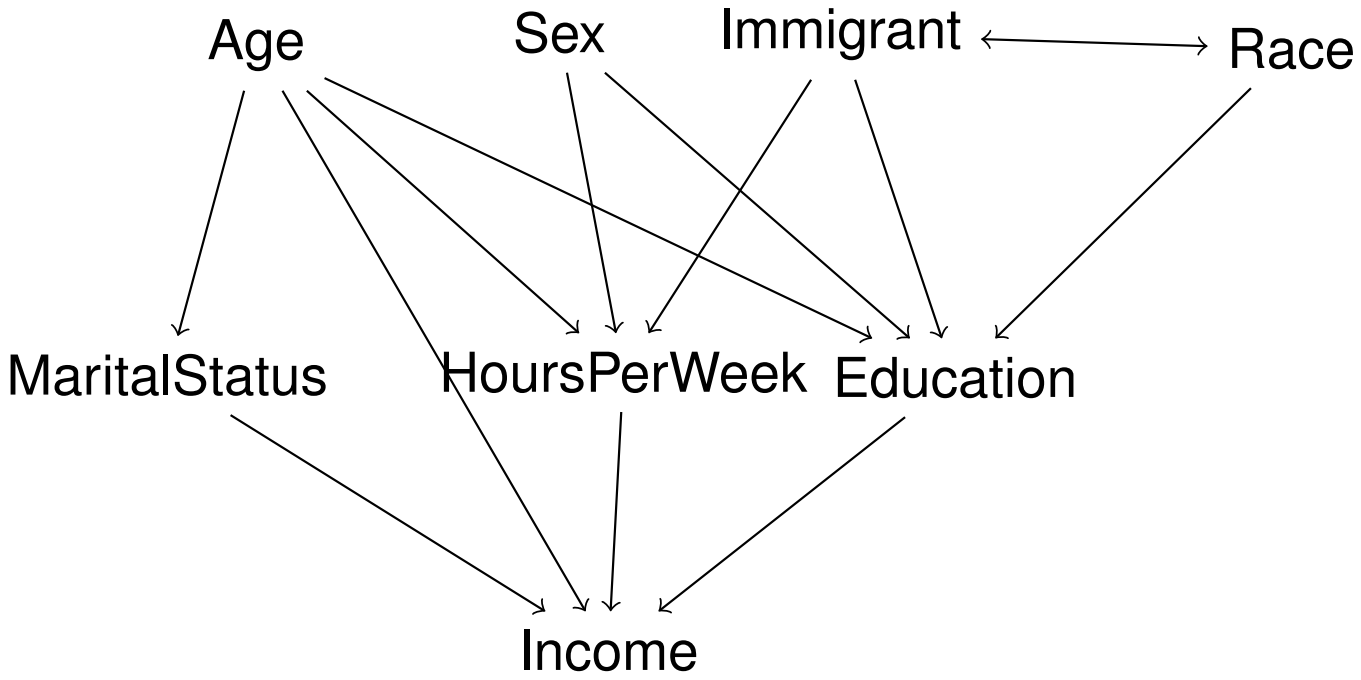


- Using *d*-Separation, we derived two conditional independencies from the net:
  1. $M_1 \perp\!\!\!\perp M_2 \mid X$
  2. $X \perp\!\!\!\perp Y \mid \{M1, M2\}$
- We could not refute the first independence.
- We did refute the second independence.

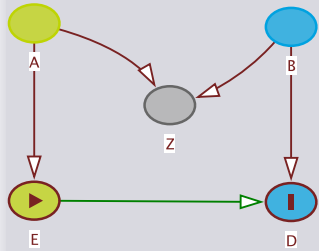


A More Realistic Example

Let us consider a hypothetical causal diagram for a part of the "Adult census income" dataset:



We will test this model on a cleaned version of the data with $\sim 30,000$ records.



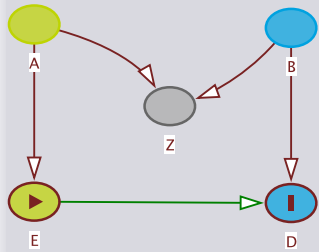
Big Data Brings High Detection Power

With enough data, most Bayesian network models can be falsified.

Example test results

Age ⫫ Immigrant	8.605148e-12
Age ⫫ Race	1.768747e-11
Age ⫫ Sex	1.876589e-61
Education ⫫ MaritalStatus | Age	1.489605e-85
HoursPerWeek ⫫ MaritalStatus | Age	2.025719e-241
HoursPerWeek ⫫ Race | Immigrant	1.957253e-85
Immigrant ⫫ MaritalStatus	9.741660e-11
Immigrant ⫫ Sex	6.016682e-01
MaritalStatus ⫫ Race	1.625036e-62
MaritalStatus ⫫ Sex	0.000000e+00
Race ⫫ Sex	1.681026e-74

All p-values (with one exepction) are extremely low. It seems that everything does depend on everything. But how strong are these dependencies?



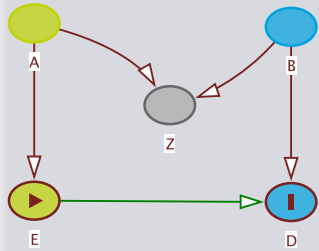
The Root Mean Square Error of Approximation (RMSEA)

Instead of a p-value, which conflates information about dependence strength and sample size, an effect size is often more useful. For chi-square tests, various effect sizes can be defined. An important one is the RMSEA:

$$\text{RMSEA} = \sqrt{\frac{\chi^2/\text{df} - 1}{N - 1}}$$

Properties of the RMSEA

- The expected RMSEA of a "true" model (independence) is 0.
- For a wrong model, the RMSEA converges to a constant positive value as $N \to \infty$ (the p-value converges to 0).
- Higher RMSEA means worse model fit.

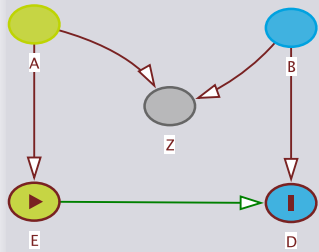


5-Minute Exercise

Exercise

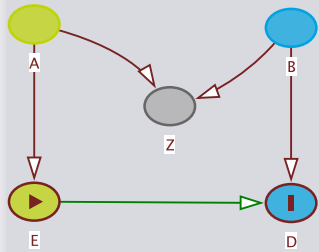
How would you interpret the following RMSEA test results? Which independencies seem particularly strongly contradicted by the data and what do you think this means?

Independence	RMSEA
Age ⫫ Immigrant	0.02110776
Age ⫫ Race	0.02081284
Age ⫫ Sex	0.04865666
Education ⫫ MaritalStatus | Age	0.04919928
HoursPerWeek ⫫ MaritalStatus | Age	0.07491911
HoursPerWeek ⫫ Race | Immigrant	0.04431913
Immigrant ⫫ MaritalStatus	0.02703967
Immigrant ⫫ Sex	0.00000000
MaritalStatus ⫫ Race	0.06843991
MaritalStatus ⫫ Sex	0.31780131
Race ⫫ Sex	0.10499237



Summary

1. Bayesian networks put conditional independence constraints on compatible probability distributions.
2. The *d*-separation criterion allows to read off these constraints from the graphical model structure.
3. The constraints can be tested statistically.

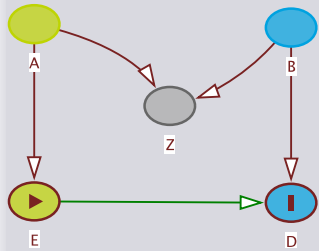


Lecture 3

Structural Equation Models

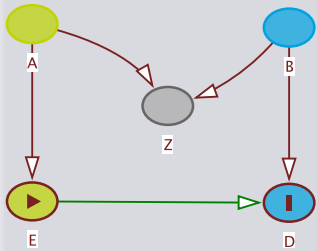
Course Bayesian Networks, October 1st, 2018

Johannes Textor
Institute for Computing and Information Sciences

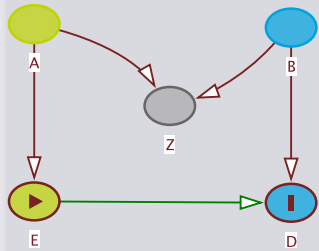


Learning Objectives 3

1. Understand the relationships between Bayesian networks and (a subset of) Structural Equation Models (SEMs).
2. Apply Bayesian network methodology to SEMs, including model testing.



Lecture Outline 3



Continuous Probability Densities

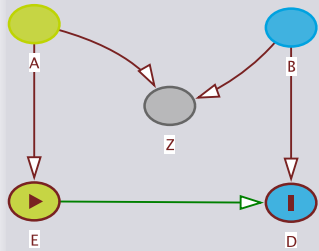
So far, we used discrete random variables and interpreted $P(X = x)$ as "the probability that variable X has the value x".

This interpretation no longer works for continuous variables. Since continuous variables have infinitely many possible values, the probability of a continuous variable to attain any of those values is 0.

Therefore, a different interpretation is needed. In this lecture, we only consider real-valued variables. For that case, $P(x)$ has the following interpretation:

$$\int_{x=x_0}^{x_1} P(x) = \Pr\left[x_0 \leq X \leq x_1\right] \ .$$

(For even more general settings, we need measure theory, which I'm not going to cover here.)



Properties of Continuous Densities

All of our favourite properties of discrete densities hold also for continuous densities, if we replace summing by integration:

$$\int_a P(a) = \int_a P(a \mid b) = 1$$

$$P(a) = \int_b P(a, b)$$

$$P(a, b) = P(a \mid b)\, P(b)$$

$$P(a, b, c) = P(a \mid b, c)\, P(b \mid c)\, P(c)$$

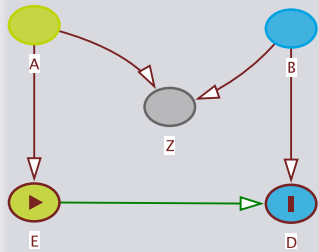
Two variables X, Y are called conditionally independent given a set of variables $\mathbf{Z}$ if

$$P(x, y \mid z) = P(x \mid z) P(y \mid z) \,.$$

Equivalently, X and Y are called independent given $\mathbf{Z}$ if

$$P(x \mid y, z) = P(x \mid z) \,.$$

(Yes, that's the same as for discrete variables.)

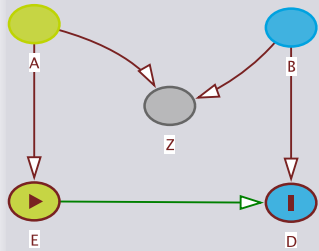


The Uniform Density

The simplest example of a continuous probability density is the uniform distribution defined by

$$P(x) = \begin{cases} \frac{1}{b-a} & a \leq X \leq b \\ 0 & \text{otherwise} \end{cases}$$

This is, however, not a very interesting distribution, since it rarely occurs in nature.



Variance and Covariance

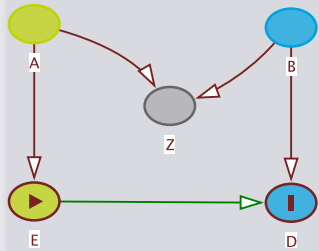
For a random variable **X**, the variance of **X** is defined by

$$\mathrm{Var}(X) = \mathrm{E}(\,(X - E(X))^2\,)\,.$$

For two random variables X and Y, the covariance is defined by

$$\mathrm{Cov}(X, Y) = \mathrm{E}(\,(X - E(X))(Y - E(Y))\,)\,.$$

(Note that $\mathrm{Var}(X) = \mathrm{Cov}(X, X)$.)



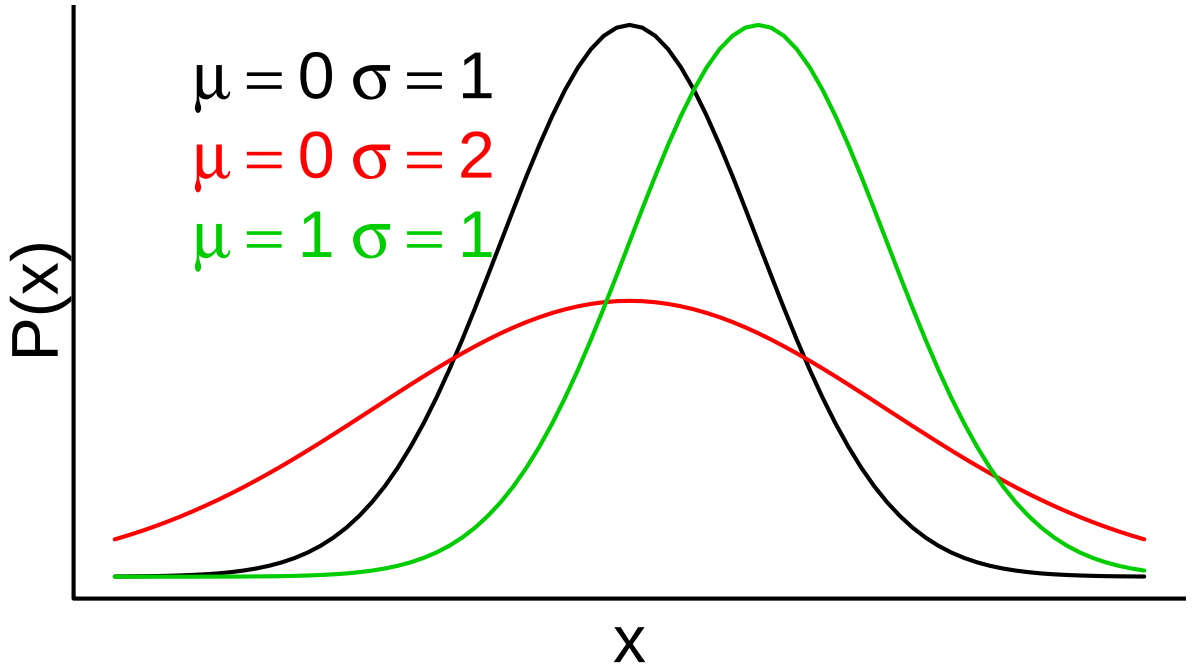
The Univariate Gaussian Distribution

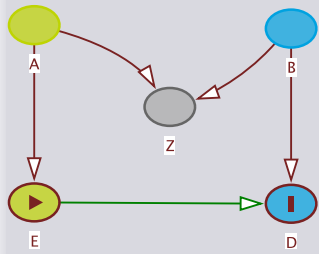
The Gaussian distribution, also called normal distribution, is defined by

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where $\mu = \mathrm{E}(X)$ is the mean and $\sigma^2 = \mathrm{Var}(X)$ is the variance.

If X has a Gaussian distribution, we write that $X \sim \mathcal{N}(\mu, \sigma^2)$.

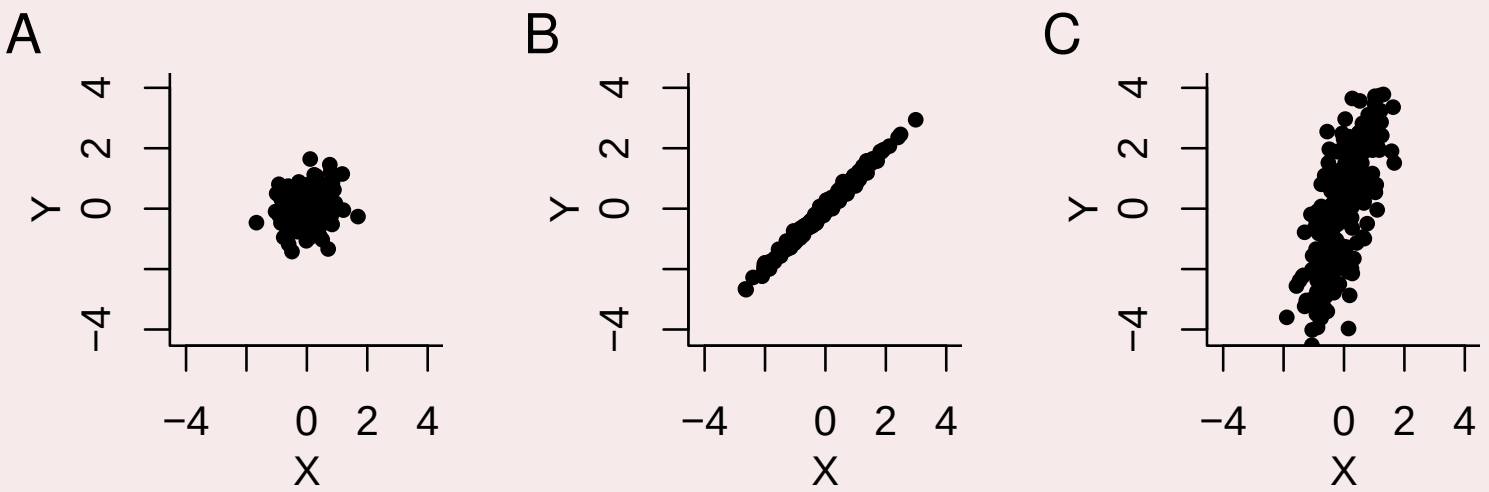




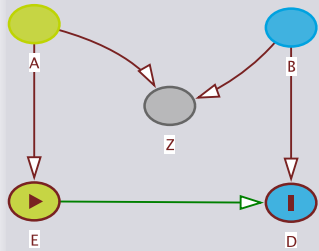
5-Minute Exercise

Exercise

Below, you see three scatterplots of samples drawn from three pairs of random variables. Order these plots with respect to: $\mathrm{Var}(X), \mathrm{Var}(Y), \mathrm{Cov}(X, Y)$.



(For instance, you could write something like "$\mathrm{Var}(X) : A > B > C$".)



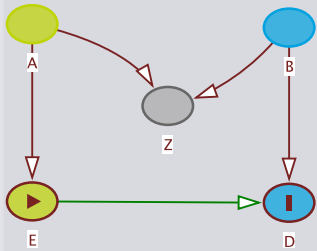
The Multivariate Gaussian Distribution

The multivariate Gaussian distribution for a vector $\mathbf{X}$ of variables is defined by

$$P(\mathbf{x}) = \frac{1}{\sqrt{2\pi|\Sigma|^2}} e^{-(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)}$$

where:

Symbol	Meaning
μ	mean vector
Σ	covariance matrix
$\|\Sigma\|$	determinant of covariance matrix
Σ^{-1}	inverse of covariance matrix



Properties of Gaussian Distributions

Summing and scaling

Let $X \sim \mathcal{N}(\mu, \sigma^2)$. Then for a scalar α,

$$\alpha X \sim \mathcal{N}(\alpha\mu, \alpha^2\sigma^2) \ .$$

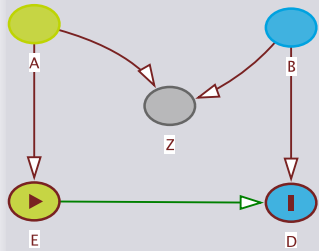
Let $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$. Then

$$X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2 + 2\,\mathrm{Cov}(X, Y)) \ .$$

(The sum of two Gaussians is also a Gaussian.)

Linear combination property

A vector of random variables $(X_1, \ldots, X_n)$ has a multivariate normal distribution if and only if every linear combination of its elements (e.g., $2X_1 + 5X_2$) has a univariate normal distribution.

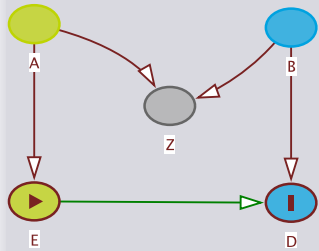


5-Minute Exercise

Exercise

Let $X \sim \mathcal{N}(0,1)$, $Y \sim \mathcal{N}(0,1)$ where $\mathrm{Cov}(X,Y) = 0$.
Please compute:

- $\mathrm{Var}(2X)$
- $\mathrm{Var}(X+X)$
- $\mathrm{Var}(X+Y)$
- $\mathrm{Var}(X+2Y)$



Conditional Covariances

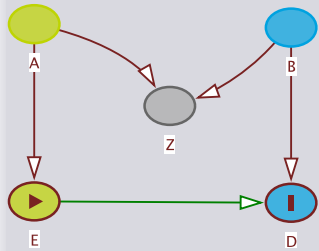
The conditional covariance is the covariance in the conditional probability distribution. It can be computed using the recursive formula

$$\mathrm{Cov}(X, Y \mid \mathbf{Z} \cup \{W\}) = \mathrm{Cov}(X, Y \mid \mathbf{Z}) - \frac{\mathrm{Cov}(X, W \mid \mathbf{Z})\,\mathrm{Cov}(Y, W \mid \mathbf{Z})}{\mathrm{Cov}(W, W \mid \mathbf{Z})}$$

In the case of a single variable,

$$\mathrm{Cov}(X, Y \mid W) = \mathrm{Cov}(X, Y) - \frac{\mathrm{Cov}(X, W)\,\mathrm{Cov}(Y, W)}{\mathrm{Var}(W)}$$

Note that the conditional covariance does not depend on the specific value of Z! This is different from conditional densities, where, in general, $P(x, y \mid Z = z_1) \neq P(x, y \mid Z = z_2)$.



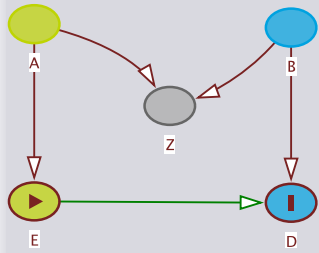
5-Minute Exercise

Exercise

Take the following covariance matrix:

	A	B	X	Y
A	1.0	0.4	0.50	0.70
B	0.4	1.0	0.80	0.70
X	0.5	0.8	1.00	0.65
Y	0.7	0.7	0.65	1.00

Compute the conditional covariance $\mathrm{Cov}(A, B \mid X)$.

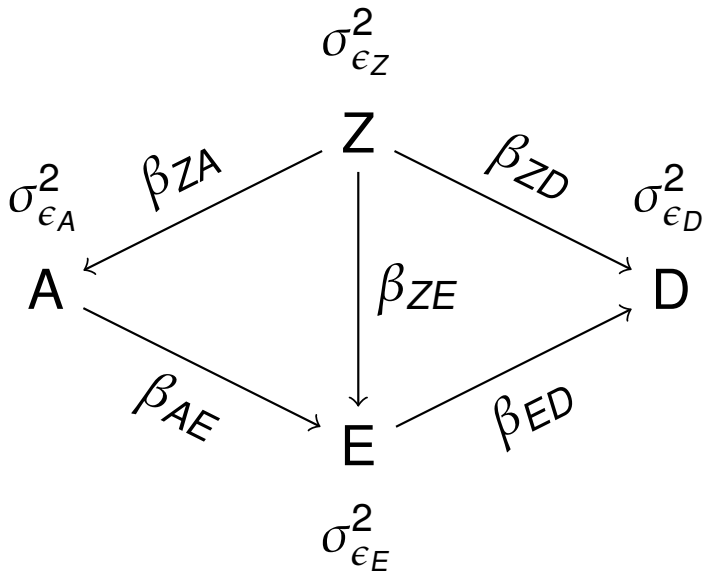


Structural Equation Models

Definition

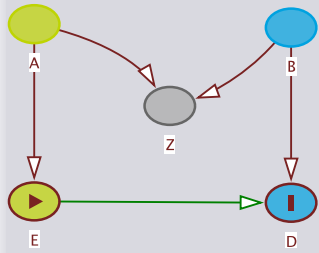
A Structural Equation Model (SEM, SEmodel) is a Bayesian network in which each variable X is a linear function of its parents and a Gaussian variable ϵ_X (called residual).

It is customary to represent a SEM as a path diagram labelled with the linear coefficients and residual variances.

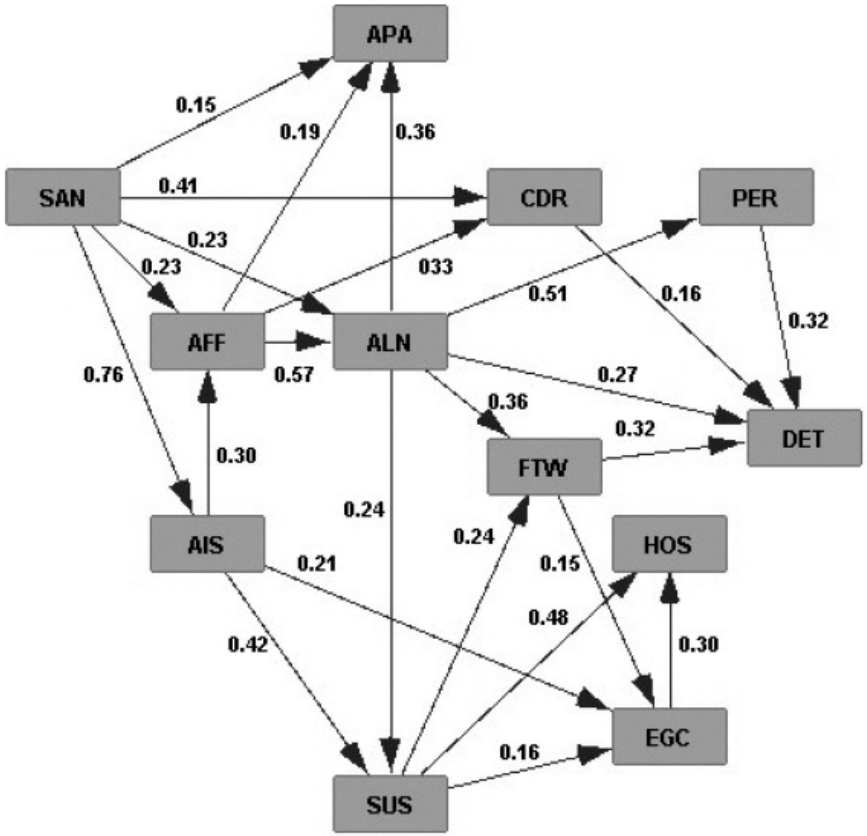


$$Z := \epsilon_Z$$
$$E := \beta_{AE} A + \beta_{ZE} Z + \epsilon_E$$
$$A := \beta_{ZA} Z + \epsilon_A$$
$$D := \beta_{ED} E + \beta_{ZD} Z + \epsilon_D$$

Residual means can also be represented, but we will assume here that all means are 0. (Real data is often "centered" to mean 0 anyway, as a preprocessing step.)

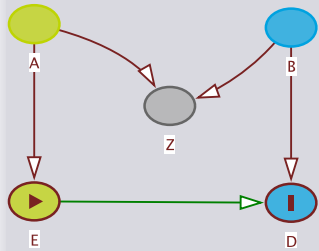


Example: Schizophrenic Disease Unfolding



SAN: Social Anxiety;
AIS: Active Isolation
AFF: Affective Flattening
SUS: Suspiciousness
EGC: Egocentrism
FTW: Living in a Fantasy World
ALN: Alienation
APA: Apathy
HOS: Hostility
CDR: Cognitive Derailment
PER: Perceptual Aberrations
DET: Delusional Thinking

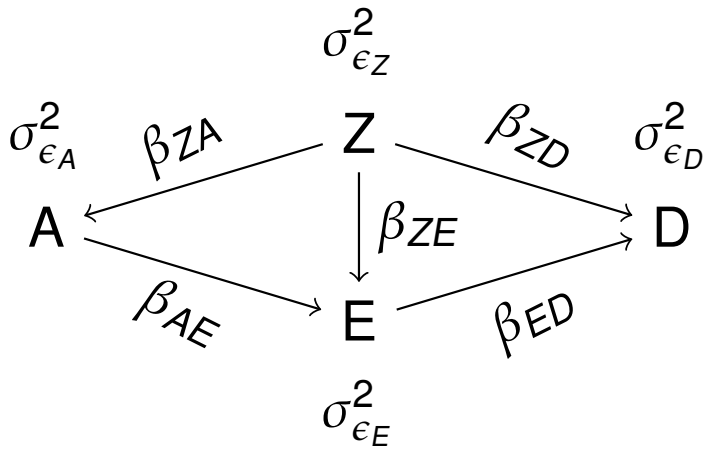
van Kampen, European Psychiatry 29(7):437–48, 2014



Multivariate Normality of SEMs

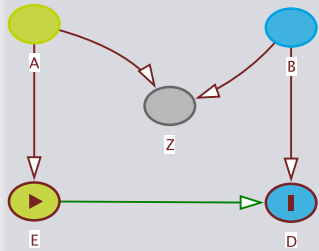
SEMs generate multivariate normal distributions.

- Each variable is a linear combination of residuals.



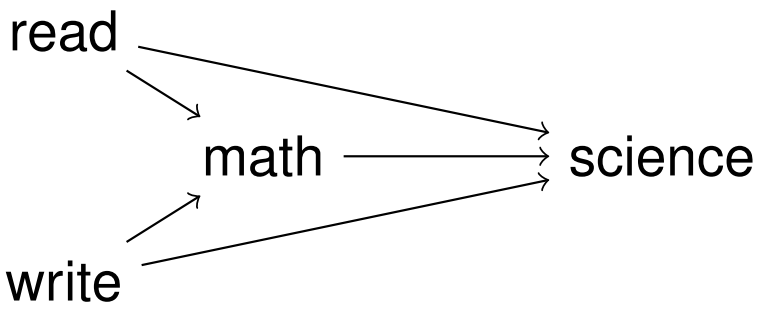
$$
\begin{aligned}
D &= \beta_{ED}E + \beta_{ZD}Z + \epsilon_D \\
&= \beta_{ED}(\beta_{AE}A + \beta_{ZE}Z + \epsilon_E) + \beta_{ZD}\epsilon_Z + \epsilon_D \\
&= \beta_{ED}(\beta_{AE}(\beta_{ZA}Z + \epsilon_A) + \beta_{ZE}Z + \epsilon_E) + \beta_{ZD}\epsilon_Z + \epsilon_D \\
&= \beta_{ED}(\beta_{AE}(\beta_{ZA}\epsilon_Z + \epsilon_A) + \beta_{ZE}\epsilon_Z + \epsilon_E) + \beta_{ZD}\epsilon_Z + \epsilon_D
\end{aligned}
$$

- Each linear combination of variables is therefore also a linear combination of residuals.
- Residuals are normally distributed.

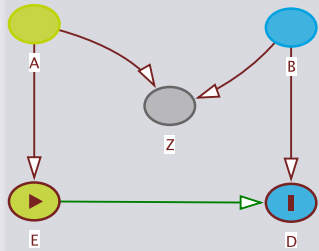


Fitting SEMs locally

How do we estimate the SEM from data? One option, as for any Bayesian network, is to estimate each node separately. This can be done by running a linear regression of each node on its parents (just like for the logistic regression in Chapter 3).



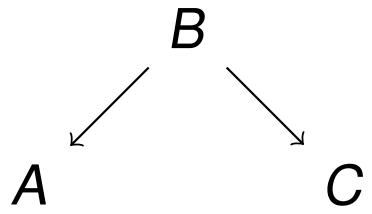

```
hsb2 <- read.table(header=T, sep=",",
'https://stats.idre.ucla.edu/wp-content/uploads/2016/02/hsb2-2.csv')
mdl <- lm( science ~ read + write + math,
                data=data.frame( scale( hsb2 ) ) )
coef(mdl) # extract path coefficients

##   (Intercept)          read         write          math
## -2.027376e-16  3.122533e-01  1.977167e-01  3.018540e-01

summary( mdl )$sigma # extract residual standard deviation

## [1] 0.7125376
```

Applying *d*-Separation to SEMs

Like any Bayesian network, SEMs can be tested by deriving conditional independence relationships. In multivariate Gaussian distributions, conditional independence entails a vanishing conditional covariance (=conditional covariance is 0).

Model:



Implication:

$A \perp\!\!\!\perp C \mid B$

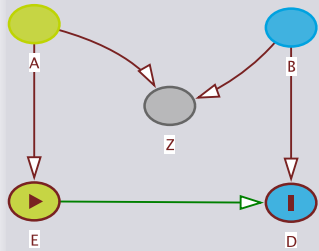
$\mathrm{Cov}(A, C \mid B) = 0$

Data covariance matrix:

	A	B	C
A	1	0.5	0.1
B	0.5	1	0.5
C	0.1	0.5	0

$$\begin{aligned} &\mathrm{Cov}(A, C \mid B) \\ &= \mathrm{Cov}(A, C) - \frac{\mathrm{Cov}(A, B)\,\mathrm{Cov}(C, B)}{\mathrm{Var}(B)} \\ &= 0.1 - 0.5^2 = -0.15 \end{aligned}$$

The model does not fit the data!



Testing for Vanishing Conditional Covariance

In practice, we can also use linear regression to test for a vanishing covariance.
Consider the linear regression model

$$Y = \beta_{XY} X + \beta_{ZY} Z + \epsilon_Z \ .$$

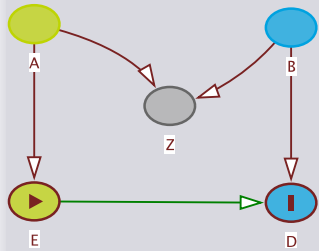
Then

$$\beta_{XY} = \frac{\mathrm{Cov}(X, Y \mid Z)}{\mathrm{Cov}(X, X \mid Z)} \ .$$

Therefore,

$$\beta_{XY} = 0 \Leftrightarrow \mathrm{Cov}(X, Y \mid Z) = 0 \ .$$

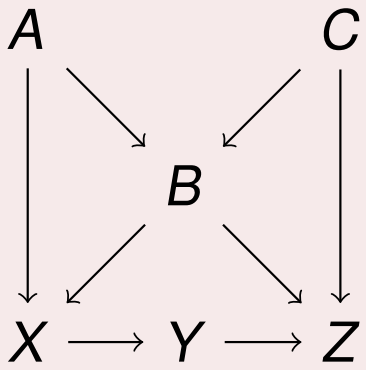
For example, to test whether $\mathrm{Cov}(X, Y \mid Z) = 0$, we can run the regression model $Y \sim X + Z$ and examine the coefficient of X. This is useful because statistical software will often provide confidence intervals and/or tests for regression coefficients.



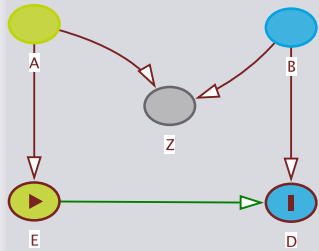
5-Minute Exercise

Exercise

Look at the following SEM again:



Write down 4 different regression equations to test this SEM (the format can be like $Y \sim X + Z$). Which coefficients would need to be 0?

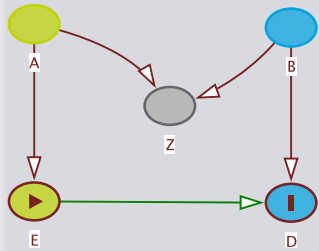


Example: Testing SEMs locally

Testing $B \leftarrow A \rightarrow C$ when the "true network" is $B \leftarrow A \rightarrow C$:

```
set.seed(1234)
library(dagitty)
g <- dagitty("dag{ A -> B [beta=0.5] ; A -> C [beta=0.5] }")
# Simulate data from SEM
N <- 1000
d <- simulateSEM( g, N=1000 )

# Test whether B _||_ C | A
lm( B ~ C + A, data=d )

##
## Call:
## lm(formula = B ~ C + A, data = d)
##
## Coefficients:
## (Intercept)            C            A
##    -0.01790      0.03632      0.52089
```

As expected, the coefficient of C is (close to) 0.

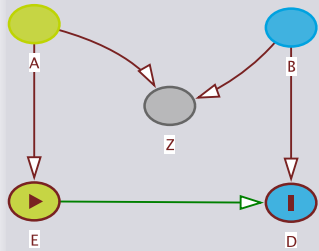


Testing SEMs locally

Testing $B \leftarrow A \rightarrow C$ when the "true network" is $B \rightarrow A \leftarrow C$:

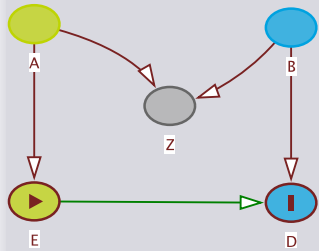
```
set.seed(1234)
library(dagitty)
g <- dagitty("dag{ A <- B [beta=0.5] ; A <- C [beta=0.5] }")
# Simulate data from SEM
N <- 1000
d <- simulateSEM( g, N=1000 )

# Test whether B _||_ C | A
lm( B ~ C + A, data=d )

##
## Call:
## lm(formula = B ~ C + A, data = d)
##
## Coefficients:
## (Intercept)            C            A
##    -0.01912     -0.29732      0.68128
```

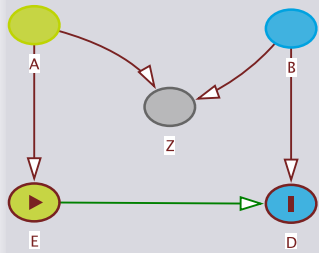
Now, the coefficient of C is no longer 0.



Summary

1. SEMs are Bayesian networks in which the nodes and arrows represent linear-Gaussian functions.
2. SEMs can be fitted node by node, and they can be tested using *d*-separation and linear regression.
3. SEMs are widely used (in certain fields) and mature software packages exist to support them.





Lecture 4

Latent Variables

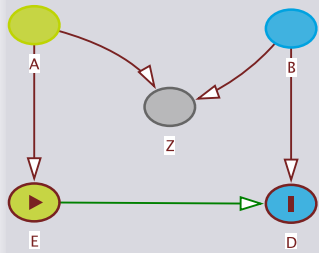
Course Bayesian Networks, October 8th, 2018

Johannes Textor

Institute for Computing and Information Sciences

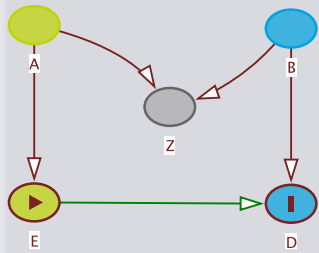
Learning Objectives 4

1. Be able to derive implied covariance matrices from SEMs.
2. Understand the concept of global model fit.
3. Understand how latent variables can be incorporated into SEMs.

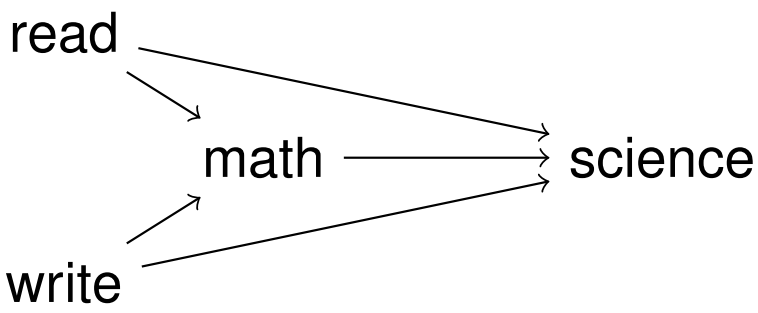


Lecture Outline 4

1. Latent Variables
2. Implied Covariance Matrices
3. Estimating Latent Variables
4. Examples



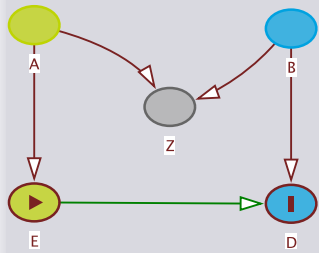
Motivation



Implication:
$\text{read} \perp\!\!\!\perp \text{write}$

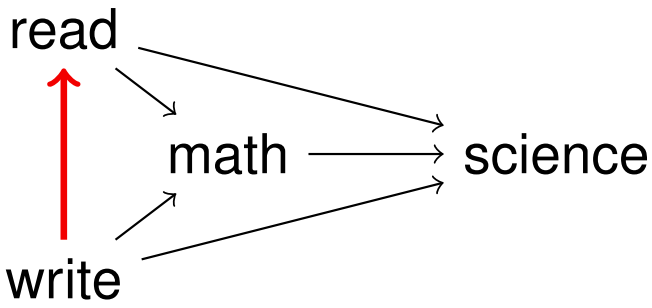
```
hsb2 <- read.table(paste0('https://stats.idre.ucla.edu/',
        'wp-content/uploads/2016/02/hsb2-2.csv'),
        header=T, sep=",")
cor.test( hsb2$read, hsb2$write )

##
##  Pearson's product-moment correlation
##
## data:  hsb2$read and hsb2$write
## t = 10.465, df = 198, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4993831 0.6792753
## sample estimates:
##       cor
## 0.5967765
```

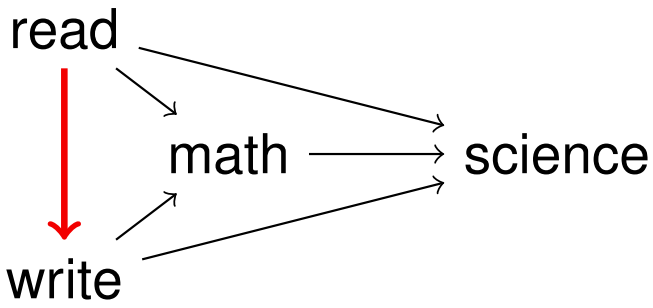
The implication is clearly false, but how do we fix this?



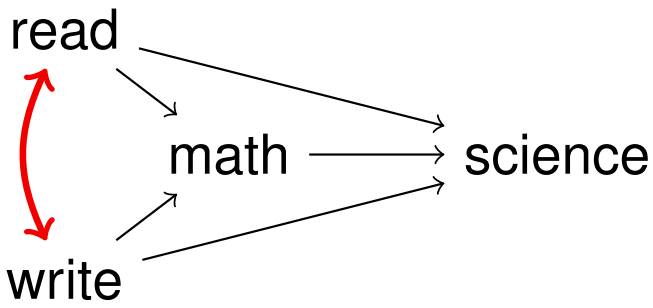
Hidden Common Causes

Possible fixes:

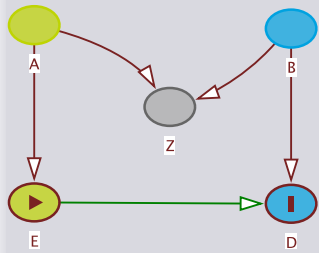




Neither seems plausible – reading and writing are both “basic” skills. In this study, they were measured at the same time. More likely, there is a hidden common cause of the two:



Intuitively: another variable influences both reading and writing skills, but we have not measured it. Therefore, we cannot include it in our model.

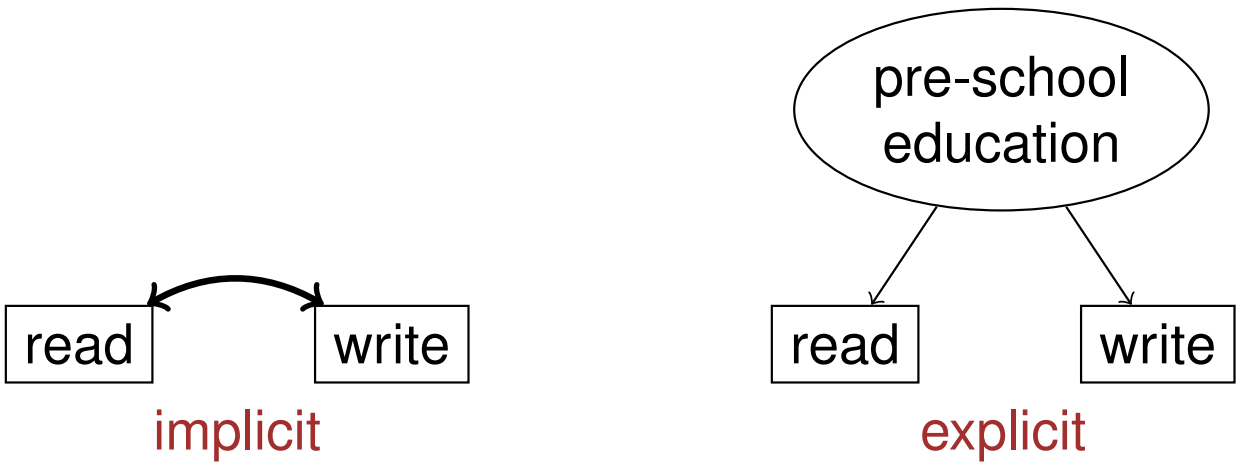


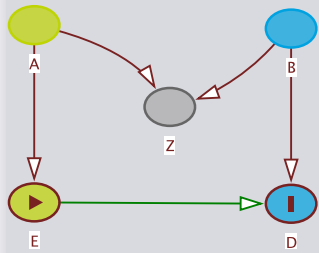
Latent Variables

A latent variable is a variable that is not observed/measured.

- Hidden variables could be measured in principle, but are not present in the data, perhaps for cost or ethical reasons. Example: family income.
- Conceptual variables (constructs) are variables that are not directly measurable quantities, and perhaps exist only in theory. Example: intelligence, charisma.

Often, we think that some latent common cause of two variables exists, but are unable to specify it exactly. We can then represent the latent variable implicitly using a bi-directed arrow. Otherwise, we can depict the latent variable explicitly.



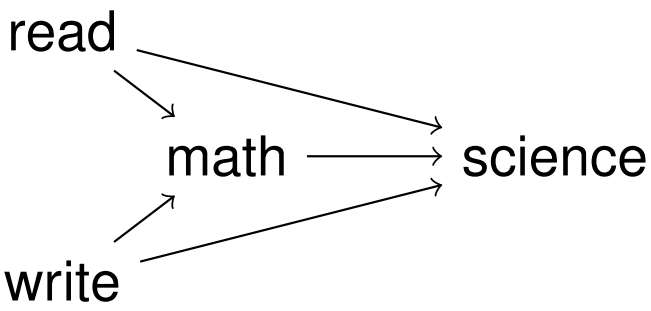


Representing Latent Variables Nonparametrically

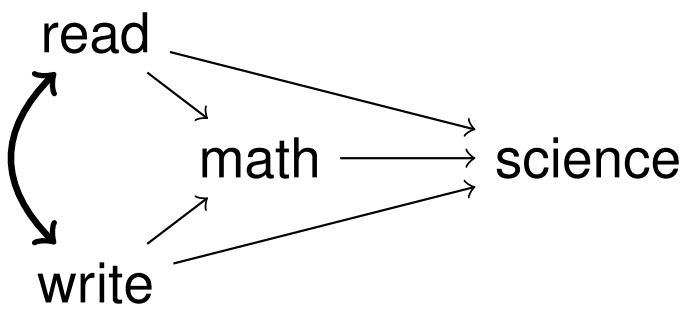
How can we represent a variable that is not observed in a Bayesian network?
In a general Bayesian network, we can represent some latent variables implicitly by joint probability tables.

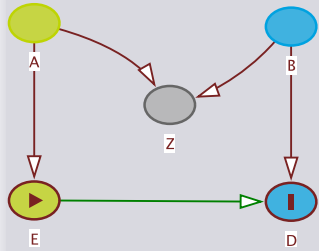
read	*P*
good	0.5
poor	0.5

write	*P*
good	0.5
poor	0.5



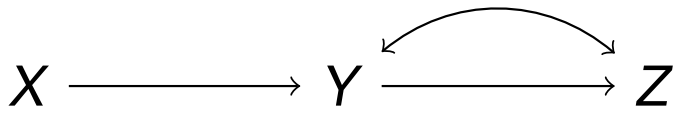
read	write	*P*
good	good	0.3
good	poor	0.2
poor	good	0.2
poor	poor	0.3



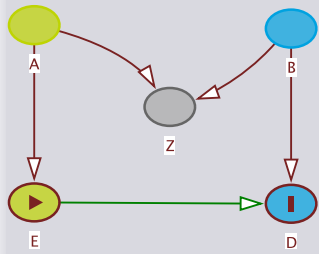


Representing Latent Variables

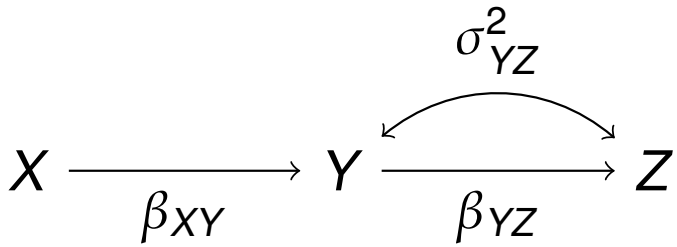
The representation as joint probability tables only works if the two variables linked by $\leftrightarrow$ have no other parents. Here's an example where this approach fails:



Y and Z depend on a latent variable, but Y also depends on X. There is no straightforward way to express this in a probability table except a full joint table for X, Y, Z – which renders the network useless.
(One) solution: Use parametric representations of latent variables.

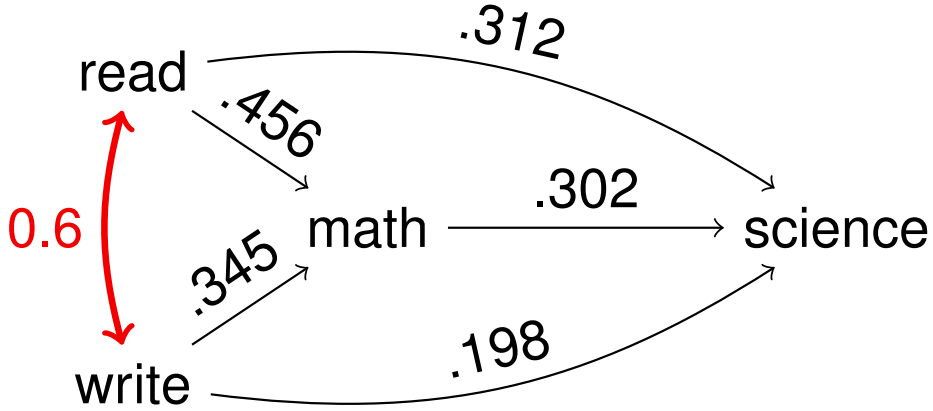


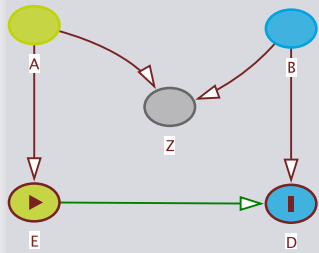
Representing Latent Variables in Structural Equation Models



$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z
\end{aligned}
$$

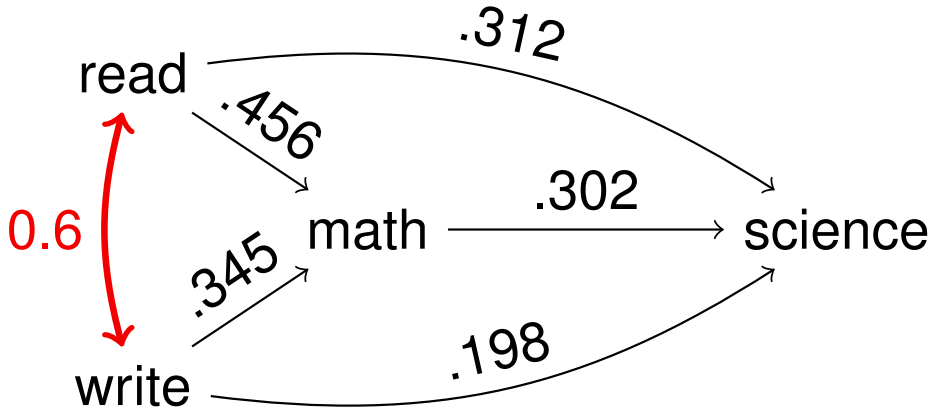
In a regular SEM, we assume that $\mathrm{Cov}(\epsilon_Y, \epsilon_Z) = 0$.
We can relax this assumption, and introduce a new parameter: the residual covariance $\mathrm{Cov}(\epsilon_Y, \epsilon_Z) = \sigma^2_{YZ}$.



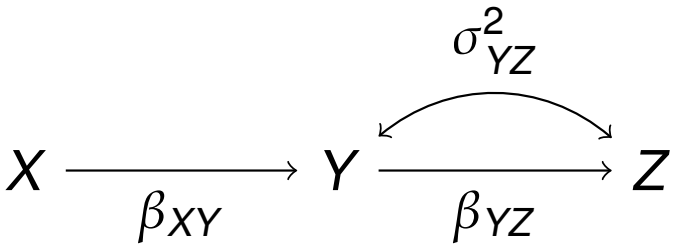


Estimating Residual Covariances

How can we estimate residual covariances from data? In some simple cases, the residual covariance can be set to the observed covariance between the two variables.

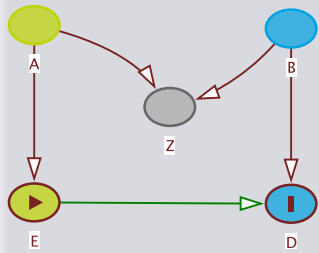


In other cases, that does not work.



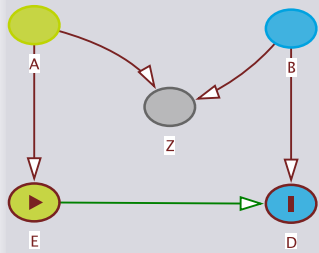
$$\mathrm{Cov}(Y,Z) = \beta_{YZ} + \sigma^2_{YZ}\,.$$

Before showing how we can estimate the residual covariance in such cases, we first need to learn about a new concept: global estimation.



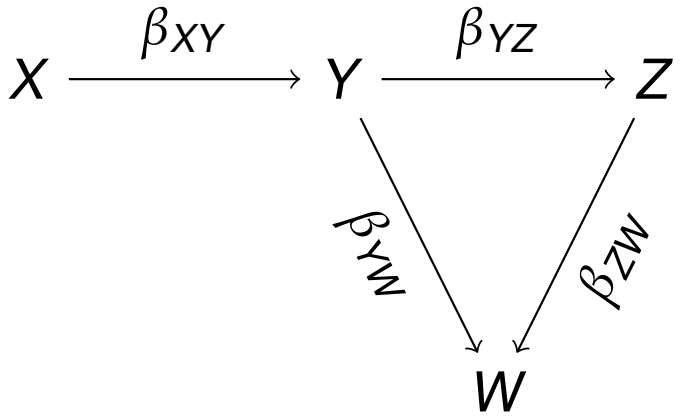
Global Estimation of Bayesian Networks

- Bayesian network parameter estimation normally works locally at each node, conditioned its parents.
- For many classes of parametric models, parameters can also be estimated globally. This means that all parameters are fitted simultaneously to the data, often using a numerical optimization algorithm.
- Global estimation can sometimes be used even if some variables are not observed.
- For structural equation models, global estimation works through the implied covariance matrix.

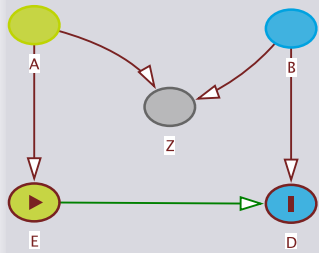


Reduction to Residuals

As we saw in the previous lecture, every variable in a SEM can be expressed as a linear combination of residuals.

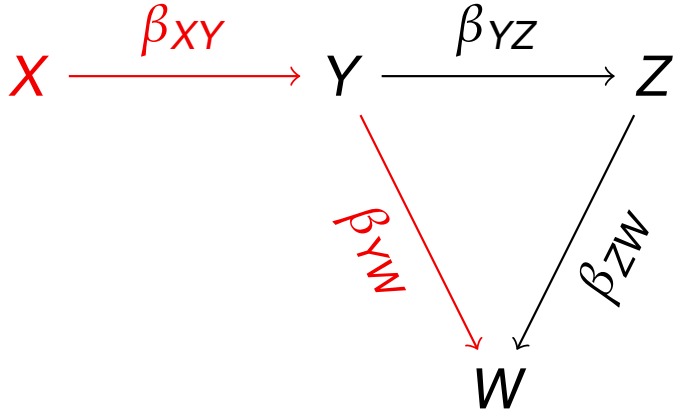


$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
&= \beta_{XY}\epsilon_X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z \\
&= \beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z \\
W &= \beta_{YW} Y + \beta_{ZW} Z + \epsilon_W \\
&= \beta_{YW}\beta_{XY}\epsilon_X + \beta_{YW}\epsilon_Y + \beta_{ZW}\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{ZW}\beta_{YZ}\epsilon_Y + \beta_{ZW}\epsilon_Z + \epsilon_W
\end{aligned}
$$

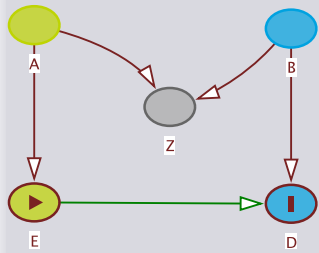


Reduction to Residuals

As we saw in the previous lecture, every variable in a SEM can be expressed as a linear combination of residuals.

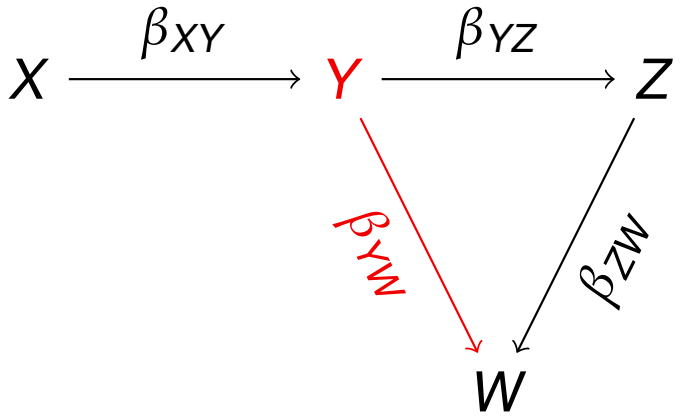


$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
&= \beta_{XY}\epsilon_X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z \\
&= \beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z \\
W &= \beta_{YW} Y + \beta_{ZW} Z + \epsilon_W \\
&= \beta_{YW}\beta_{XY}\epsilon_X + \beta_{YW}\epsilon_Y + \beta_{ZW}\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{ZW}\beta_{YZ}\epsilon_Y + \beta_{ZW}\epsilon_Z + \epsilon_W
\end{aligned}
$$

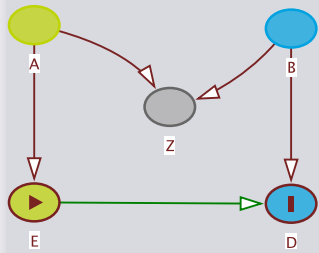


Reduction to Residuals

As we saw in the previous lecture, every variable in a SEM can be expressed as a linear combination of residuals.

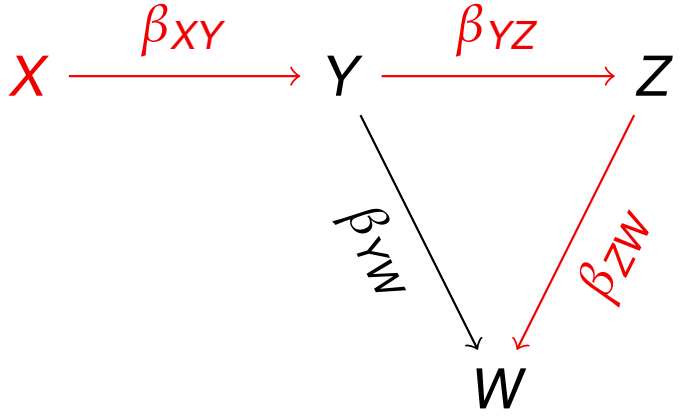


$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
&= \beta_{XY}\epsilon_X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z \\
&= \beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z \\
W &= \beta_{YW} Y + \beta_{ZW} Z + \epsilon_W \\
&= \beta_{YW}\beta_{XY}\epsilon_X + \beta_{YW}\epsilon_Y + \beta_{ZW}\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{ZW}\beta_{YZ}\epsilon_Y + \beta_{ZW}\epsilon_Z + \epsilon_W
\end{aligned}
$$

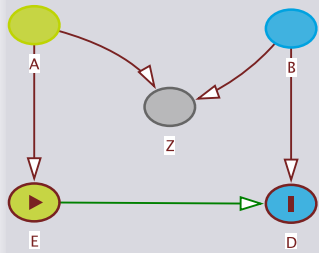


Reduction to Residuals

As we saw in the previous lecture, every variable in a SEM can be expressed as a linear combination of residuals.

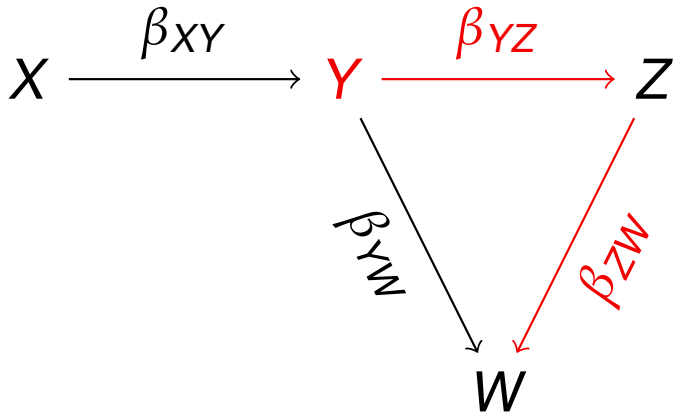


$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
&= \beta_{XY}\epsilon_X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z \\
&= \beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z \\
W &= \beta_{YW} Y + \beta_{ZW} Z + \epsilon_W \\
&= \beta_{YW}\beta_{XY}\epsilon_X + \beta_{YW}\epsilon_Y + \beta_{ZW}\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{ZW}\beta_{YZ}\epsilon_Y + \beta_{ZW}\epsilon_Z + \epsilon_W
\end{aligned}
$$

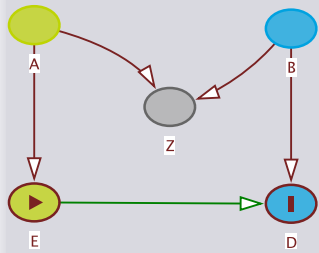


Reduction to Residuals

As we saw in the previous lecture, every variable in a SEM can be expressed as a linear combination of residuals.

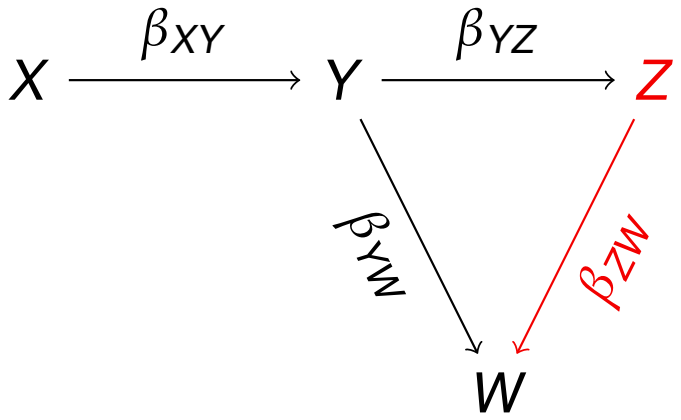


$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
&= \beta_{XY}\epsilon_X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z \\
&= \beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z \\
W &= \beta_{YW} Y + \beta_{ZW} Z + \epsilon_W \\
&= \beta_{YW}\beta_{XY}\epsilon_X + \beta_{YW}\epsilon_Y + \beta_{ZW}\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{ZW}\beta_{YZ}\epsilon_Y + \beta_{ZW}\epsilon_Z + \epsilon_W
\end{aligned}
$$

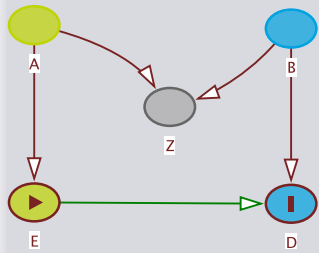


Reduction to Residuals

As we saw in the previous lecture, every variable in a SEM can be expressed as a linear combination of residuals.

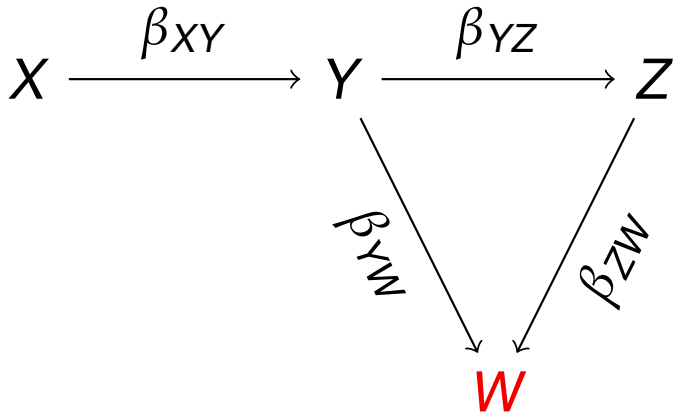


$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
&= \beta_{XY}\epsilon_X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z \\
&= \beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z \\
W &= \beta_{YW} Y + \beta_{ZW} Z + \epsilon_W \\
&= \beta_{YW}\beta_{XY}\epsilon_X + \beta_{YW}\epsilon_Y + \beta_{ZW}\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{ZW}\beta_{YZ}\epsilon_Y + \beta_{ZW}\epsilon_Z + \epsilon_W
\end{aligned}
$$

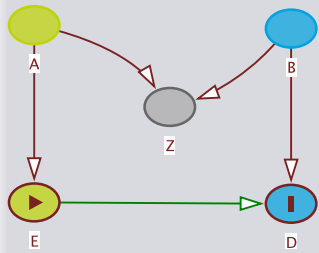


Reduction to Residuals

As we saw in the previous lecture, every variable in a SEM can be expressed as a linear combination of residuals.



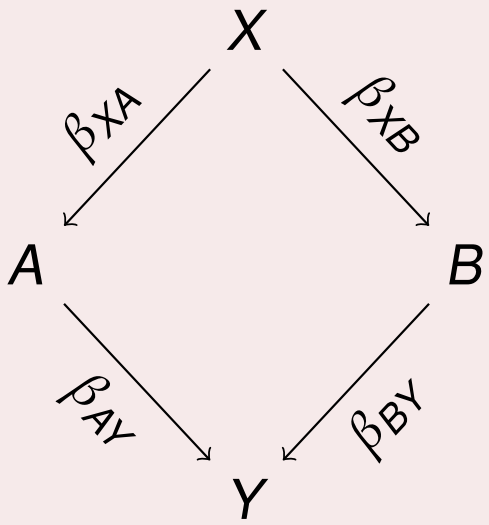
$$
\begin{aligned}
X &= \epsilon_X \\
Y &= \beta_{XY} X + \epsilon_Y \\
&= \beta_{XY}\epsilon_X + \epsilon_Y \\
Z &= \beta_{YZ} Y + \epsilon_Z \\
&= \beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z \\
W &= \beta_{YW} Y + \beta_{ZW} Z + \epsilon_W \\
&= \beta_{YW}\beta_{XY}\epsilon_X + \beta_{YW}\epsilon_Y + \beta_{ZW}\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{ZW}\beta_{YZ}\epsilon_Y + \beta_{ZW}\epsilon_Z + \epsilon_W
\end{aligned}
$$

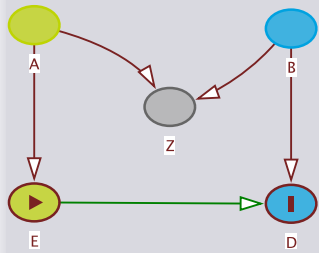


5-Minute Exercise

Exercise

Write down Y as a linear combination of residuals in the following SEM.





The Ancestor Rule

In general, we can express each variable X in a SEM S as follows:

$$X = \sum_{A \in \mathrm{An}(X)} \sum_{\text{paths } p \text{ from } A \text{ to } X} \lambda_p \epsilon_A \ ,$$

where

$$\mathrm{An}(X) = \{A \mid A \text{ is an ancestor of } X \text{ in } S\}$$

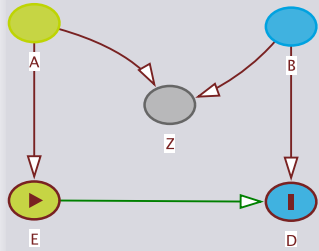
and

$$\lambda_p = \prod_{U \to V \text{ on } p} \beta_{UV} \ .$$

For example, if $p = A \to B \to C$, then $\lambda_p = \beta_{AB}\beta_{BC}$.

The ancestor rule

To compute the reduction to residuals r of X:
$r := 0$
For each ancestor A of X:
    For each path p from A to X:
        $r :=$ r + (product of all coefficients on p) $\mathrm{Var}(\epsilon_A)$



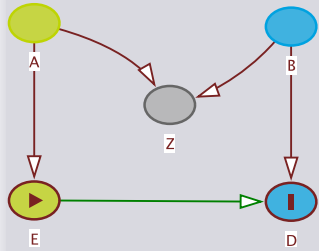
10-Minute Exercise

Exercise

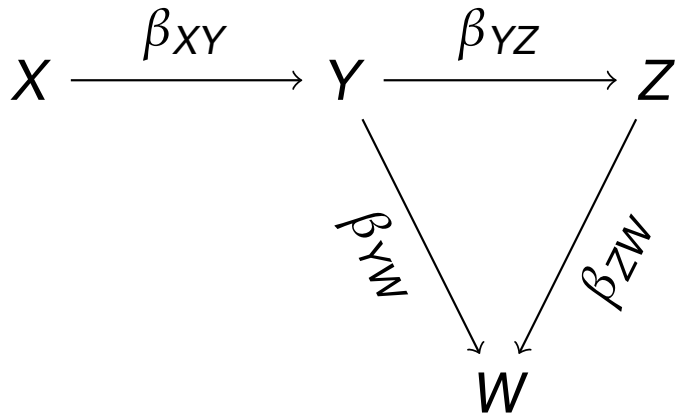
Often, structural equation models are applied to “scaled” data, where all variances are 1. In that case, the residual parameter variances are often omitted from the model.

In the following model, what does the value of σ^2_Z need to be such that the variance of Z is 1?

$$X \xrightarrow[0.5]{1} Y \xrightarrow[0.5]{1} Z \qquad \text{?}$$

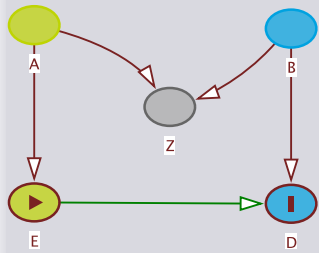


Implied Covariances



Assuming all variables have mean 0, we can write that

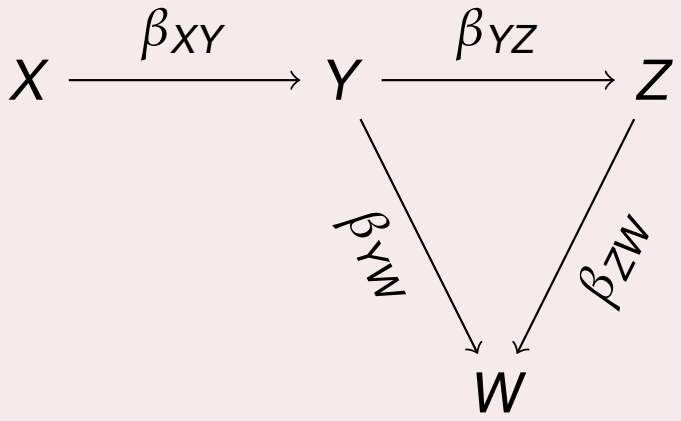
$$
\begin{aligned}
&\mathrm{Cov}(X,Z) = \mathrm{E}(XZ) \\
=&\mathrm{E}(\epsilon_X(\beta_{YZ}\beta_{XY}\epsilon_X + \beta_{YZ}\epsilon_Y + \epsilon_Z)) \\
=&\mathrm{E}(\epsilon_X\beta_{YZ}\beta_{XY}\epsilon_X + \epsilon_X\beta_{YZ}\epsilon_Y + \epsilon_X\epsilon_Z) \\
=&\mathrm{E}(\epsilon_X\beta_{YZ}\beta_{XY}\epsilon_X) + \mathrm{E}(\epsilon_X\beta_{YZ}\epsilon_Y) + \mathrm{E}(\epsilon_X\epsilon_Z) \\
=&\beta_{YZ}\beta_{XY}\mathrm{E}(\epsilon_X\epsilon_X) + \beta_{YZ}\mathrm{E}(\epsilon_X\epsilon_Y) + \mathrm{E}(\epsilon_X\epsilon_Z) \\
=&\beta_{YZ}\beta_{XY}\mathrm{Var}(\epsilon_X) + \beta_{YZ}\mathrm{Cov}(\epsilon_X,\epsilon_Y) + \mathrm{Cov}(\epsilon_X,\epsilon_Z) \\
=&\beta_{YZ}\beta_{XY}\mathrm{Var}(\epsilon_X) + 0 + 0
\end{aligned}
$$

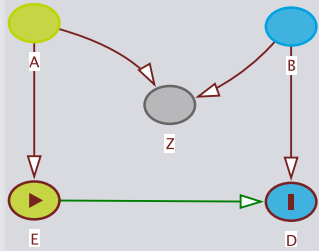


5-Minute Exercise

Exercise

Please determine $\mathrm{Cov}(Y, Z)$!





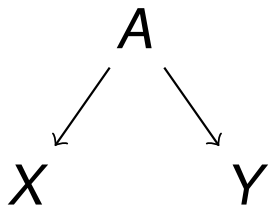
The Trek Rule

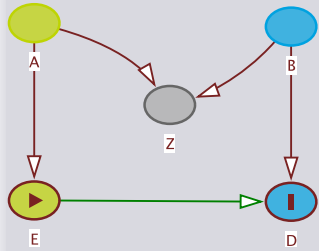
The trek rule is a general algorithm for computing implied covariances $\mathrm{Cov}(X, Y)$ in SEMs. It is based on considering all paths from common ancestors of X and Y to X and Y.

The trek rule

For each common ancestor A of X and Y:
  For each path p_X from A to X:
    For each path p_Y from A to Y:
      $r :=$ r + (product of all coefficients on p_X)
        (product of all coefficients on p_Y) $\mathrm{Var}(\epsilon_A)$

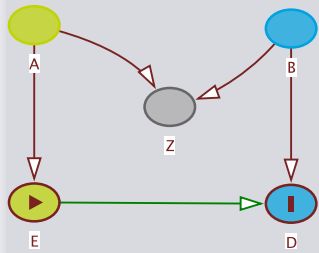
The name trek refers to the path pair (p_X, p_Y). We view these as a trek up and down a mountain:





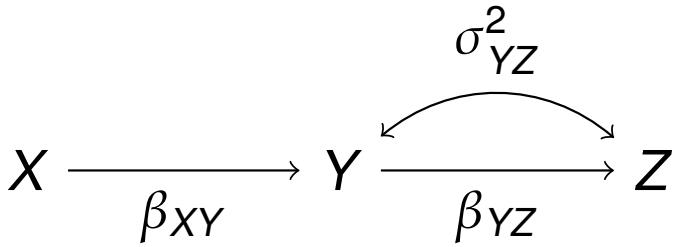
Example Implementation of the Trek Rule

```
library( dagitty )
g <- dagitty( "dag { U -> { X -> Y } ; Y -> Z  }" )
common.ancestors <- intersect( ancestors( g, "Y" ),
                               ancestors( g, "Z" ) )
for( a in common.ancestors ){
   for( p1 in paths( g, a, "Y", directed=TRUE )$paths ){
      for( p2 in paths( g, a, "Z", directed=TRUE )$paths ){
         cat( p1, " ; ",  p2, "\n")
      }
   }
}

## Y  ;  Y -> Z
## X -> Y  ;  X -> Y -> Z
## U -> X -> Y  ;  U -> X -> Y -> Z
## U -> X -> Y  ;  U -> Y -> Z
## U -> Y  ;  U -> X -> Y -> Z
## U -> Y  ;  U -> Y -> Z
```

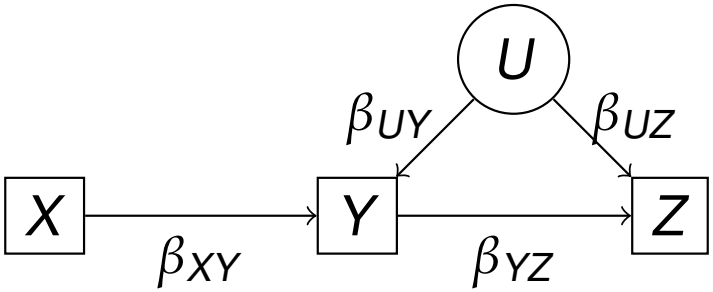



Estimating Latent Variables

Now let us return to our initial question: How do we estimate the parameters of this model?



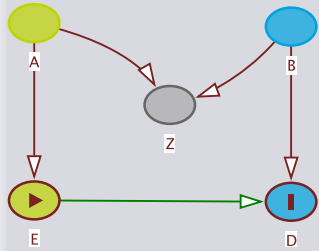
First, we replace the bi-directed arrow by a latent variable.



We now compute the implied covariance matrix of this model, with respect to the observed variables X, Y, and Z.

The covariance matrix has 6 unique entries: $\mathrm{Var}(X)$, $\mathrm{Var}(Y)$, $\mathrm{Var}(Z)$, $\mathrm{Cov}(X,Y)$, $\mathrm{Cov}(Y,Z)$, $\mathrm{Cov}(X,Z)$.

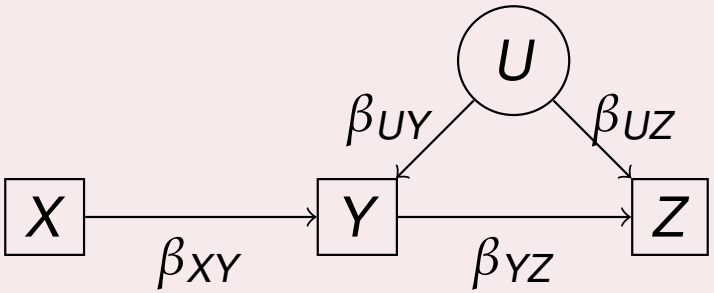
And we have 8 parameters: $\sigma^2_X, \sigma^2_Y, \sigma^2_Z, \sigma^2_U, \beta_{XY}, \beta_{YZ}, \beta_{UY}, \beta_{UZ}$.



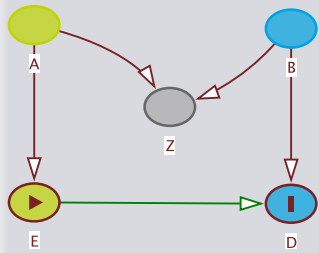
10-Minute Exercise

Exercise

Are there two different parameter combinations for this SEM that generate exactly the same covariance matrix?



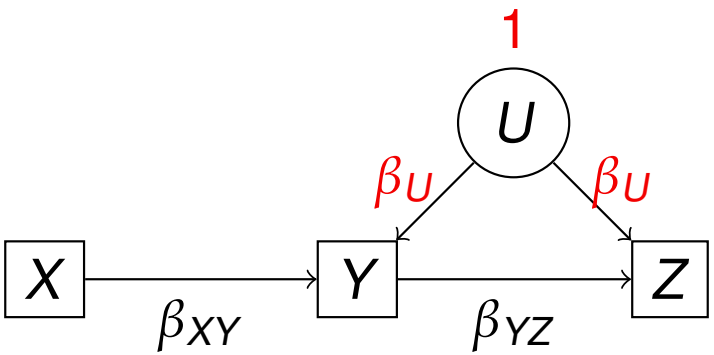
Hint: Suppose you set all parameters to 1 except β_{UY} and β_{UZ}. Can you find two different settings for β_{UY} and β_{UZ} that should give exactly the same covariance matrix?



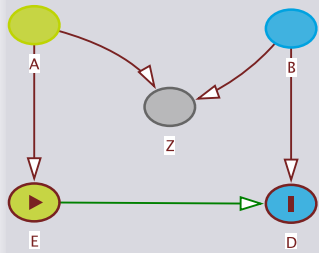
Identification

If there are more unknowns (=parameters) than data points (=covariance matrix entries), then the model is not identified. In this case, we need to impose some constraints on the parameters.
Typical constraints:

- $\mathrm{Var}(U) = 1$ (we can't know the unit of something we don't measure)
- $\beta_{UY} = \beta_{UZ}$ (we can't distinguish between the two effects)

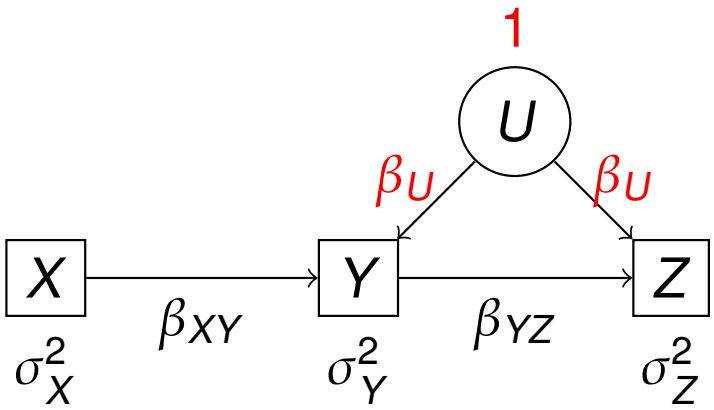


This brings us down to 6 parameters for 6 covariance entries.
The resulting model is (just) identified.



Example Usage of the Trek Rule (1)

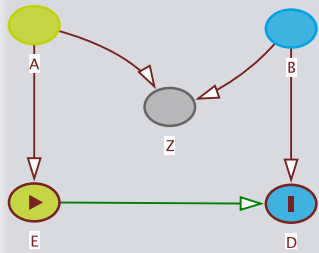
Let' use the trek rule to devise the off-diagonal elements of this model's implied covariance matrix:



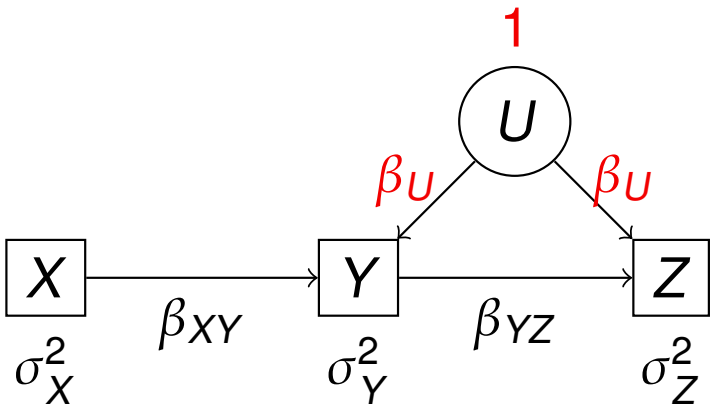
Simplest example: $\text{Cov}(X, Y)$

Common ancestors of X and Y	X
Paths from X to X	X
Paths from X to Y	$X \rightarrow Y$
Treks from X to Y	$X \rightarrow Y$

So $\text{Cov}(X, Y) = \sigma^2_X \beta_{XY}$.



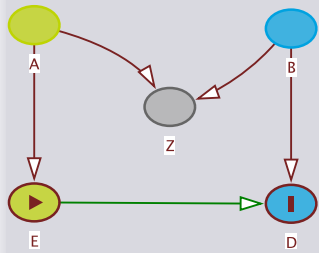
Example Usage of the Trek Rule (2)



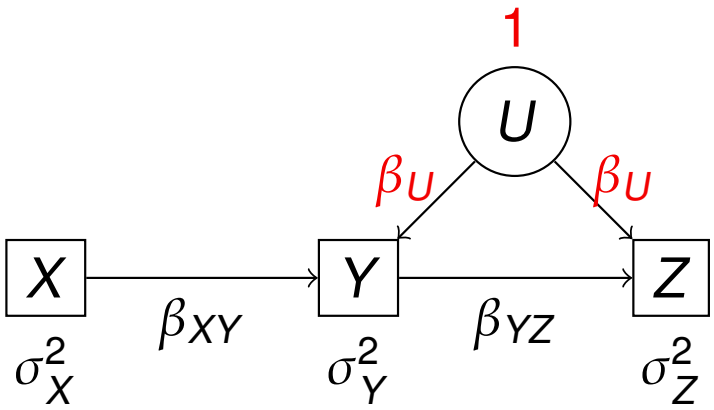
$$\text{Cov}(X, Z)$$

Common ancestors of X and Z	X
Paths from X to X	X
Paths from X to Y	$X \to Y \to Z$
Treks from X to Z	$X \to Y \to Z$

So $\text{Cov}(X, Z) = \sigma^2_X \beta_{XY} \beta_{YZ}$.

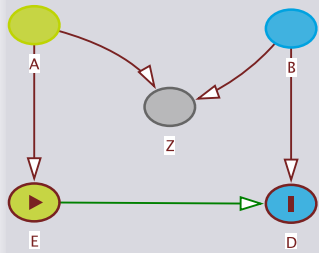


Example Usage of the Trek Rule (3)



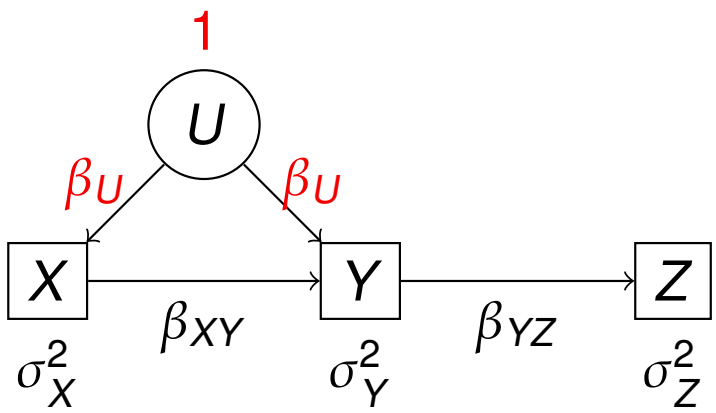
Common ancestors of Y and Z	X, Y, U
Paths from X to Y Paths from X to Z	$X \to Y$ $X \to Y \to Z$
Paths from Y to Y Paths from Y to Z	Y $Y \to Z$
Paths from U to Y Paths from U to Z	$U \to Y$ $U \to Z, U \to Y \to Z$
Treks from X to Z	$Y \leftarrow X \to Y \to Z, Y \to Z,$ $Y \leftarrow U \to Z, Y \leftarrow U \to Y \to Z$

So $\text{Cov}(Y,Z) = \sigma^2_X \beta^2_{XY} \beta_{YZ} + \sigma^2_Y \beta_{YZ} + \beta^2_U + \beta^2_U \beta_{YZ}$.

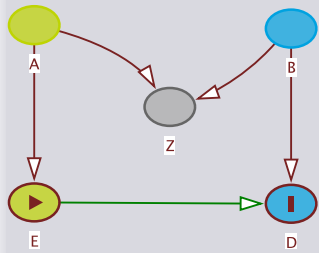


Remark on Identification

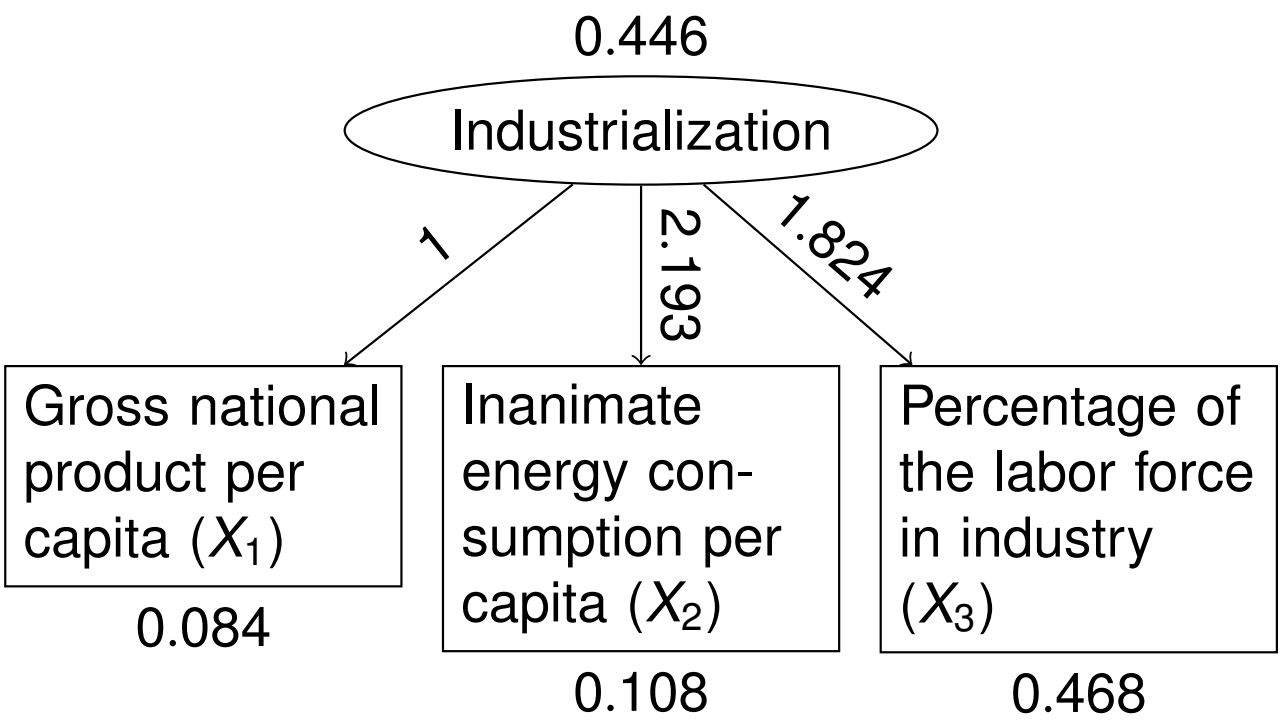
- We have said that models with more parameters than covariance matrix entries are not identified.
- But having no more parameters than covariance matrix entries is not sufficient for having an identified model.



- Deciding whether a model is identified or not is a difficult problem, and an active research topic.

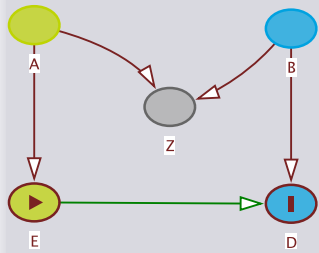


Example: Measurement Model



Example implementation in R

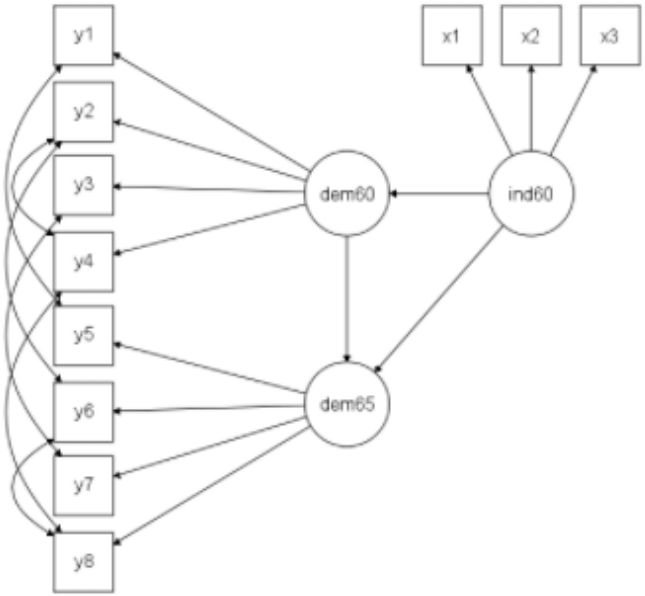
```
suppressMessages( library(lavaan) )
mdl <- sem( 'ind =~ x1 + x2 + x3', data=PoliticalDemocracy )
coef( mdl )

##  ind=~x2  ind=~x3   x1~~x1   x2~~x2   x3~~x3 ind~~ind
##    2.193    1.824    0.084    0.108    0.468    0.446
```

Example: Structural Model

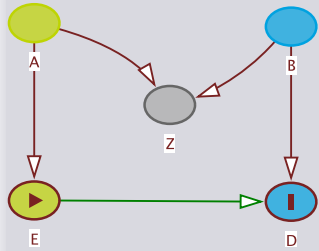
First impression

To get a first impression of how lavaan works in practice, consider the following example of a SEM model. The figure below contains a graphical representation of the model that we want to fit.



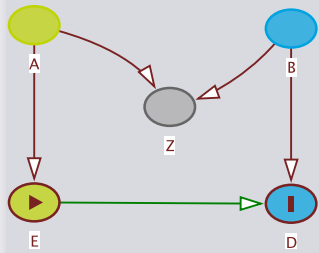

```
model <- '
   # latent variables
     ind60 =~ x1 + x2 + x3
     dem60 =~ y1 + y2 + y3 + y4
     dem65 =~ y5 + y6 + y7 + y8
   # regressions
     dem60 ~ ind60
     dem65 ~ ind60 + dem60
   # residual covariances
     y1 ~~ y5
     y2 ~~ y4 + y6
     y3 ~~ y7
     y4 ~~ y8
     y6 ~~ y8
'
fit <- sem(model,
           data=PoliticalDemocracy)
summary(fit)
```

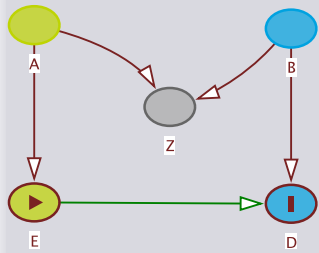
Source: http://lavaan.ugent.be



Summary

1. Latent variables are unobserved, but important variables that affect our observed variables.
2. Because latent variables are not observed, we need to do "smart things" to estimate their parameters.
3. In SEMs, we can estimate latent variables through the implied covariance matrix.





Lecture 5

Markov Equivalence and Structure Learning

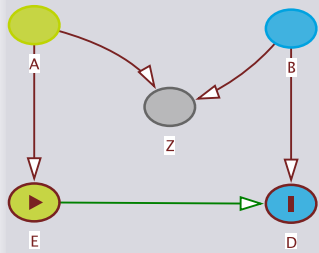
Course Bayesian Networks, November 5th, 2018

Johannes Textor

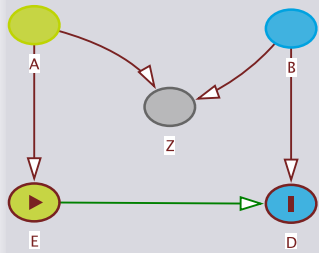
Institute for Computing and Information Sciences

Learning Objectives 5

1. Recognize when two Bayesian networks are statistically equivalent.
2. Understand the concept of “faithfulness” to a probability distribution.
3. Understand how Bayesian network structure can be derived from data.



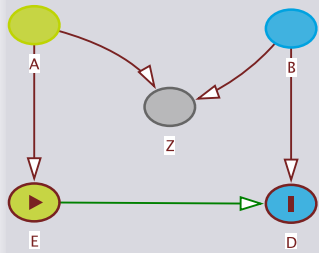
Lecture Outline 5



Today's Topic

So far, we've worked with Bayesian networks assuming they were built by a human expert. But often, we have no strong theory on how a network should look like.
We will now learn about how to derive networks from data algorithmically.
However, there are two fundamental limitations to any such approach, which we first need to appreciate:

- Faithfulness
- Markov Equivalence



Structure Learning: Defining Our Goal

The structure learning problem is defined as follows:

input: a probability distribution P.

output: a Bayesian network G such that P is consistent with G.

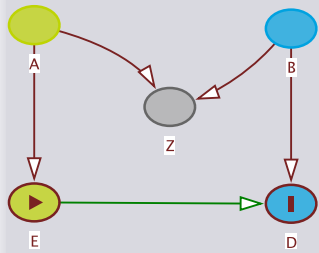
We have seen before that consistency is fully determined by conditional independence – P is consistent with G if and only if each conditional independence implied by G holds in P.

This means that we can rephrase the structure learning problem as follows:

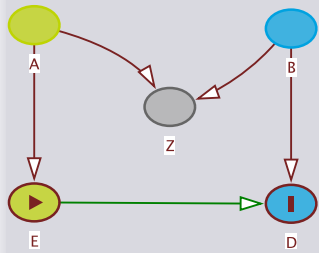
input: a list of all conditional independence statements that hold in a probability distribution P.

output: a Bayesian network G such that P is consistent with G.

(In reality, we will need to infer the conditional independence statements from data.)



5-Minute Exercise



Exercise

Consider the following example structure learning problem for three variables X,M,Y:

input: $X \perp\!\!\!\perp Y \mid M$

output: ?

Please fill in the question mark and provide a valid output for the structure learning problem!

Trivial Consistency

Let us make the previous exercise slightly more complicated. We have four variables A, B, C, D.

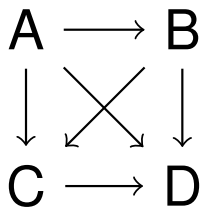
input: $C \perp\!\!\!\perp A \mid B$, $D \perp\!\!\!\perp A, B \mid C$

output: ?

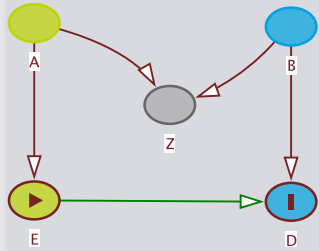
What is the "laziest" solution we can come up with? Recall the definition:

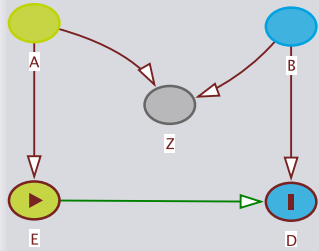
> *P is consistent with G if and only if each conditional independence implied by G holds in P.*

So what if G does not imply any conditional independence?



This is a valid solution, no matter what the input is!





Structure Learning and Model Complexity

Expanding our previous example, the structure learning problem

input: $C \perp\!\!\!\perp A \mid B, D \perp\!\!\!\perp A, B \mid C$

output: ?

is solved by all of the following networks:

Occam's Razor

If there are several possible explanations for a given phenomenon, prefer the simplest explanation.

In our case, the "simplest" (most parsimonious) explanation means that we pick the solution with the fewest edges.

Faithfulness

A Bayesian network G is faithful to a probability distribution P if G implies exactly the set of conditional independences that hold in P.

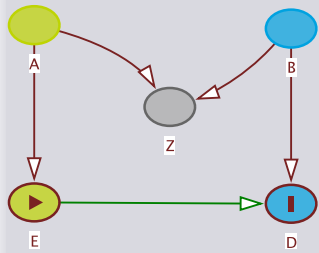
Faithful structure learning problem

input: a list of all conditional independence statements that hold in a probability distribution P.

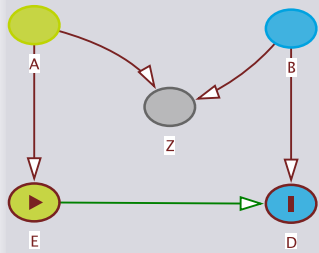
output: a Bayesian network G such that P is faithful to G.

Faithfulness is a heuristic – there is often no reason to assume that only "faithful" network structures are interesting! In particular, the true "causal" Bayesian network that has generated P (if it exists) need not necessarily be faithful to P.

However, faithfulness is a useful heuristic because it delivers sparse networks.



5-Minute Exercise



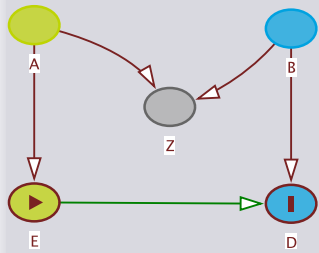
Exercise

Please provide a faithful solution for the following structure learning problem for variables A,B,C:

input: $A \perp\!\!\!\perp C$

output: ?

The Concept of Markov Equivalence



Consider the following two Bayesian networks:

$$X \longrightarrow M \longrightarrow Y \qquad\qquad X \longleftarrow M \longleftarrow Y$$

Both of these networks imply one, and only one, conditional independence:

$$X \perp\!\!\!\perp Y \mid M$$

This means that both networks are consistent with exactly the same probability distributions. Such networks are called Markov equivalent.

If there are several Markov equivalent Bayesian networks G that fulfil an input list of conditional independences, then there is no unique output for the faithful structure learning problem.

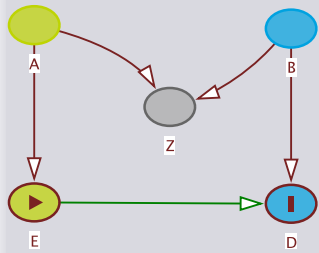
Markov Equivalence Classes

The set $[G]$ of all Bayesian networks G' that is equivalent to some Bayesian network G is called the Markov equivalence class of G.
The Markov equivalence class can contain only one network; in those cases, there is in fact a unique faithful solution to the structure learning problem! For example:

$$\left[X \nearrow M \nwarrow Y \right] = \left\{ X \nearrow M \nwarrow Y \right\}$$

But most often, Markov equivalence classes contain several networks.

$$\left[X \nearrow M \searrow Y \right] = \left\{ X \swarrow M \nwarrow Y,\ X \swarrow M \searrow Y,\ X \nearrow M \searrow Y \right\}$$



Characterizing Markov Equivalence

The skeleton of a Bayesian network $G = (\mathbf{V}, \mathbf{E})$ is the undirected graph $G_s = (\mathbf{V}, \mathbf{E}_s)$ in which $\mathbf{E}_s = \{i - j \mid i \to j \in \mathbf{E}\}$. (The skeleton is obtained from G by replacing all arrows with undirected edges.)

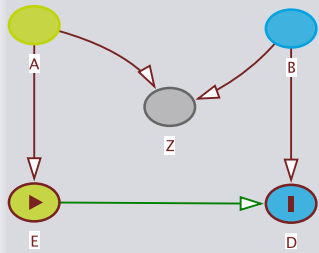
A v-structure in a Bayesian network $G = (\mathbf{V}, \mathbf{E})$ is a set of three variables U, V, W such that $U \to V \leftarrow W$ is an induced subgraph of G. (Thus, $U \to V \in \mathbf{E}$, $W \to V \in \mathbf{E}$, $U \to W \notin \mathbf{E}$, and $W \to U \notin \mathbf{E}$.)

Theorem (Verma and Pearl, 1990)

Two Bayesian networks $G = (\mathbf{V}, \mathbf{E}_G)$ and $H = (\mathbf{V}, \mathbf{E}_H)$ are Markov equivalent if and only if

- *G and H have the same skeleton; and*
- *G and H have the same v-structures.*

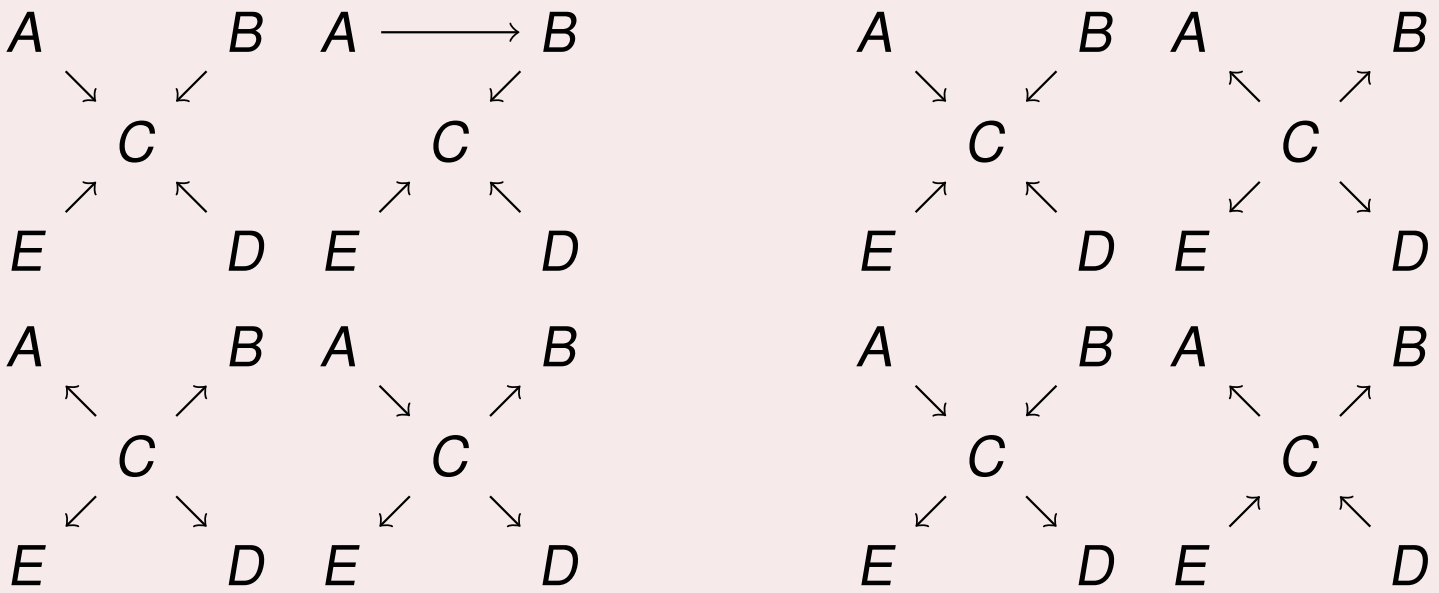
For the proof, see https://arxiv.org/pdf/1304.1108.pdf

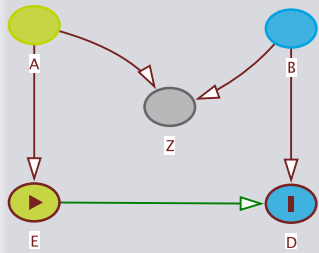


5-Minute Exercise

Exercise

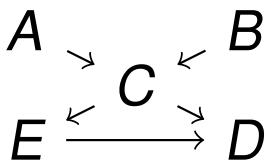
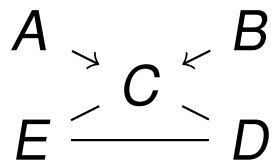
For each of the following pairs of Bayesian networks, please decide whether or not they are Markov equivalent.



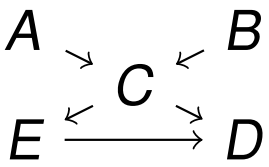
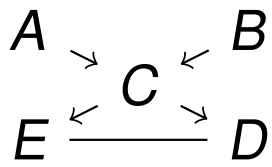


Graph Patterns and CPDAGs

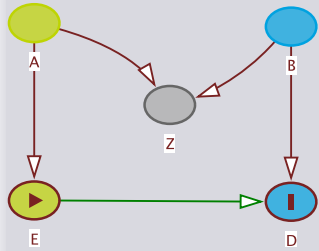
A simple way to represent an equivalence class is a mixed graph in which only the edges that are part of v-structures are drawn directed, and the other edges undirected. Such a mixed graph is called a graph pattern.

For example, the graph pattern of  is  .

Often, undirected edges in a graph pattern can only be oriented in one direction without creating a new v-structure (which would change the equivalence class). If all such edges are replaced by directed edges, we obtain the completed partial DAG or CPDAG.

For example, the CPDAG of  is  .

A CPDAG is a good way to represent all possible outputs of a faithful structure learning problem.



Faithful Structure Learning Finalized

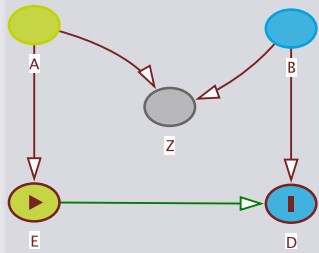
We have now learned about two fundamental limitations of the structure learning problem: (Un)faithfulness, and Markov equivalence.
Keeping in mind these limitations, we can now formulate our final version of the problem, which reflects the extent to which we can actually solve it:

Faithful structure learning problem

input: a list of all conditional independence statements that hold in a probability distribution P.

output: a CPDAG G such that P is faithful to all DAGs represented by G.

We will now learn about an algorithm that solves this kind of structure learning problem.



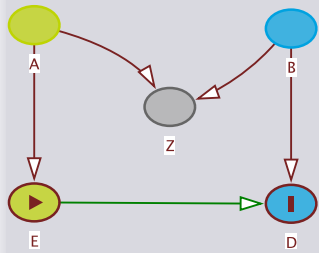
The IC Algorithm

The inferred causation algorithm (Pearl and Verma, 1994) proceeds in two steps. It first infers the skeleton, and then the v-structures.

The IC Algorithm

- Start with a graph containing no edges.
- For each pair of variables X and Y, search for a set $\mathbf{Z}_{XY}$ such that the independence $X \perp\!\!\!\perp Y \mid \mathbf{Z}_{XY}$ is either in the input list, or follows from those in the input list. If no such set exists, link X and Y by an undirected edge $X - Y$.
- For each pair of variables X and Y that are not linked, but have a common neighbour W ($X - W - Y$), check whether $W \in \mathbf{Z}_{XY}$. If not, then add arrowheads pointing to W, i.e. ($X \rightarrow W \leftarrow Y$).
- Orient the resulting graph pattern into a CPDAG.

The original paper and correctness proof is at:
http://ftp.cs.ucla.edu/pub/stat_ser/r156-reprint.pdf



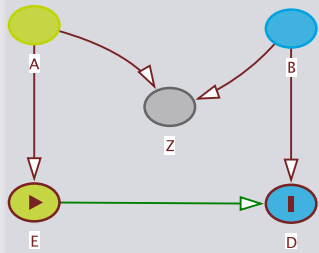
5-Minute Exercise

Exercise

Apply the IC algorithm to derive a graph pattern for the following set of conditional independences:

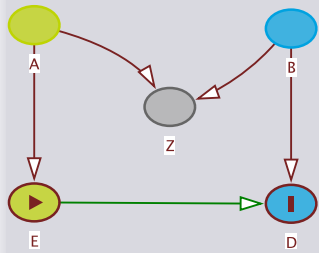
- $C \perp\!\!\!\perp E, D \mid A, B$
- $E \perp\!\!\!\perp B, C \mid A$
- $D \perp\!\!\!\perp A, C \mid B, E$

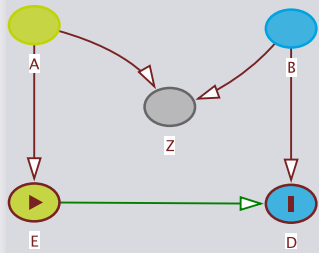
Keep in mind the decomposition property of conditional independence – e.g., $C \perp\!\!\!\perp E, D \mid A, B$ implies both $C \perp\!\!\!\perp E \mid A, B$ and $C \perp\!\!\!\perp D \mid A, B$.



Summary

1. Different Bayesian networks can imply exactly the same constraints on their compatible probability distributions.
2. Therefore, it most often not possible to uniquely identify Bayesian network structures from data.
3. Instead, the IC algorithm can identify equivalence classes of Bayesian networks from data.





Lecture 6

Structure Learning Algorithms

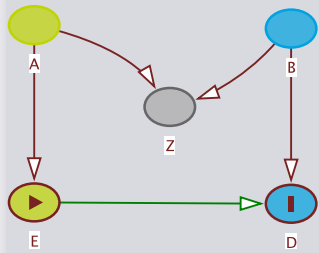
Course Bayesian Networks, November 12th, 2018

Johannes Textor

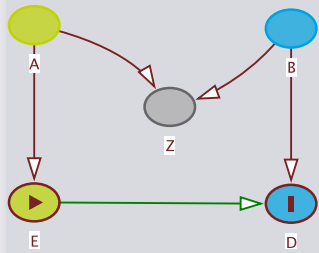
Institute for Computing and Information Sciences

Learning Objectives 6

1. Understand the principles and limitations of conditional independence testing for continuous data.
2. Apply structure learning ideas in practice.
3. Efficiently implement the IC algorithm.



Lecture Outline 6



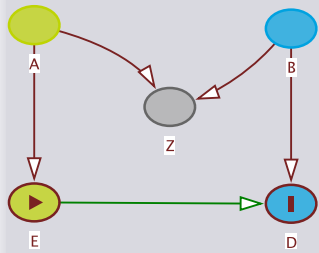
1. Testing Conditional Independence

2. The PC Algorithm

The IC Algorithm

The inferred causation algorithm (Pearl and Verma, 1994) operates on a list of conditional independencies.

But how do we obtain the list of conditional independencies?



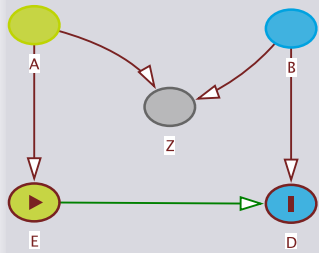
Testing Conditional Independence

In practice, we do not have an input list of conditional independencies. Instead, we need to test conditional independence statistically. This is no easy task, and a major barrier to structure learning.

We briefly touched this topic before, and we know that a statement like $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ can be tested by examining the coefficient of Y in a regression equation

$$X \sim Y + \mathbf{Z}$$

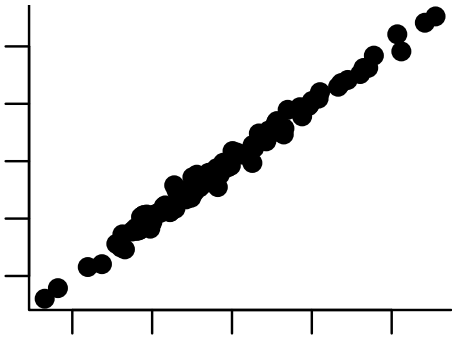
However, this was for normally distributed data only.



Testing Independence

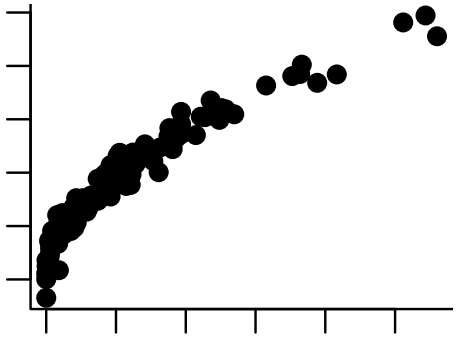
We start with the "simplest" case: independence statements of the form $X \perp\!\!\!\perp Y$ (thus, $\mathbf{Z} = \emptyset$). An important issue is that dependence can be complex.

linear



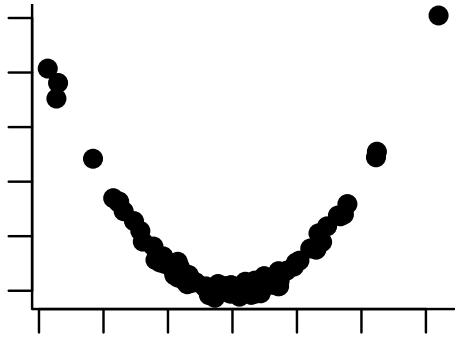
```
lm( y ~ x )$coef[2]

##     x
## 0.997
```

monotone



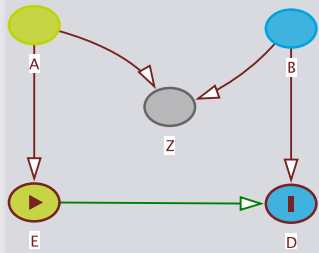
```
lm( y ~ x )$coef[2]

##     x
## 0.443
```

non-monotone



```
lm( y ~ x )$coef[2]

##      x
## -0.158
```

Our linear regression test is designed for linear relations. This can lead to wrong conclusions.

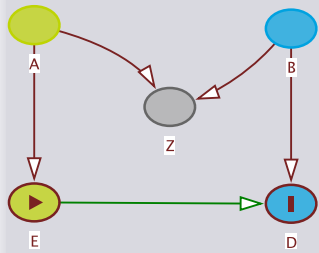


5-Minute Exercise

Exercise

Which, if any, of the below statements is correct?

- If X and Y are statistically independent, then there is no linear dependence between them.
- If X and Y are statistically dependent, then there is a linear dependence between them.
- If there is a linear dependence between X and Y, then X and Y are not statistically independent.
- If there is no linear dependence between X and Y, then X and Y are statistically independent.

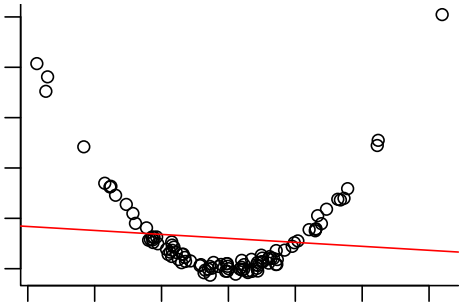


Modelling Non-Linear Relations

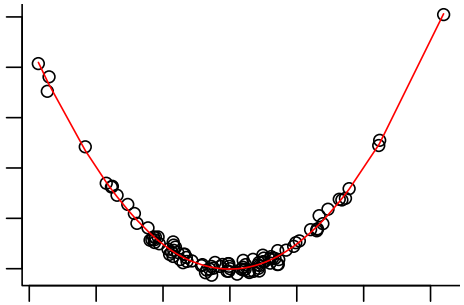
Faced with complex non-linear relationships, one option is to use free-form regression instead of linear regression.

One approach to free-form regression is local polynomial regression, often called LOWESS (locally weighted scatterplot smoothing) or LOESS.

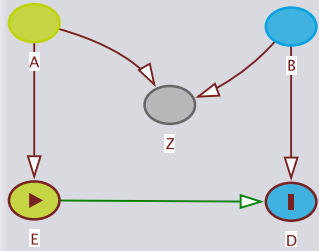
```
m <- lm( y ~ x )
```



```
m <- loess( y ~ x )
```



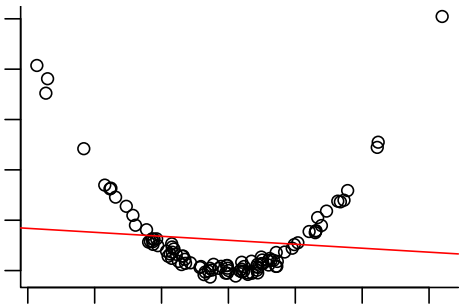
Many functions are locally well approximated by polynomials.



Testing Independence with LOESS

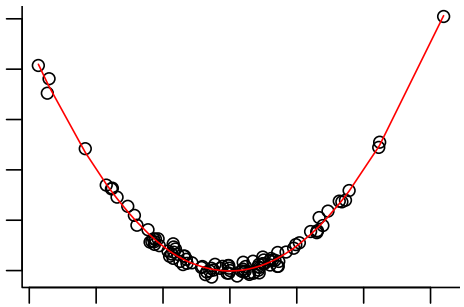
One way to test for dependence is to determine the correlation of the regression predictions $E[Y \mid X]$ with the actual values Y. This works for both linear and non-linear regression.

```
m <- lm( y ~ x )
cor( predict( m ), y )
```



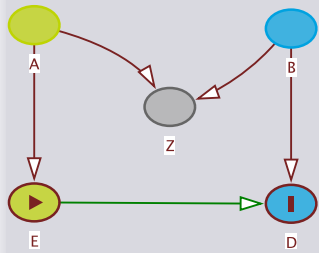
```
## [1] 0.0948
```

```
m <- loess( y ~ x )
cor( predict( m ), y )
```



```
## [1] 0.996
```

We can assess the statistical significance of these correlations using a permutation test.



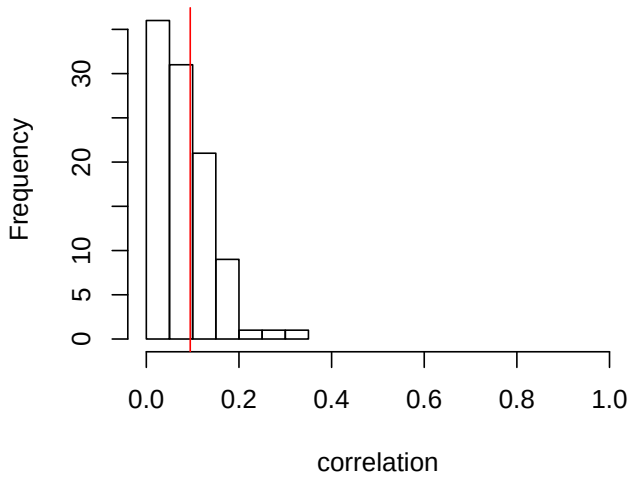
Testing Independence with LOESS

Permutation test for independence between X and Y

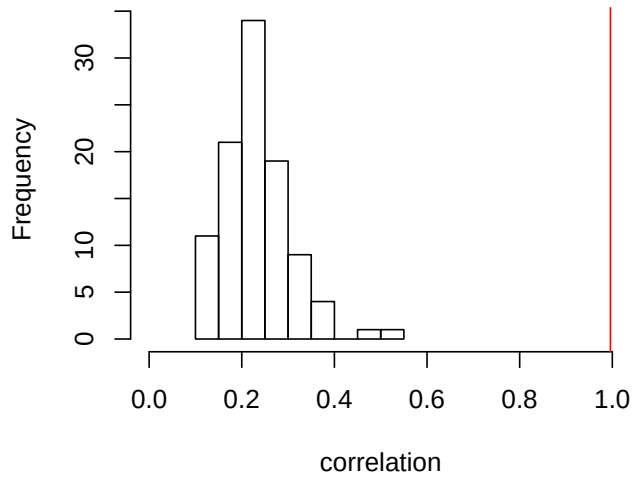
- Determine $\theta = \mathrm{Cor}(E[Y \mid X], Y)$.
- Generate n random permutations $X^{(i)}$ of X.
- Determine $\theta^{(i)} = \mathrm{Cor}(E[Y \mid X^{(i)}], Y)$
- Determine

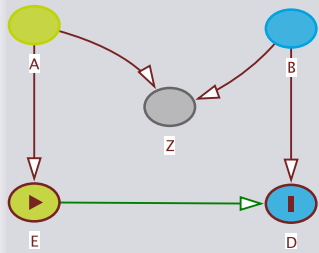
$$p = \frac{\{i : |\theta^{(i)}| > |\theta|\}}{n}$$

```
replicate(100,cor(
 predict(lm(y~sample(x))),y))
```




```
replicate(100,cor(
 predict(loess(y~sample(x))),y))
```





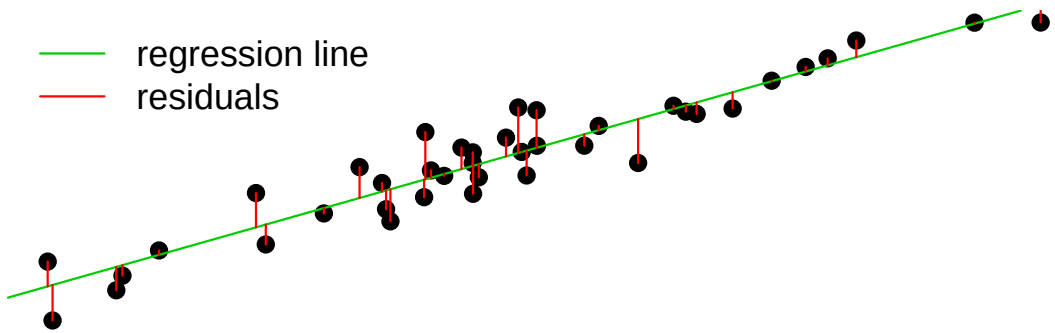
Conditional Independence for Linear Relations

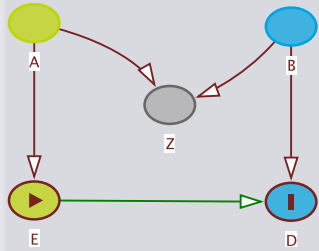
Let's now consider conditional independence again – i.e., $\mathbf{Z} \neq \emptyset$. For multivariate normal data, we can run the regression

$$Y \sim X + \mathbf{Z}$$

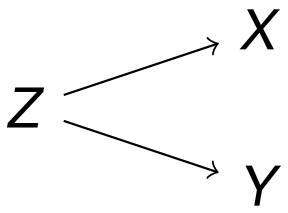
and examine the coefficient of X. An equivalent way to phrase this is by using residuals:

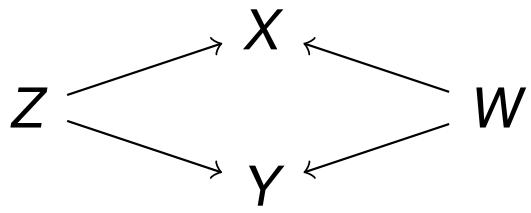
- Regress X on $\mathbf{Z}$ and determine the residuals $r_X = X - E[X \mid \mathbf{Z}]$.
- Regress Y on $\mathbf{Z}$ and determine the residuals $r_Y = Y - E[Y \mid \mathbf{Z}]$.
- If r_X and r_Y correlate, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ does not hold.

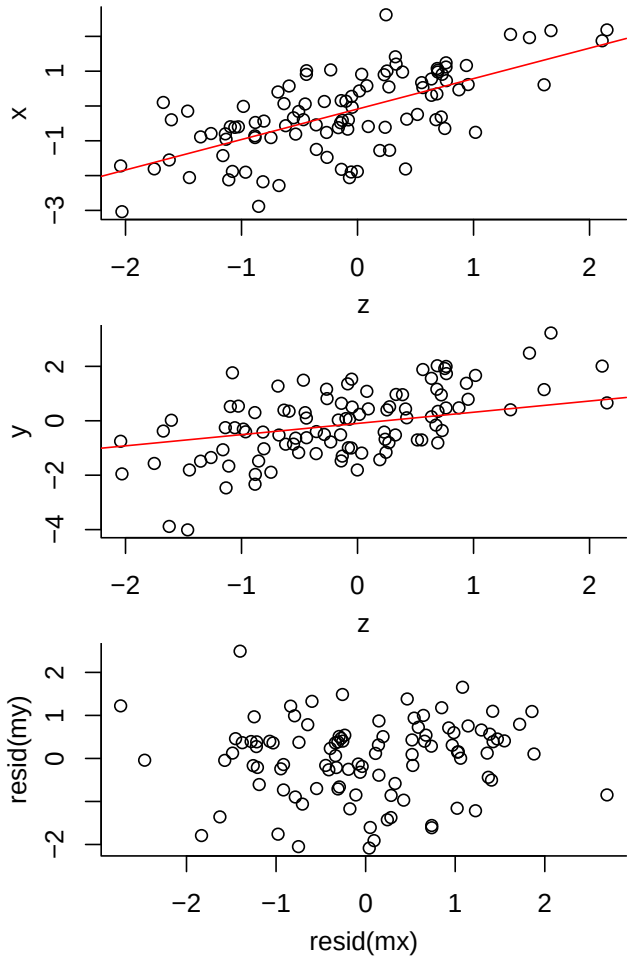


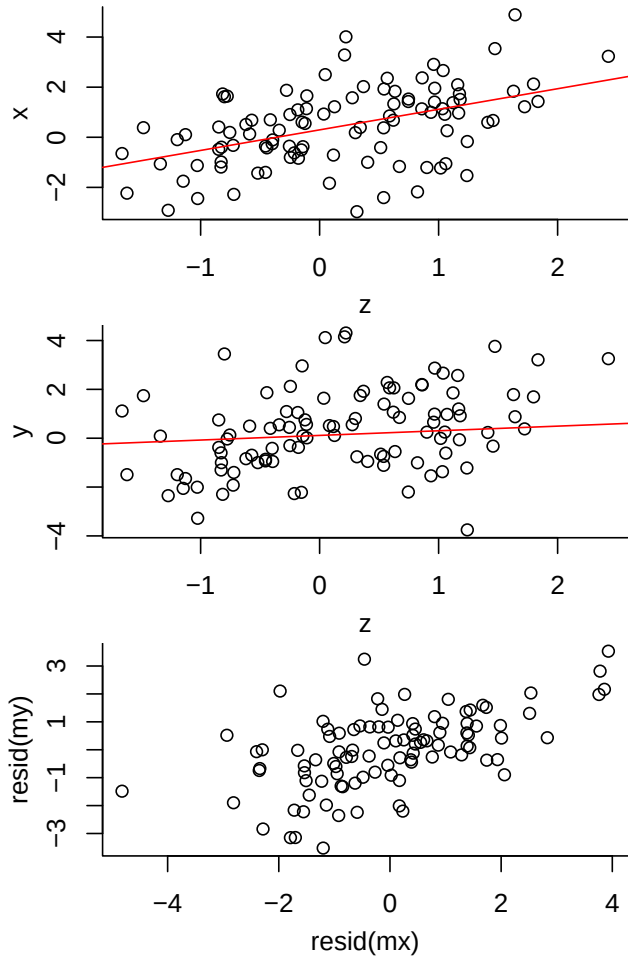


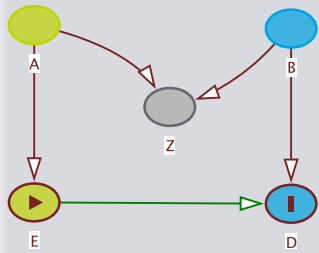
Example: Testing $X \perp\!\!\!\perp Y \mid Z$







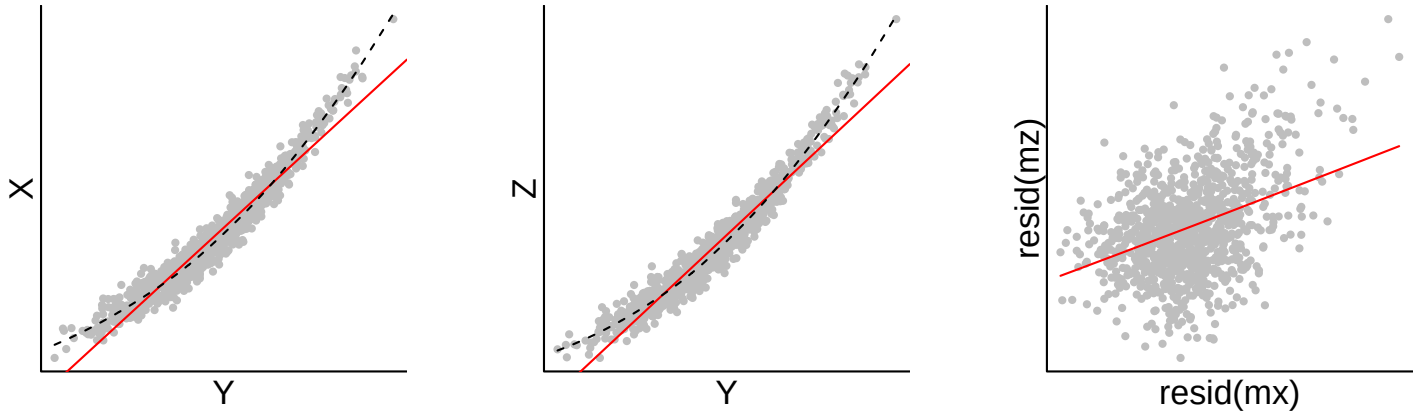




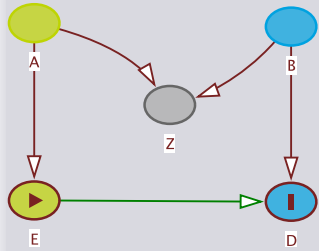
Residual Correlatedness Can Be Misleading

In the simple case $\mathbf{Z} = \emptyset$, linear regression testing was prone to false negatives (concluding independence for dependent variables), but not to false positives (concluding dependence for independent variables).
Unfortunately, this is not true for $\mathbf{Z} \neq \emptyset$. Below we give an example for a nonlinear model

$$X \leftarrow Y \rightarrow Z \,.$$

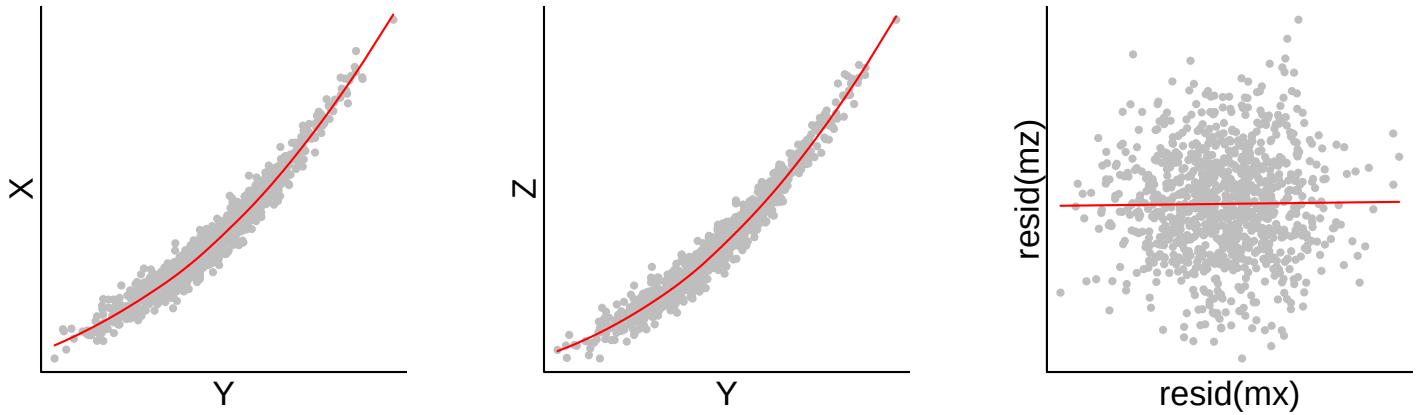


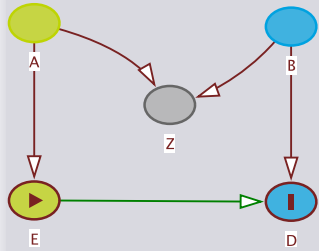
The residuals appear correlated because linear regression does not fit the $X \sim Y$ and $Z \sim Y$ relationships. But X and Z are nevertheless conditionally independent.



Free-Form Regression Residuals

Again, we can attempt to model nonlinear $X \sim Y$ and $Z \sim Y$ relationships using free-form regression. This approach is less prone to false positives if there is a clear nonlinear pattern.





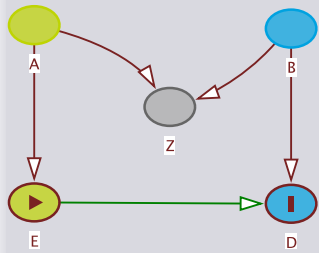
Recap: The IC Algorithm

We recapitulate the IC algorithm from last week:

The IC Algorithm

- Start with a graph containing no edges.
- For each pair of variables X and Y, search for a set $\mathbf{Z}_{XY}$ such that the independence $X \perp\!\!\!\perp Y \mid \mathbf{Z}_{XY}$ is either in the input list, or follows from those in the input list. If no such set exists, link X and Y by an undirected edge $X - Y$.
- For each pair of variables X and Y that are not linked, but have a common neighbour W ($X - W - Y$), check whether $W \in \mathbf{Z}_{XY}$. If not, then add arrowheads pointing to W, i.e. ($X \rightarrow W \leftarrow Y$).
- Orient the resulting graph pattern into a CPDAG.

Given what we now know about conditional independence, how would be go about finding the set $\mathbf{Z}_{XY}$?



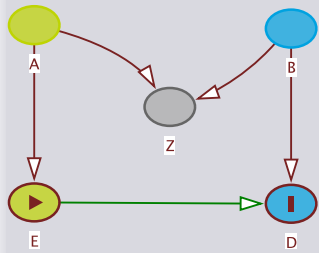
Modification 1

We begin by rephrasing the algorithm slightly.

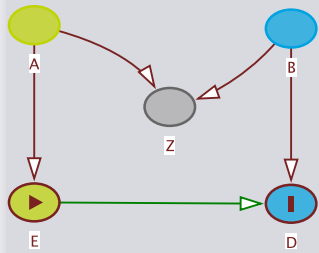
The IC Algorithm

- Start with a graph containing edges between all pairs of variables.
- For each pair of variables X and Y, search for a set $\mathbf{Z}_{XY}$ such that the independence $X \perp\!\!\!\perp Y \mid \mathbf{Z}_{XY}$ is either in the input list, or follows from those in the input list. If any such set exists, remove the edge between X and Y.
- For each pair of variables X and Y that are not linked, but have a common neighbour W ($X - W - Y$), check whether $W \in \mathbf{Z}_{XY}$. If not, then add arrowheads pointing to W, i.e. ($X \rightarrow W \leftarrow Y$).
- Orient the resulting graph pattern into a CPDAG.

We shall see in a minute why this is better.



Example: The Sachs et al. Data



RESEARCH ARTICLE

Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data

Karen Sachs[1,*], Omar Perez[2,*], Dana Pe'er[3,*], Douglas A. Lauffenburger[1,†], Garry P. Nolan[2,†]

+ Author Affiliations

↵† To whom correspondence should be addressed. E-mail: lauffen@mit.edu (D.A.L.); gnolan@stanford.edu (G.P.N.)

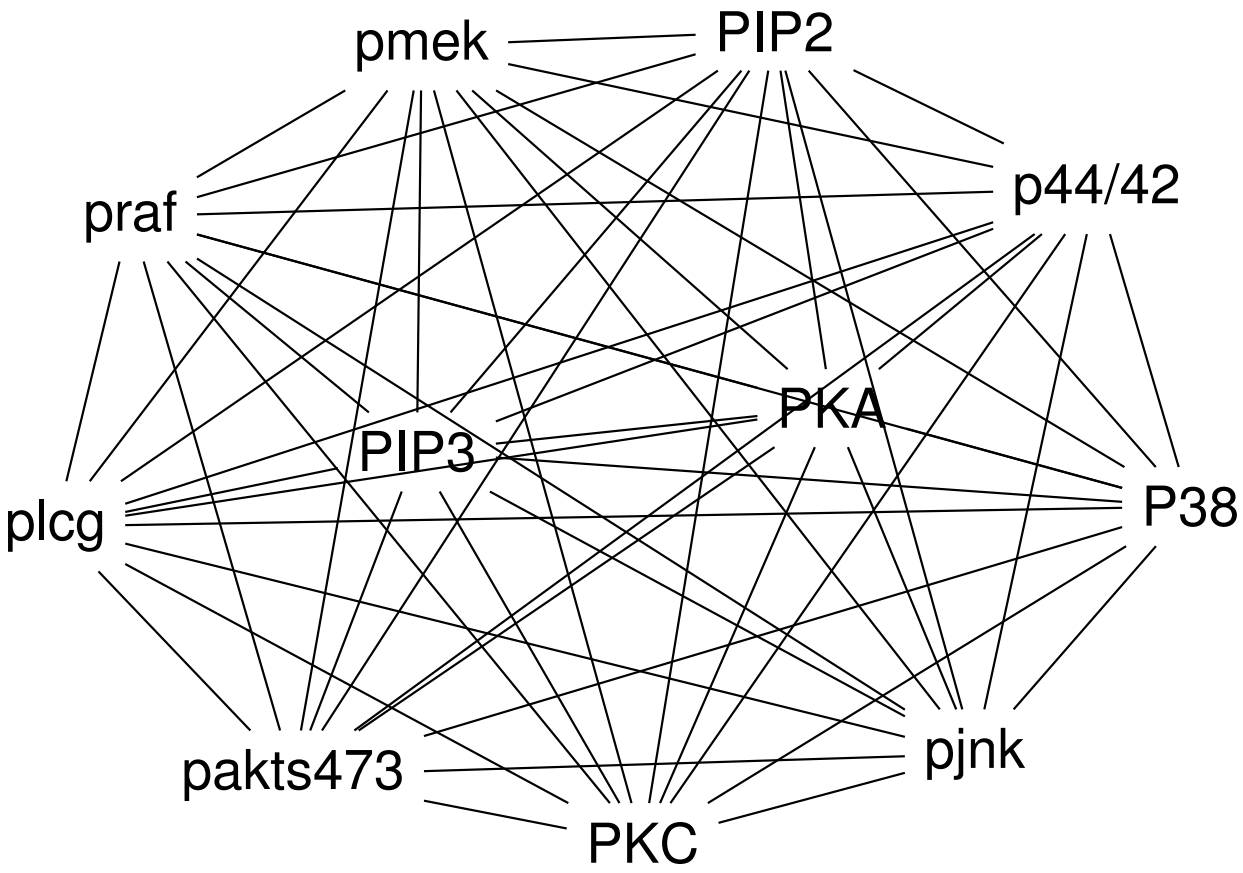
↵* These authors contributed equally to this work.

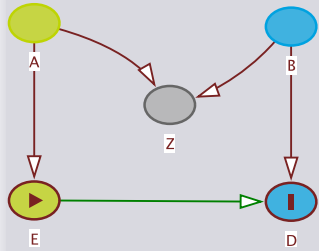
Science 22 Apr 2005:
Vol. 308, Issue 5721, pp. 523-529
DOI: 10.1126/science.1105809

Sachs et al. were interested in the interactions between a set of 11 proteins. They used a Bayesian network learning algorithm for this purpose.

Inference Algorithm: Step 1

We begin with a network containing all possible links between variables.





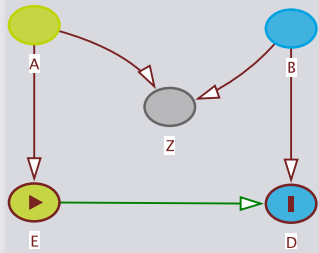
Keeping S Small

Now we need examine sets $\mathbf{Z}_{XY}$ for each pair of X and Y. There are 11 variables, so $2^9 = 512$ sets can be examined.
Insight: A smaller $\mathbf{Z}_{XY}$ is better than a larger $\mathbf{Z}_{XY}$, and an empty $\mathbf{Z}_{XY}$ is best.

Thus, we first check which variables are independent. We do this by applying a cutoff of 0.05.

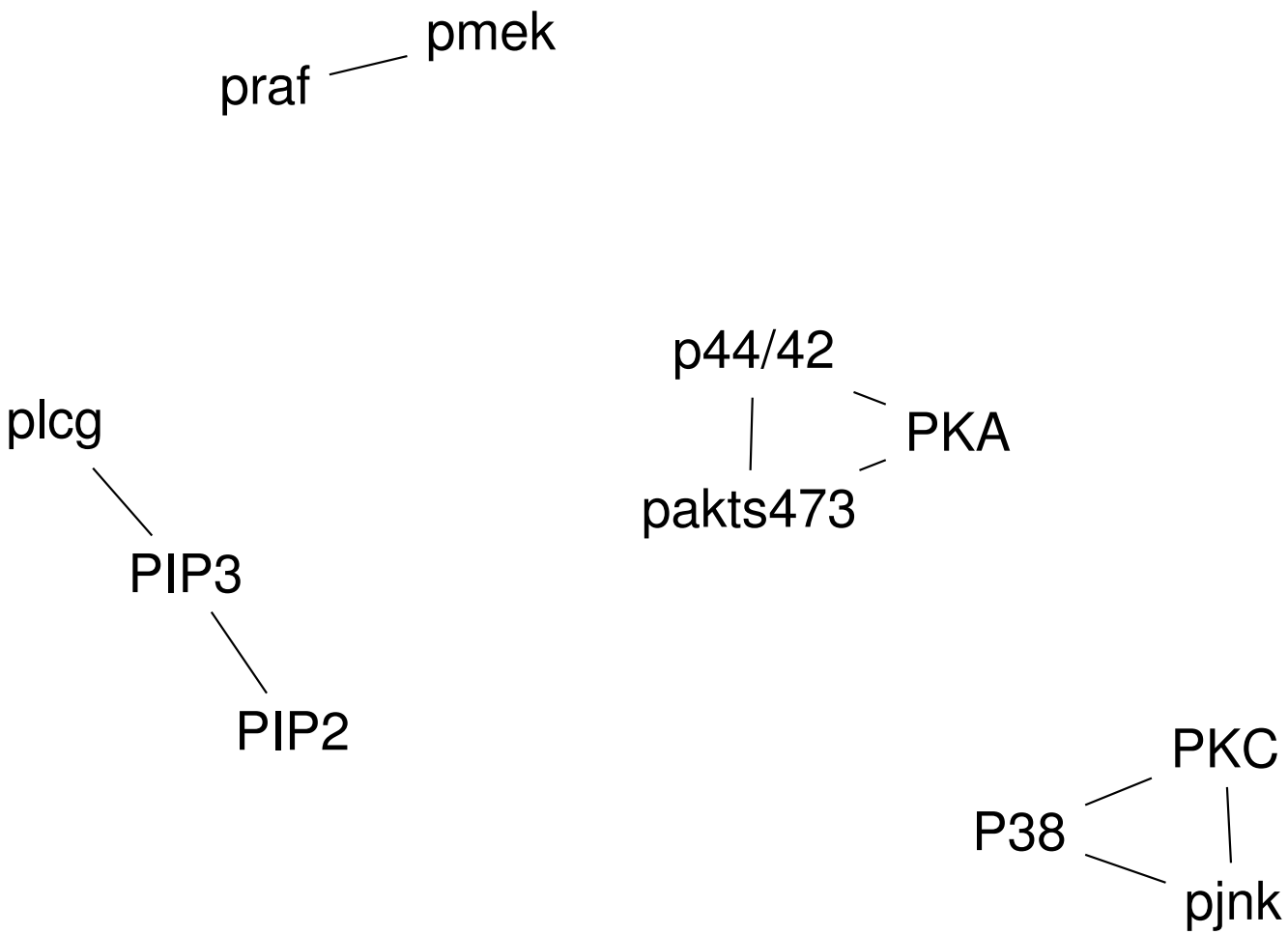
```
library( Hmisc )
Mp <- rcorr( d )$P
as.integer(Mp < 0.05)
```

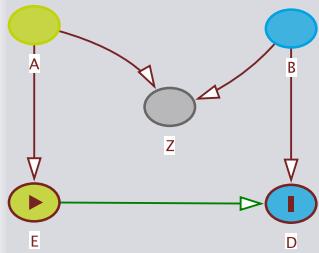
```
##           praf pmek plcg PIP2 PIP3 p44/42 pakts473 PKA PKC P38 pjnk
## praf        NA    1    0    0    0       0        0   0   0   0    0
## pmek         1   NA    0    0    0       0        0   0   0   0    0
## plcg         0    0   NA    0    1       0        0   0   0   0    0
## PIP2         0    0    0   NA    1       0        0   0   0   0    0
## PIP3         0    0    1    1   NA       0        0   0   0   0    0
## p44/42       0    0    0    0    0      NA        1   1   0   0    0
## pakts473     0    0    0    0    0       1       NA   1   0   0    0
## PKA          0    0    0    0    0       1        1  NA   0   0    0
## PKC          0    0    0    0    0       0        0   0  NA   1    1
## P38          0    0    0    0    0       0        0   0   1  NA    1
## pjnk         0    0    0    0    0       0        0   0   1   1   NA
```

Inference Algorithm: After Step 1

We have already reduced our graph a lot:

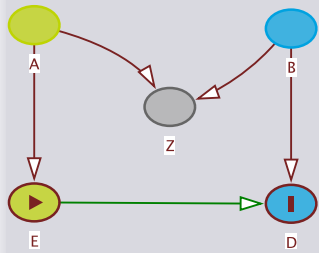




5-Minute Exercise

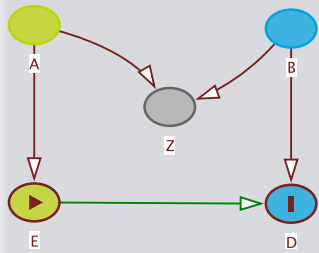
Exercise

For those variable pairs X, Y that are still linked by an edge, we now need to continue searching for a separator $\mathbf{Z}_{XY}$.
Do we still need to search through all possible sets of variables when searching for separators, or can we ignore some of these sets to speed up the search?



Summary

1. Testing conditional independence is difficult, and requires proper modelling of the relationships between variables.
2. It is preferable to perform tests for small or empty **Z** whenever possible.
3. Most algorithms for structure learning are aware of this and use tricks to avoid tests with large **Z**.
4. The IC algorithm plus such tricks is known as the PC algorithm.

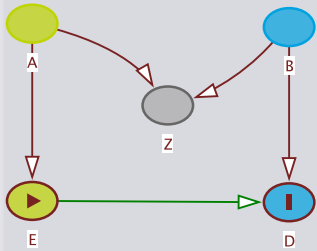


Lecture 7

Managing Network Complexity

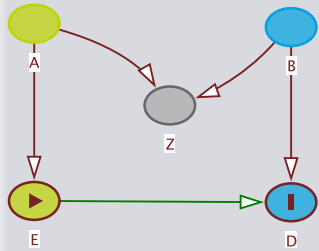
Course Bayesian Networks, November 19th, 2018

Johannes Textor
Institute for Computing and Information Sciences



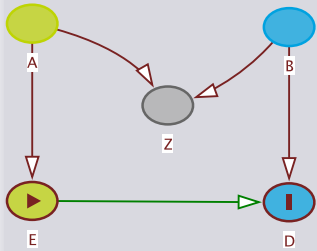
Learning Objectives 7

1. Understand how Bayesian networks represent data-generating processes.
2. Keep Bayesian networks manageable.
3. Be able to interpret and fit parametric Bayesian networks.



Lecture Outline 7

1. Recap: Interpretation of Bayesian Networks
2. Manageable Networks: Tricks of the Trade
3. Parametric Representations

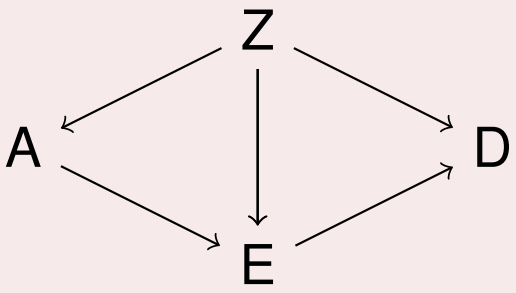


Bayesian Networks as Factorizations

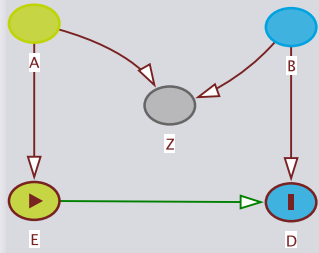
A Bayes net is a graphical representation of a joint probability density as a factorization into conditional PDFs.

Example

Consider the probability density $P(A, Z, D, E)$ with the following Bayes net:



$$
\begin{aligned}
& P(A, Z, D, E) \\
=& P(Z)\, P(A \mid Z) \\
& P(E \mid A, Z)\, P(D \mid E, Z)
\end{aligned}
$$

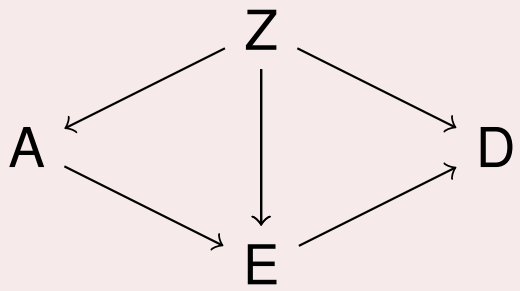


Bayesian Networks as Independence Maps

Instead of a factorization, we can also characterize the compatible densities by a basis set of conditional independencies.
For instance, the following set of statements is a basis set:

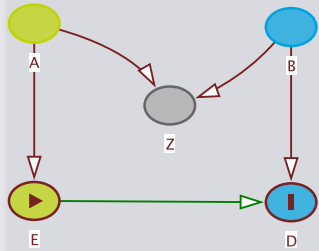
$$X \perp\!\!\!\perp \text{non-descendants}(X) \mid \text{parents}(X)$$

Example



$$D \perp\!\!\!\perp A \mid E, Z$$

(D is the only node with non-descendants that are not parents)



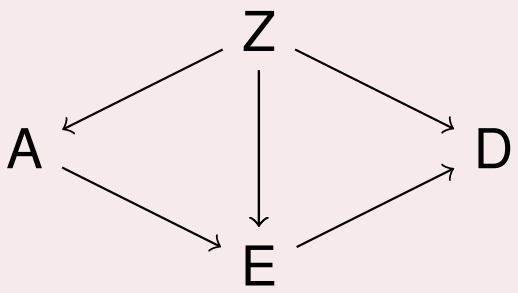
Structural Models

The last equivalent view of a Bayes net is as a layout of a data-generating process – a structural model.
A structural model consists of a graph $G = (\mathbf{V}, \mathbf{E})$ and a set of functions $\{f_X \mid x \in \mathbf{V}\}$ such that, for each variable X:

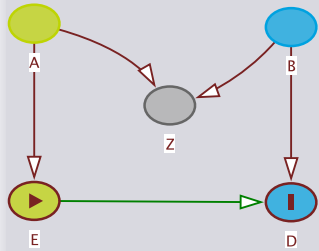
$$X := f_X(\mathrm{pa}_X, \epsilon_X)$$

- f_X is any deterministic function.
- pa_X is the set of all parents of X in G.
- ϵ_X is a random variable.
- All ϵ_X are mutually independent.

Example



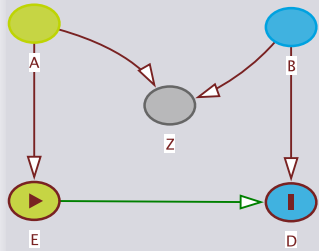
$$\begin{aligned} Z &:= f_Z(\epsilon_Z) \\ E &:= f_E(A, Z, \epsilon_E) \\ A &:= f_A(Z, \epsilon_A) \\ D &:= f_D(Z, E, \epsilon_D) \end{aligned}$$



Bayesan Networks Aren't Always Efficient

We motivated Bayesian networks as a way to store a probability density P as many small tables instead of one big table. But we may still encounter scalability issues.

- "Rich" discrete values imply large probability tables: $P(\text{Salary} \mid \text{Age}, \text{Gender})$
- Continuous values cannot be stored in tables at all.
- Tables can still be large for nodes with many parents.



Binning

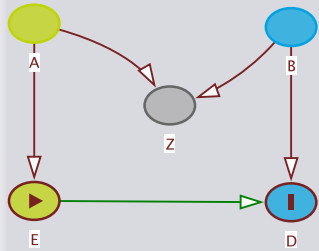
Continuous or “rich” discrete variables are often binned into ordinal variables.

- Age $\mapsto \{\text{child}, \text{adolescent}, \text{adult}, \text{middle age}, \text{old age}\}$

age	category
0-10	child
11-20	adolescent
21-40	adult
41-60	middle age
>60	old age

- Temperature $\mapsto \{\text{cold}, \text{mild}, \text{hot}\}$

Binning avoids having to estimate too many parameters, especially if some combinations of values are not observed in the data (zero probabilities).

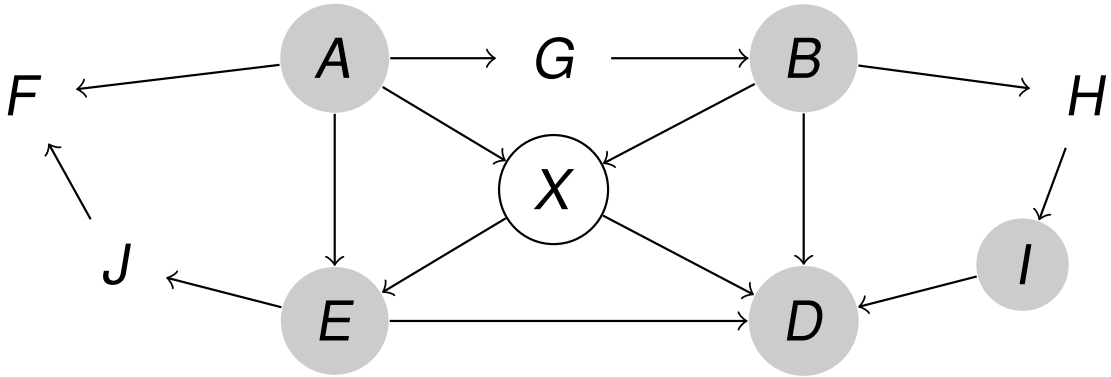


Pruning: The Markov Blanket

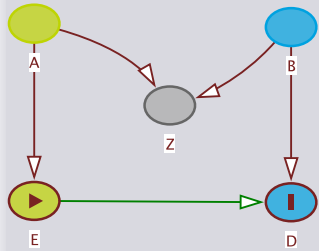
For some use cases of a Bayes net, such as predicting the value of a specific node from the other nodes, it may turn out that not all nodes are actually necessary.

Definition

The Markov blanket ∂X of a node $X \in \mathbf{V}$ in a DAG $G = (\mathbf{V}, \mathbf{E})$ contains the parents of X, the children of X, and the children's other parents.



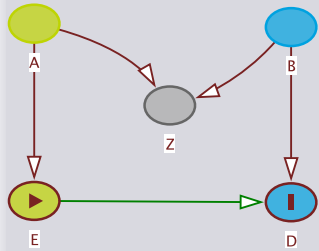
Nodes that are not in the Markov blanket of X can be ignored for predicting X.



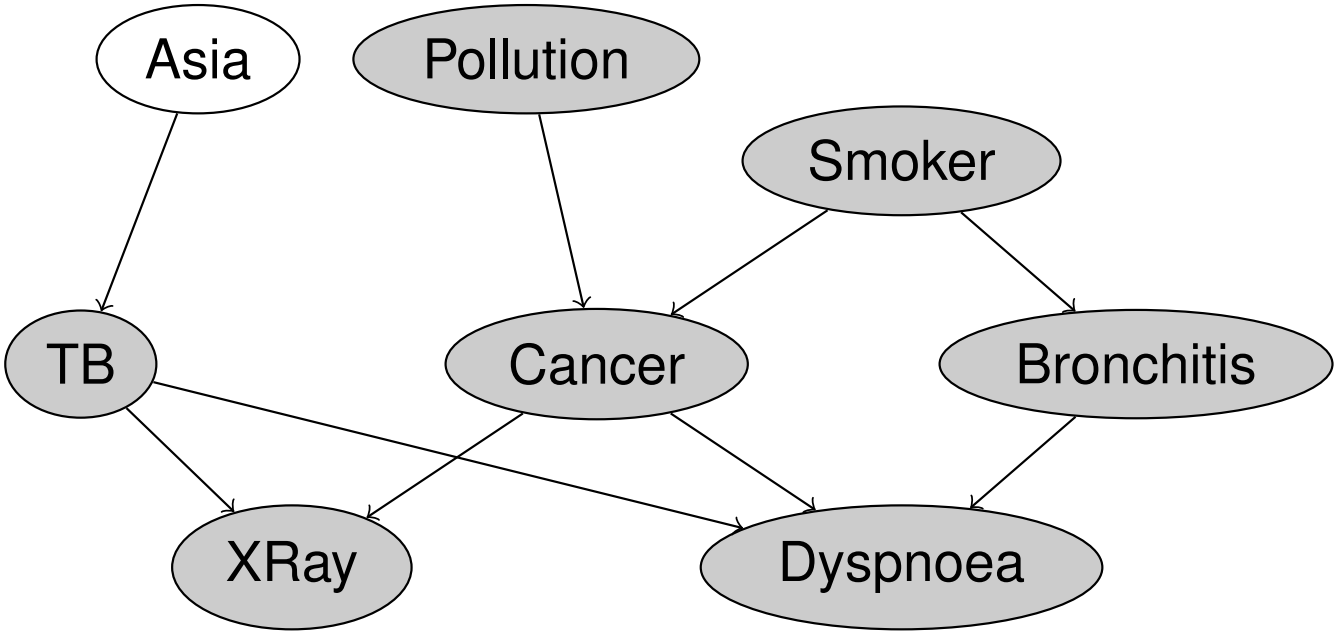
5-Minute Exercise

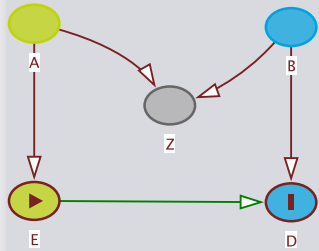
Exercise

Prove that the Markov blanket ∂X d-separates X from all other nodes in G!



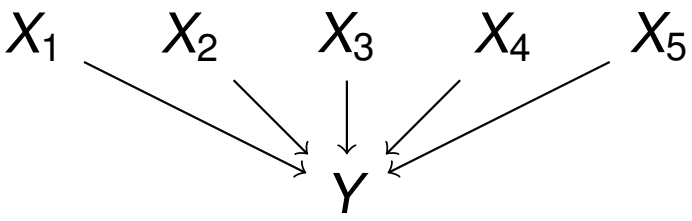
Markov Blanket: Example





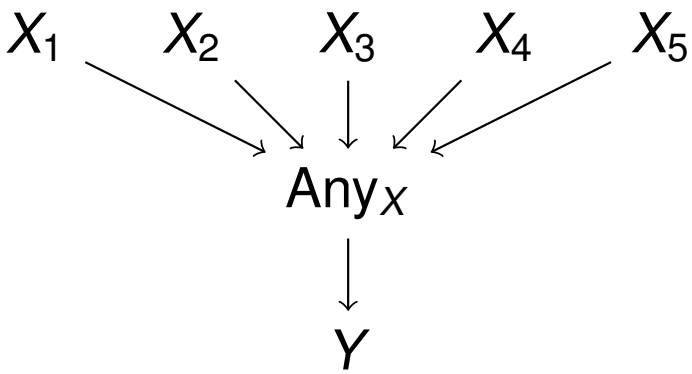
Divorcing Too Many Parents

Networks in which some nodes have too many parents frequently pose problems.
For example, 2^n parameters are necessary to represent a conditional density $P(y \mid x_1, \ldots, x_n)$ where all variables are binary.

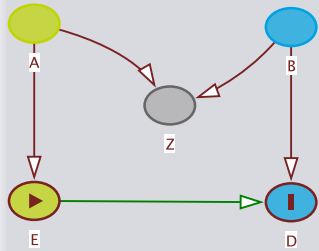


$5 + 2^5 = 37$ parameters

By divorcing, one creates a new node that is a deterministic combination of some other nodes (often, a logical disjunction). All of the combined nodes' former children now depend on this single node.

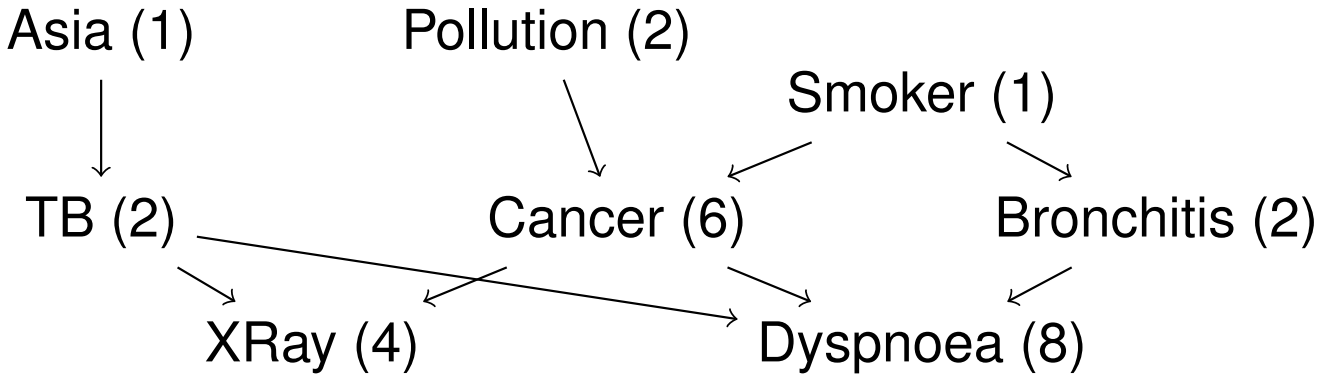


$5 + 2^1 = 7$ parameters

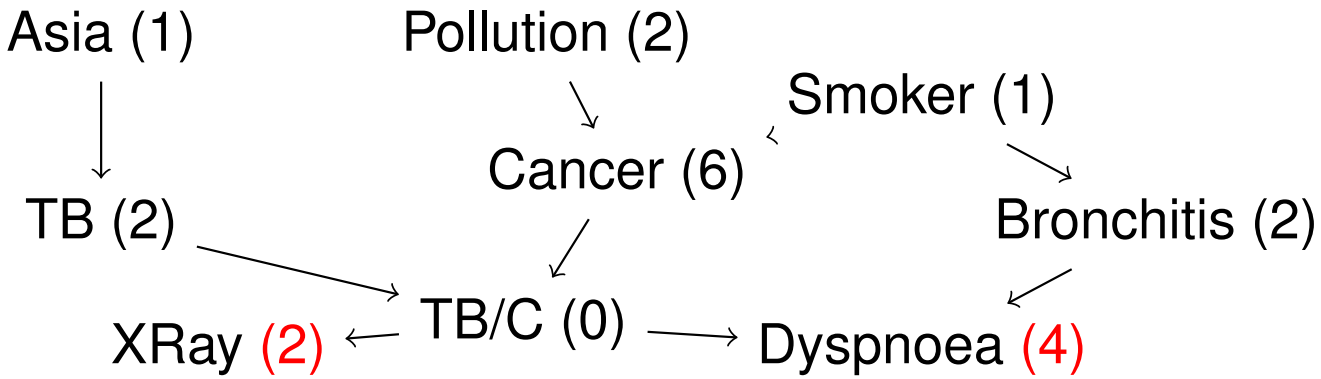


Divorcing: Example

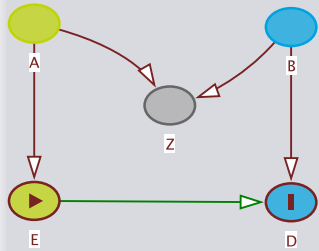
Let's revisit the "asia" network, where all variables are binary, except for "Pollution", which is ternary. This network requires 26 unique parameters.



Divorcing "TB" and Cancer" reduces this to 20 parameters.



Node "TB/C" is the disjunction of "TB" and "Cancer".



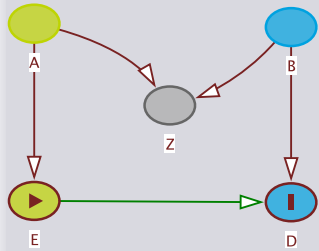
Issues With Binning and Divorcing

So far, we have annotated the nodes of Bayesian networks with simple probability tables. To reduce the size of such tables, we have introduced binning and divorcing. But these approaches bring new problems:

- Binning loses information, and relies on (often arbitrary) cut-offs.
- Divorcing loses flexibility and induces new independencies.

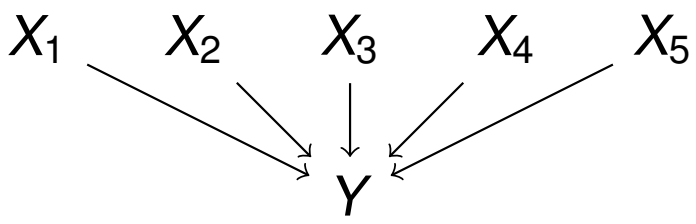
A further approach is to replace probability tables by parametric probability distributions (e.g., ninomial, negative binomial, geometric, hypergeometric, Poisson, ...)

Such parametric networks are often called directed probabilistic graphical models rather than Bayes nets.



Example: The Logit Model

For n parent variables,



$5 + 2^5$ parameters

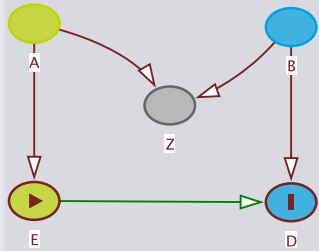
For $0 < p < 1$, define

$$\text{logit(p)} = \log \frac{p}{1-p}$$

Then

$$\log \frac{P(Y_1 = 1 \mid x_1, \ldots, x_n)}{1 - P(Y_1 = 1 \mid x_1, \ldots, x_n)} = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n$$

Note that the X_i do not need to be binary for this model – they can even be continuous!
The logit model is also called the logistic regression model.



Parameter Estimation for Bayesian Networks

Formally, we are replacing the population probability density P by an estimate P_θ with parameters θ. The parameters θ are usually found by maximizing the likelihood for a sample $S = (x_1^{(1)}, \ldots, x_n^{(1)}), \ldots, (x_1^{(k)}, \ldots, x_n^{(k)})$:

$$\mathcal{L}(S \mid \theta) = \prod_i P_\theta(X_1 = x_1^{(i)}, \ldots, X_n = x_n^{(i)})$$

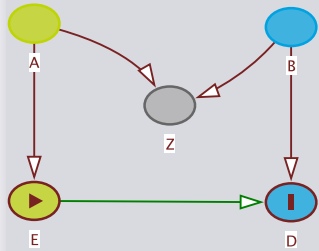
Typically, we instead use the log-likelihood

$$\log \mathcal{L}(S \mid \theta) = \sum_i \log P_\theta(X_1 = x_1^{(i)}, \ldots, X_n = x_n^{(i)})$$

By inserting the factorization, we can express this as

$$\log \mathcal{L}(S \mid \theta) = \sum_j \sum_i \log P_\theta(X_j = x_j^{(i)} \mid \mathrm{pa}_{X_j} = \mathrm{pa}_{x_j}^{(i)})$$

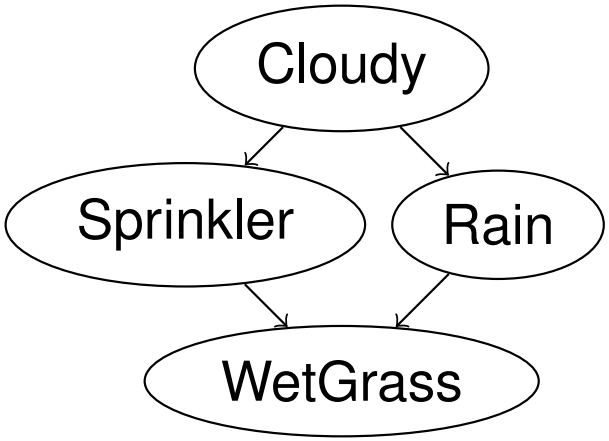
This means we can estimate θ separately for each node.



Example: The Sprinkler Network

We observe that the grass is wet. There are two possible reasons: it could have rained, or the sprinkler could have been turned on. Both possible reasons depend on whether it is cloudy on that day.

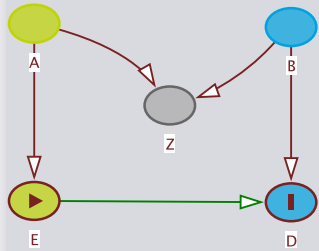
C	P
1	0.5



C	S	P
0	1	0.5
1	1	0.1

C	R	P
0	1	0.2
1	1	0.8

S	R	W	P
0	0	1	0.01
0	1	1	0.9
1	0	1	0.9
1	1	1	0.99

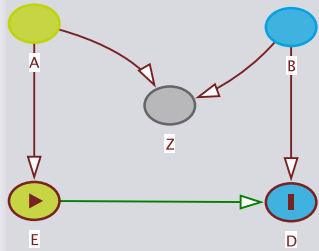


Estimating the Sprinkler Network Parameters

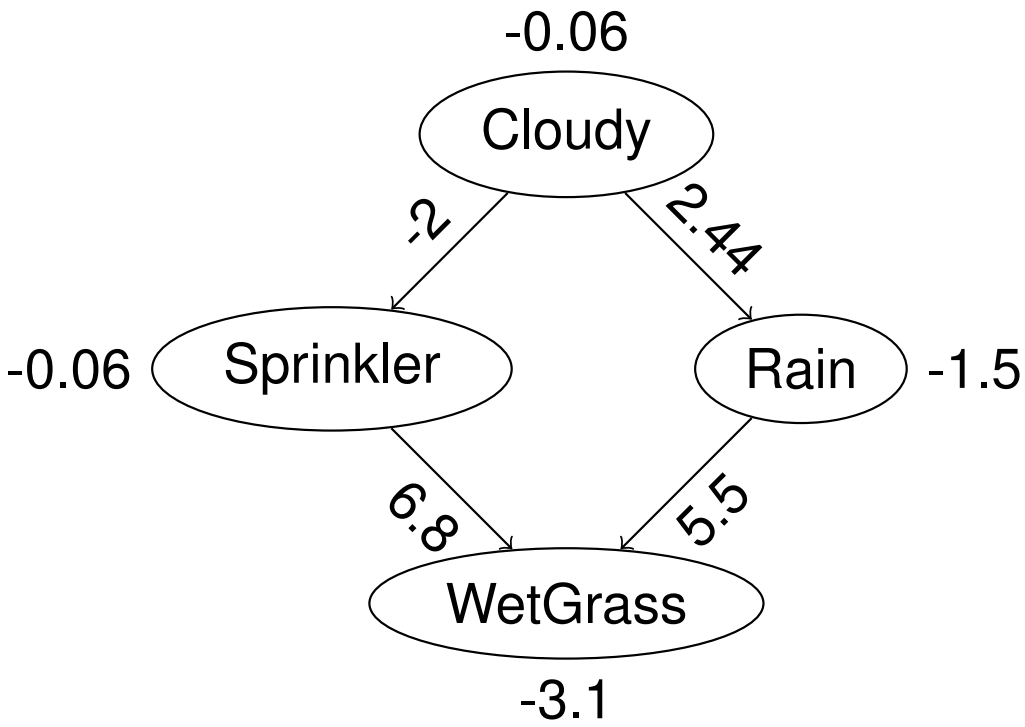
In R, we can find the parameters of a logistic regression model by maximum likelihood as follows:

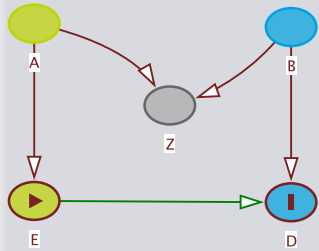
```
d <- read.csv("sprinkler.csv")
m.cloudy <- glm( cloudy~1, d, family="binomial" )
m.sprinkler <- glm( sprinkler~cloudy, d, family="binomial" )
m.rain <- glm( rain~cloudy, d, family="binomial" )
m.wetgrass <- glm( wetgrass~sprinkler+rain, d, family="binomial" )
coef( m.wetgrass )

## (Intercept)   sprinkler        rain
##   -3.076410    6.815945    5.459369
```

For each node, we get one intercept value (=base log odds-ratio) and a coefficient for each parent (=log odds-ratio). The coefficient represents the strength of the influence.



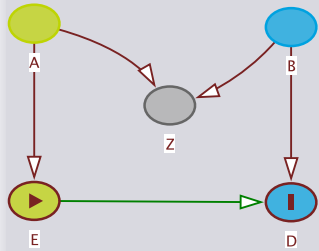
The Fitted Sprinkler Network





Summary

1. We now know three interpretations of Bayesian networks: factorization maps, independence maps, and structural models.
2. These interpretations are different, but equivalent to each other.
3. We learnt some “tricks of the trade” to keep Bayesian network size manageable.
4. Parametric distributions are often used in real-world Bayesian networks instead of raw probability tables.

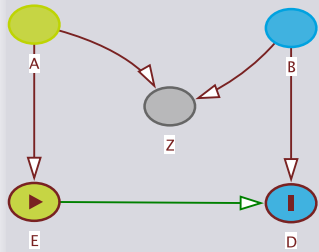


Lecture 8

Interventions and Covariate Adjustment

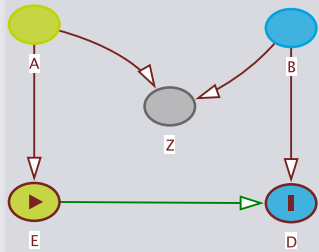
Course Bayesian Networks, November 26th, 2018

Johannes Textor
Institute for Computing and Information Sciences

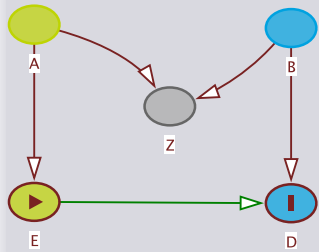


Learning Objectives 8

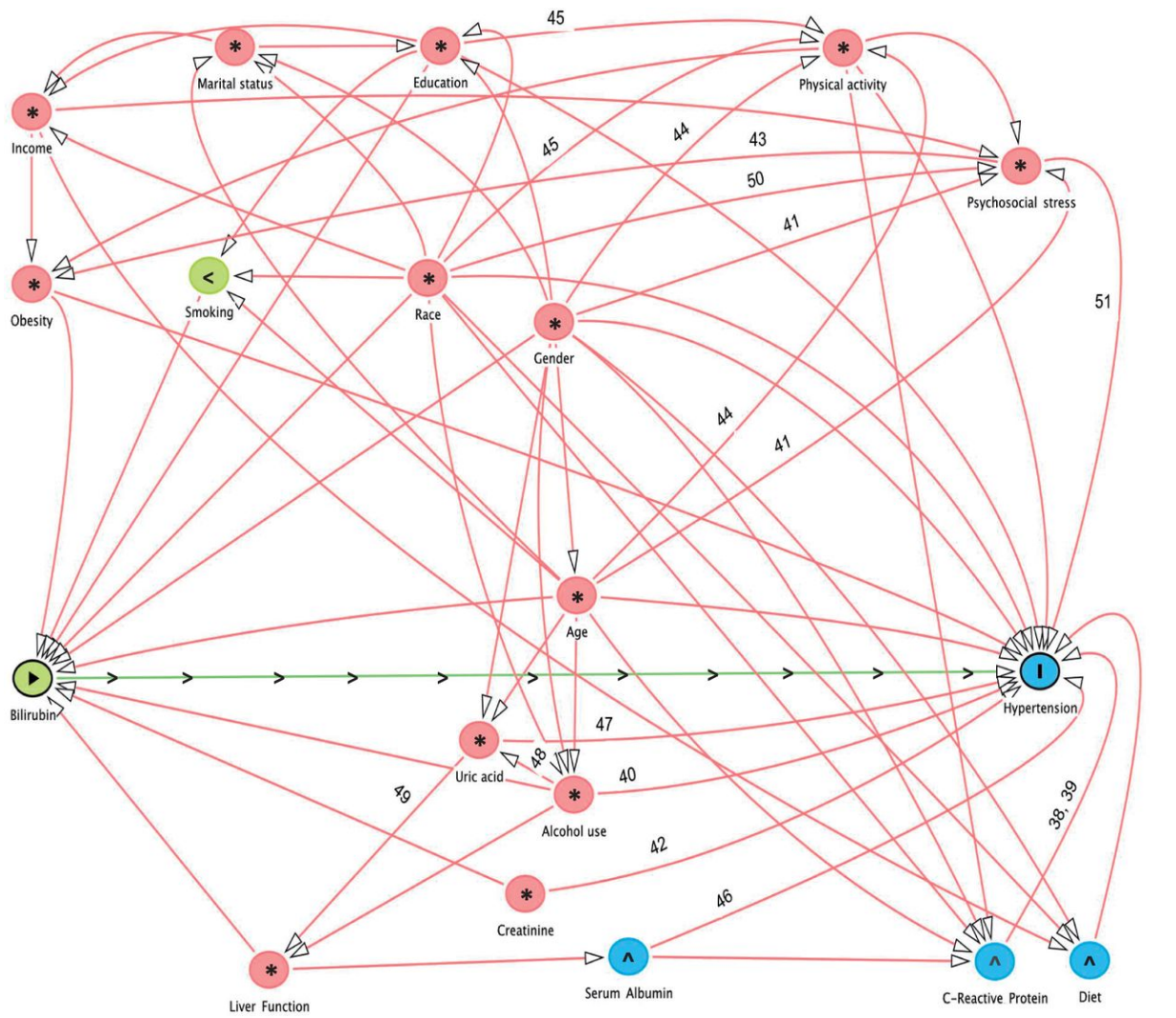
1. Be familiar with the do-operator to define causal effects.
2. Understand how DAGs help to infer causation.
3. Be able to apply the back-door criterion for covariate adjustment.

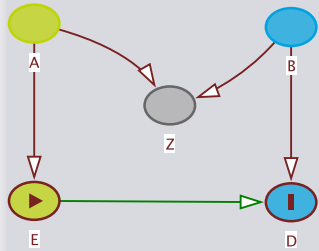


Lecture Outline 8



So What is This About?

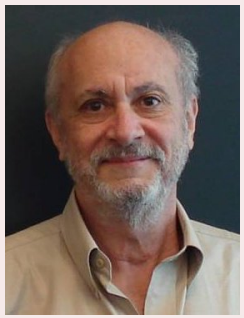
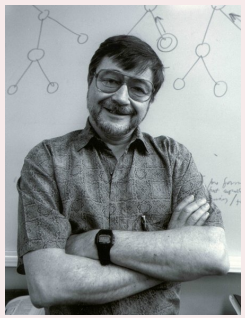


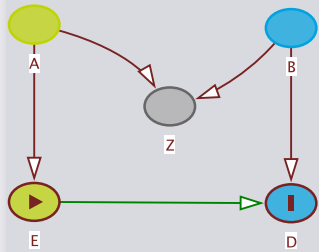


Causality Theory

- Structural causal models provide a language to encode causal relationships – the basis of a causality theory.
- Causality theory help decide when, and how, causation can be inferred from domain knowledge and data.

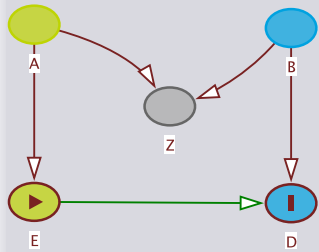
Some people who contributed to causality theories

Donald Rubin	Judea Pearl	Donald Campbell	Phil Dawid	Clive Granger
				



[. . .] all approaches to causation are variants or abstractions of [. . .] structural theory [. . .].

– Judea Pearl, "Causality"

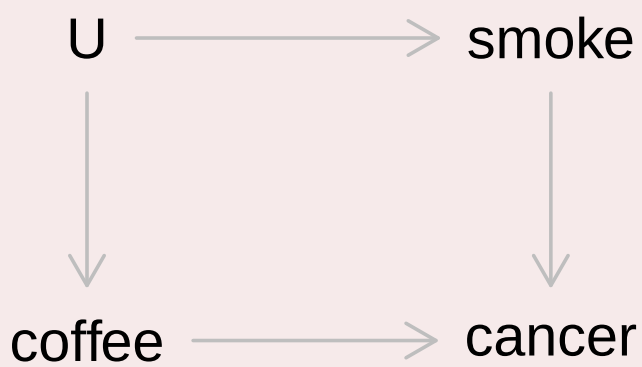


Modelling Interventions: The do-Operator

In Bayesian networks, a (model-implied) causal effect can be defined as the correlation in a modified model where all influences stemming from "natural causes" of the exposure variable are removed.

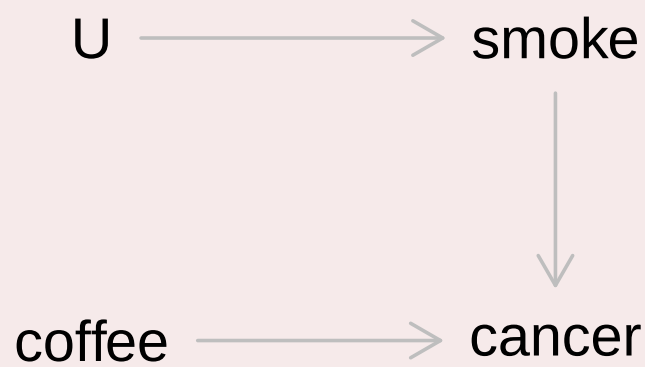
$P(\text{cancer} \mid \text{coffee})$

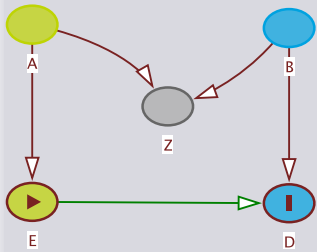
Observational Regime



$P(\text{cancer} \mid \mathbf{do}(\text{coffee}))$

Interventional regime

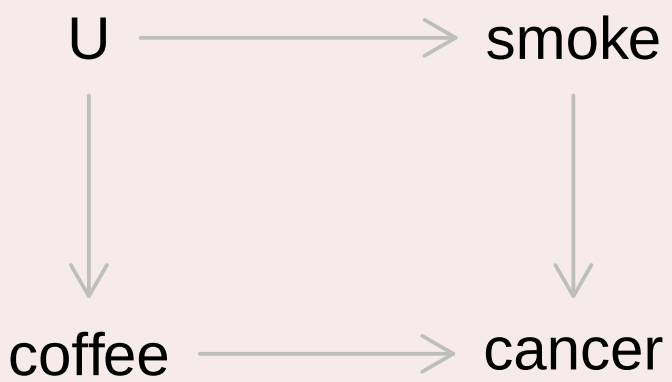




Modelling Interventions: The do-Operator

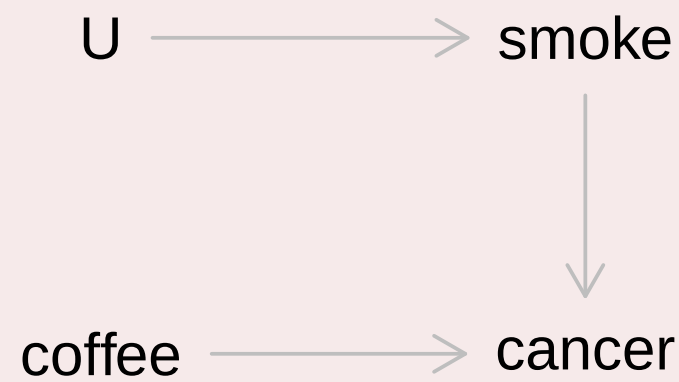
$$P(\mathbf{v}|do(\mathbf{X} = \mathbf{x})) = \prod_{X_i \in \mathbf{V} \setminus \mathbf{X}} P(x_i \mid \mathrm{pa}(x_i))$$

Observational regime

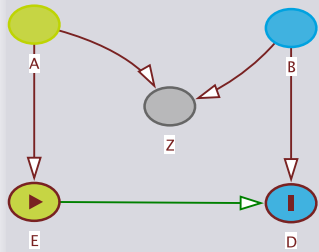


$P(\text{cancer}, \text{coffee}, \text{smoke}, \text{U})$
$= P(\text{cancer} \mid \text{coffee}, \text{smoke})$
$P(\text{U})\ P(\text{smoke} \mid \text{U})\ P(\text{coffee} \mid \text{U})$

Interventional regime



$P(\text{cancer}, \text{smoke}, \text{U} \mid \mathbf{do}(\text{coffee}))$
$= P(\text{cancer} \mid \text{coffee}, \text{smoke})$
$P(\text{U})\ P(\text{smoke} \mid \text{U})$



Back to Simpson's Paradox

Suppose a new treatement for a disease is tested in a trial with the following results:

	Cured	Not Cured
Treated	20	20
Not Treated	16	24

$P(C = 1 \mid T = 1) = 0.5$

$P(C = 1 \mid T = 0) = 0.4$

Now the investigator wants to know whether the treatment is more effective in men or women, and gets the following results:

Males	Cured	Not Cured
Treated	18	12
Not Treated	7	3

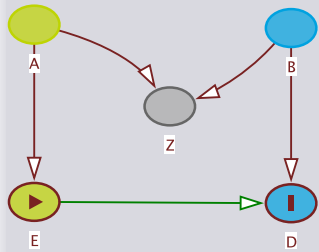
$P(C = 1 \mid T = 1, S = m) = 0.6$
$P(C = 1 \mid T = 0, S = m) = 0.7$

Females	Cured	Not Cured
Treated	2	8
Not Treated	9	21

$P(C = 1 \mid T = 1, S = f) = 0.2$
$P(C = 1 \mid T = 0, S = f) = 0.3$

What does this mean? Do we give the treatment or not?

Judea Pearl:
Simpson's Paradox: An Anatomy
http://bayes.cs.ucla.edu/R264.pdf

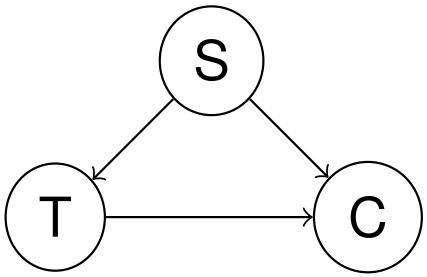


Resolving the Paradox: Step 1

First, we build a Bayesian network that represents the causal structure of the problem as well as the probabilities.

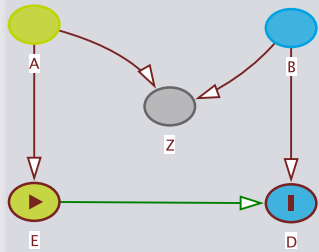
S	P
f	0.5

C	S	P
0	1	0.5
1	1	0.1



S	T	C	P
f	0	1	0.3
f	1	1	0.2
m	0	1	0.7
m	1	1	0.6

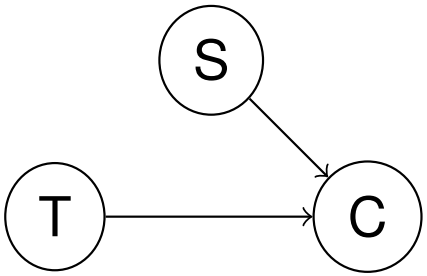
This network describes the observational regime, where we have
$P(C = 1 \mid T = 1) = 0.5$ and $P(C = 1 \mid T = 0) = 0.4$



Resolving the Paradox: Step 2

Now let us modify our Bayesian network to obtain the interventional regime, in which we set T ourselves.

S	P
f	0.5



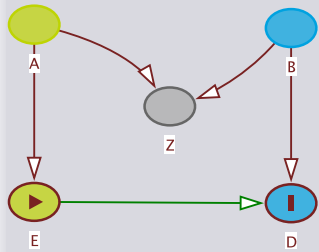
S	T	C	P
f	0	1	0.3
f	1	1	0.2
m	0	1	0.7
m	1	1	0.6

This network describes the interventional distribution

$$P(c, s \mid \mathrm{do}(t) = P(c \mid s, t)P(s)$$

. Note that $\mathrm{do(t)}$ is conditioned on and therefore there are no parameters for t in the network!
Let us compute the causal effect of the treatment:

$$P(C = 1 \mid \mathrm{do}(T = 1))$$



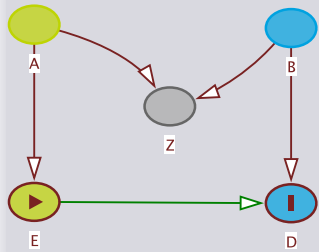
Computing the Treatment Effect

If we do give the treatment:

$$
\begin{aligned}
P(C = 1 \mid \mathrm{do}(T = 1)) &= \sum_s P(C = 1, S = s \mid T = 1) \\
&= \sum_s P(C = 1 \mid S = s, T = 1) P(S = s) \\
&= P(C = 1 \mid S = f, T = 1) P(S = f) + \\
&\quad P(C = 1 \mid S = m, T = 1) P(S = m) \\
&= 0.2 \cdot 0.5 + 0.6 \cdot 0.5 = 0.4
\end{aligned}
$$

If we don't give the treatment:

$$
\begin{aligned}
P(C = 1 \mid \mathrm{do}(T = 0)) &= \sum_s P(C = 1, S = s \mid T = 0) \\
&= \sum_s P(C = 1 \mid S = s, T = 0) P(S = s) \\
&= P(C = 1 \mid S = f, T = 0) P(S = f) + \\
&\quad P(C = 1 \mid S = m, T = 1) P(S = m) \\
&= 0.3 \cdot 0.5 + 0.7 \cdot 0.5 = 0.5
\end{aligned}
$$



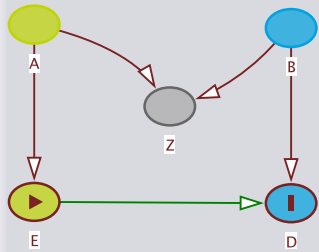
5-Minute Exercise

Exercise

Look at these four probabilities:

- $P(C = 1 \mid T = 0) = 0.4$
- $P(C = 1 \mid T = 1) = 0.5$
- $P(C = 1 \mid \mathrm{do}(T = 0)) = 0.5$
- $P(C = 1 \mid \mathrm{do}(T = 1)) = 0.4$

What do they tell you about the effect of the treatment and the observational data?



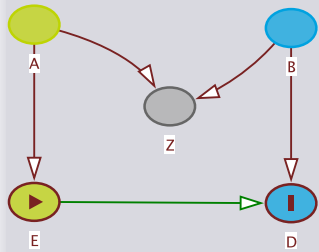
Covariate Adjustment

Take a close look again at how we determined the effect of the treatment:

$$
\begin{aligned}
P(C = 1 \mid \mathrm{do}(T = 1)) &= \sum_s P(C = 1, S = s \mid T = 1) \\
&= \sum_s P(C = 1 \mid S = s, T = 1) P(S = s) \\
&= P(C = 1 \mid S = f, T = 1) P(S = f) + \\
&\quad P(C = 1 \mid S = m, T = 1) P(S = m) \\
&= 0.2 \cdot 0.5 + 0.6 \cdot 0.5 = 0.4
\end{aligned}
$$

We in fact obtained the causal effect by averaging the observational estimates for men and women!

Taking such averages is one form of covariate adjustment.



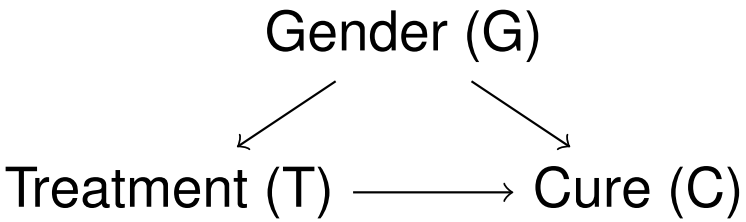
Resolving the Paradox

The Paradox arises due to a mismatch between seeing (the result of the trial) and doing (actually giving the treatment).

We observe $P(c \mid t)$ but we need $P(c \mid \mathrm{do}(t))$.

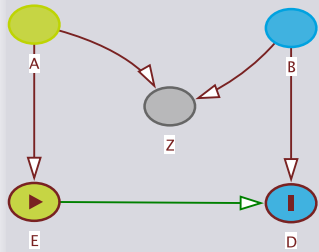
In our data we see that

- males were more likely to take the treatment, and
- males have a higher recovery rate:



$$
\begin{aligned}
P(c,g,t) &= P(g)P(c \mid g,t)P(t \mid g) \\
P(c,g \mid \mathrm{do}(t)) &= P(g)P(c \mid g,t) \\
P(c \mid \mathrm{do}(t)) &= \sum_g P(g)P(c \mid g,t)
\end{aligned}
$$

The causal effect $P(c \mid \mathrm{do}(t))$ is the average of the males and female effects, and we should not give the treatment.



Covariate Adjustment

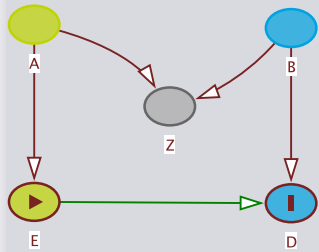
- We often work with observational data (cohort studies, case-control studies, . . .).
- Causal inference from observational data requires making assumptions about the underlying process.

  No causes in – no causes out

  (Nancy Cartwright, "Hunting Causes and Using Them")
- DAGs allow to encode those assumptions in a principled fashion and derive their implications.

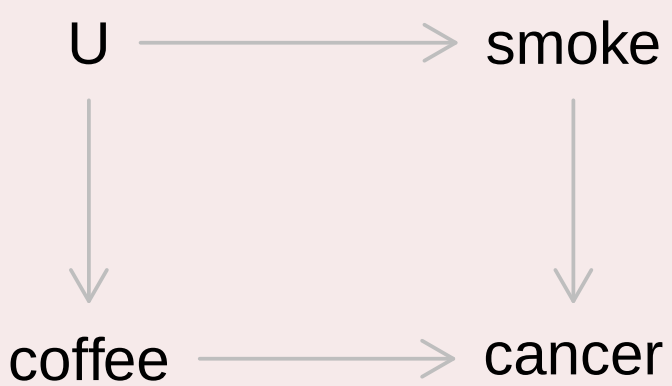
The structural approach to causality

Given this DAG, is the causal effect of X on Y identifiable and if so, how?



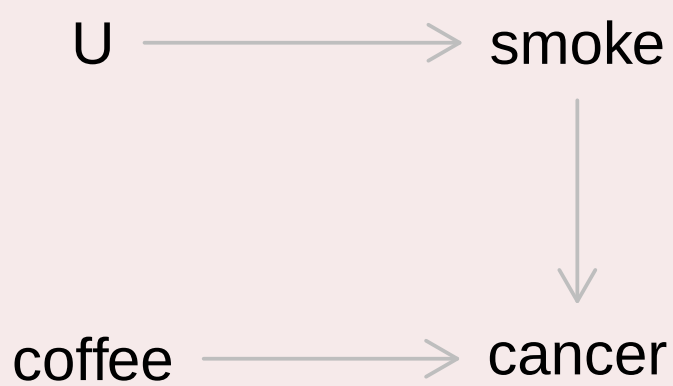
Modelling Interventions: The do-Operator

Observational regime



$P(\text{cancer}, \text{coffee}, \text{smoke}, \text{U})$
$= P(\text{cancer} \mid \text{coffee}, \text{smoke})$
$P(\text{U})\ P(\text{smoke} \mid \text{U})\ P(\text{coffee} \mid \text{U})$

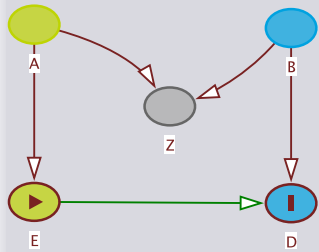
Interventional regime



$P(\text{cancer}, \text{smoke}, \text{U} \mid \mathbf{do}(\text{coffee}))$
$= P(\text{cancer} \mid \text{coffee}, \text{smoke})$
$P(\text{U})\ P(\text{smoke} \mid \text{U})$

These equations imply that:

$$P(\text{cancer} \mid \mathbf{do}(\text{coffee})) = \sum_{\text{smoke}} P(\text{cancer} \mid \text{smoke}, \text{coffee}) P(\text{smoke})$$



Covariate Adjustment

We just derived the covariate adjustment formula:

$$P(\text{cancer} \mid \mathbf{do}(\text{coffee})) = \sum_{\text{smoke}} P(\text{cancer} \mid \text{smoke}, \text{coffee}) P(\text{smoke})$$

To estimate the causal effect of coffee drinking on cancer, we adjust for the confounder "smoking".

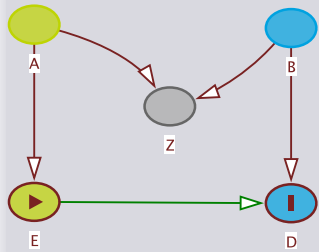
Back-Door Criterion (Pearl, 2009)

Given a dataset that has been generated by a structural causal model G, let $\mathbf{Z}$ be a set of covariates such that

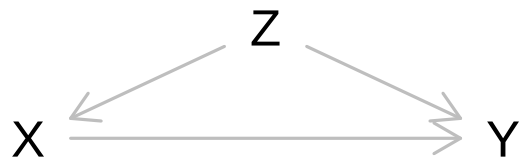
- $\mathbf{Z}$ does not contain any descendants of X;
- $\mathbf{Z}$ d-separates all paths from X to Y starting with $X \leftarrow$.

Then $\mathbf{Z}$ is an adjustment set for the causal effect X on Y, and

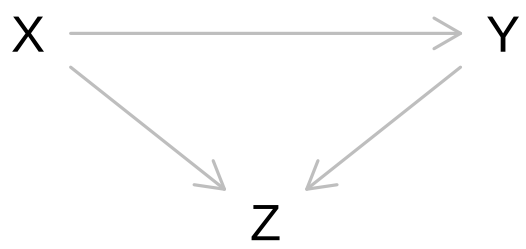
$$P(y \mid do(x)) = \sum_{\mathbf{z}} P(y \mid x, \mathbf{z}) P(\mathbf{z})$$



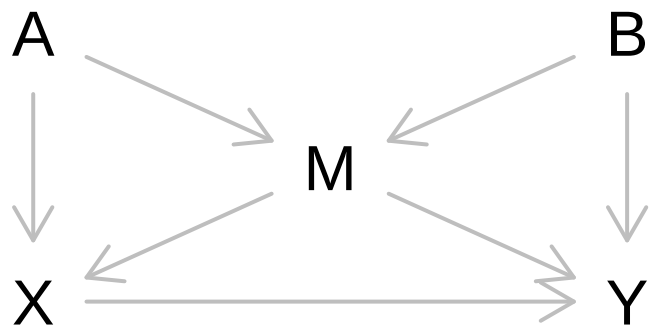
Examples




```
g <- "dag {Z -> {X->Y}}"
adjustmentSets(g,"X","Y")

## { Z }
```

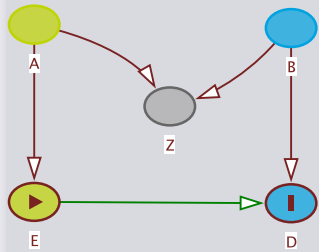



```
g <- "dag{Z <- {X->Y}}"
adjustmentSets(g,"X","Y")

##  {}
```



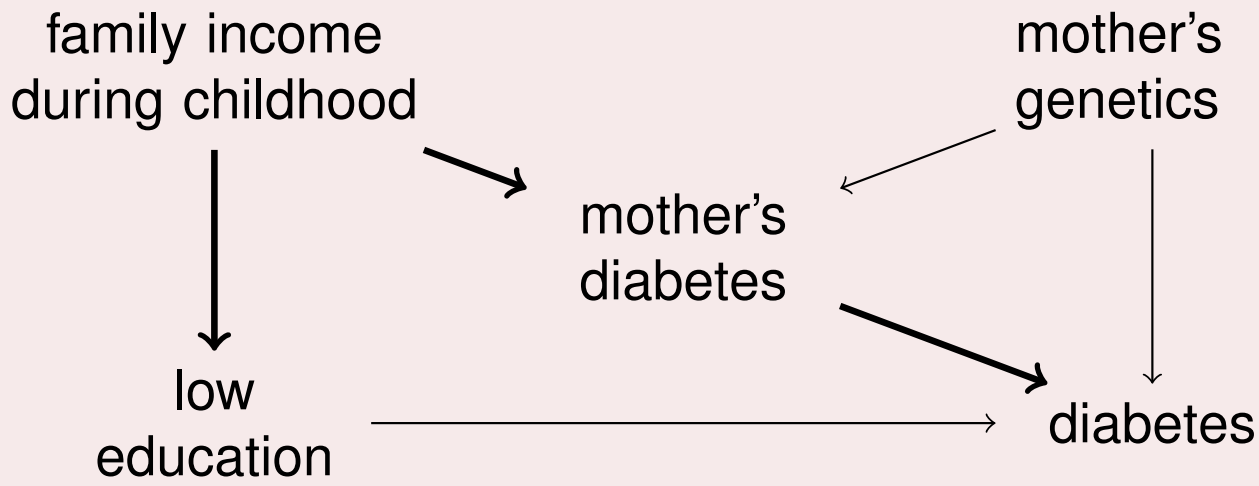

```
g <- "dag{A->{M->X}->Y B->{M->Y}}"
adjustmentSets(g,"X","Y")
```

```
## { A, M }
## { B, M }
## { A, B, M }
```

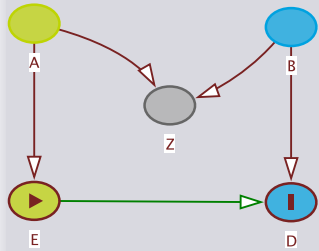
DAGs in Epidemiology

How large is the effect of low education on diabetes risk?



Rothman, Greenland & Lash, Modern Epidemiology, 2008

- Epidemiologists use DAGs to represent causal assumptions.
- The DAGs are drawn by hand (most often), generated from data (seldomly), or both (sometimes).

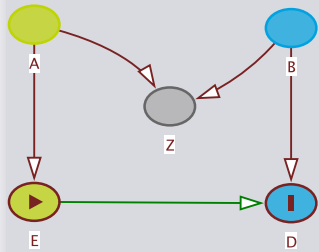


Questions for a Causal Bayesian Network

Hi Mr. Textor,

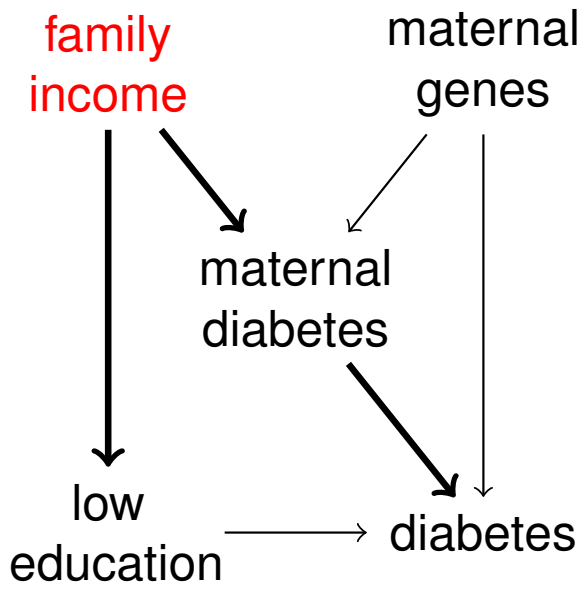
I am trying to learn more about DAGs. I want to see if DAGitty can be used for the attached causal diagram to answer a few of my questions. I am having problems with using the program to help answer these questions.

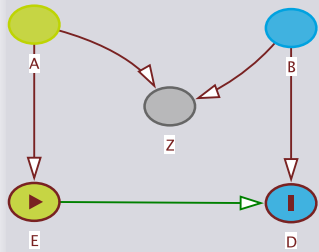
Can you give me some assistance?



Questions for a Causal Bayesian Network

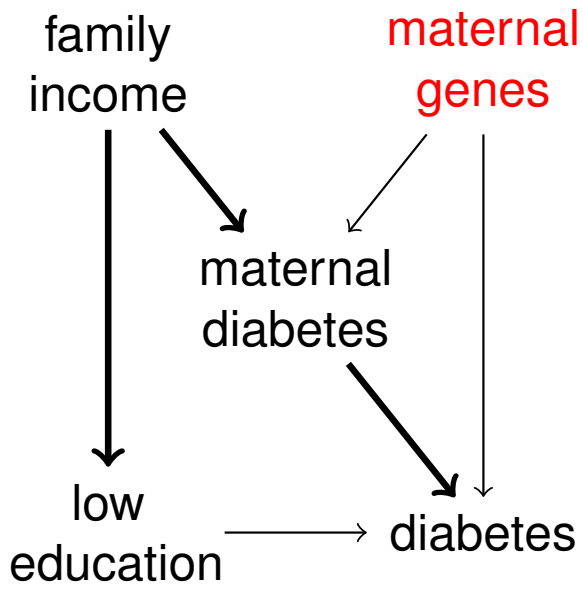
1. Which variable would control for confounding and so reduce bias in estimating the causal effect of the exposure (E) on the disease (D)?

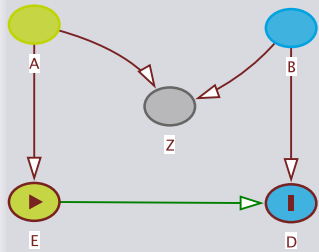




Questions for a Causal Bayesian Network

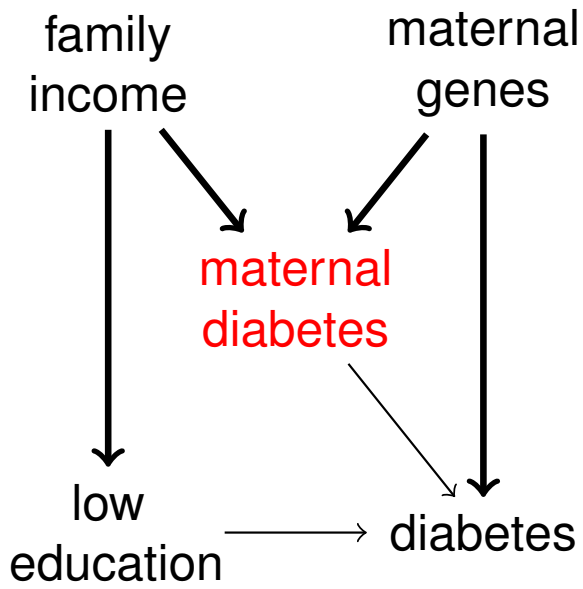
2. Which variable would not impact on the bias in the estimate of causal effect of E on D?

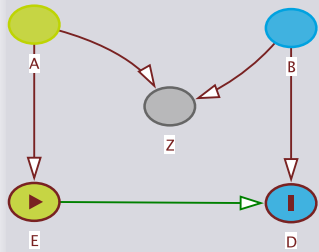




Questions for a Causal Bayesian Network

3. Which variable in the model potentially introduces (additional) bias in the estimate of the causal effect of E on D?

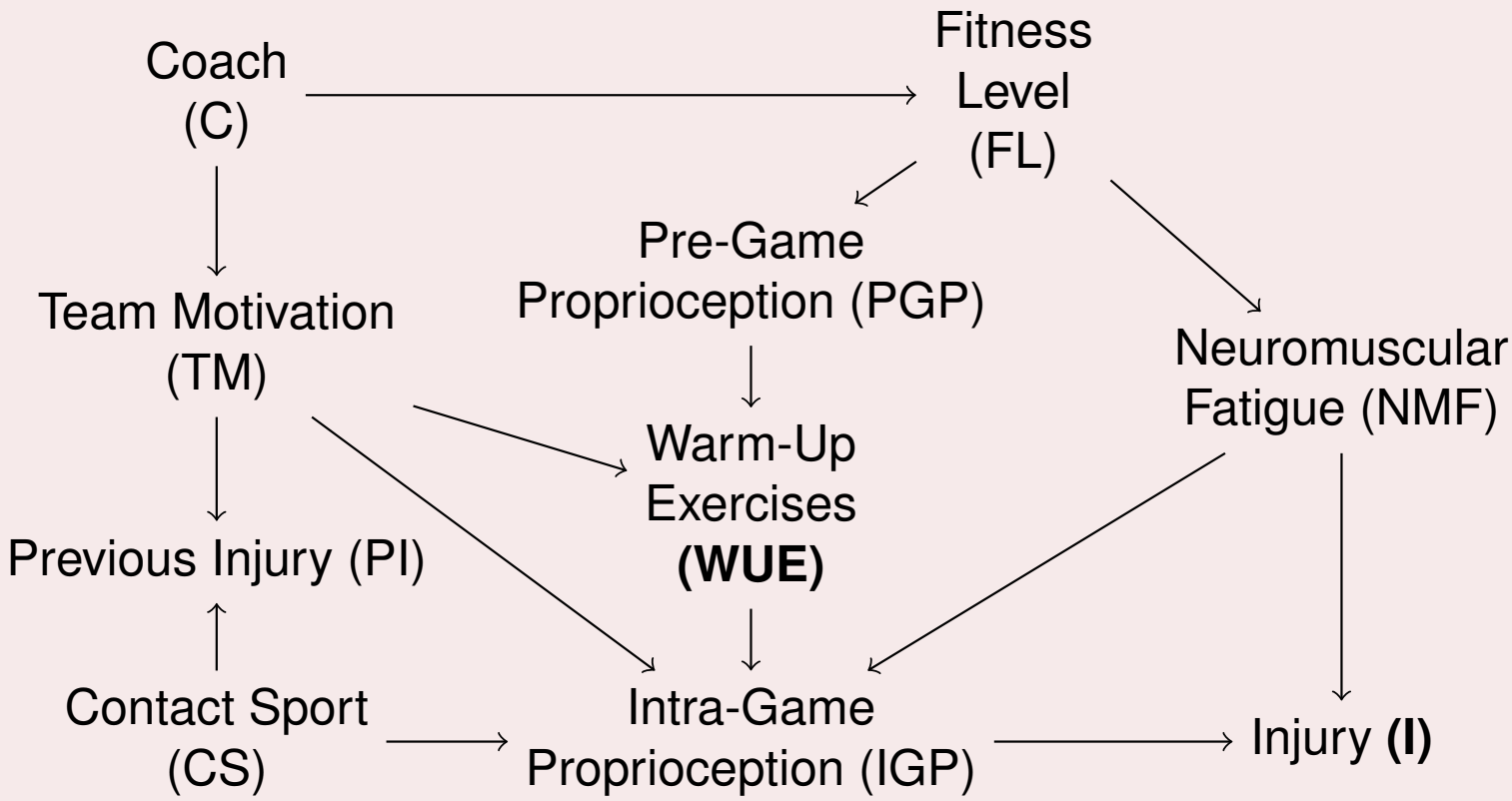




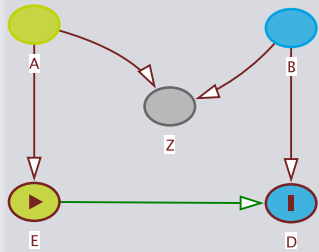
5-Minute Exercise

Exercise

Suppose we want to compute $P(i \mid \mathrm{do}(wue))$ and we know that the following DAG represents our domain correctly:

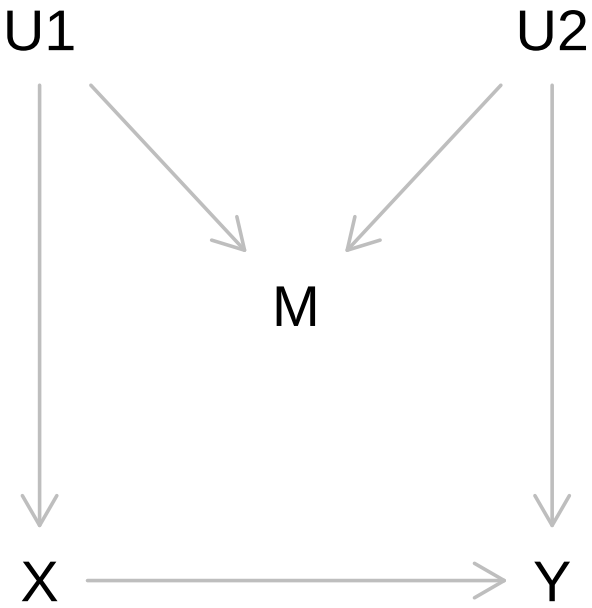


Give 3 different adjustment sets that can be used to remove confounding!



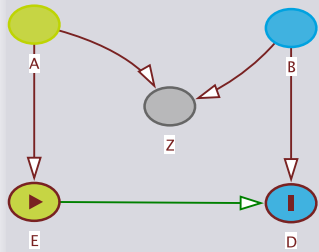
Necessity of Causal Analysis

The “more is better” misconception: “Adjust for all pre-treatment covariates that correlate with exposure and outcome”.
Counterexample: the M-bias graph.



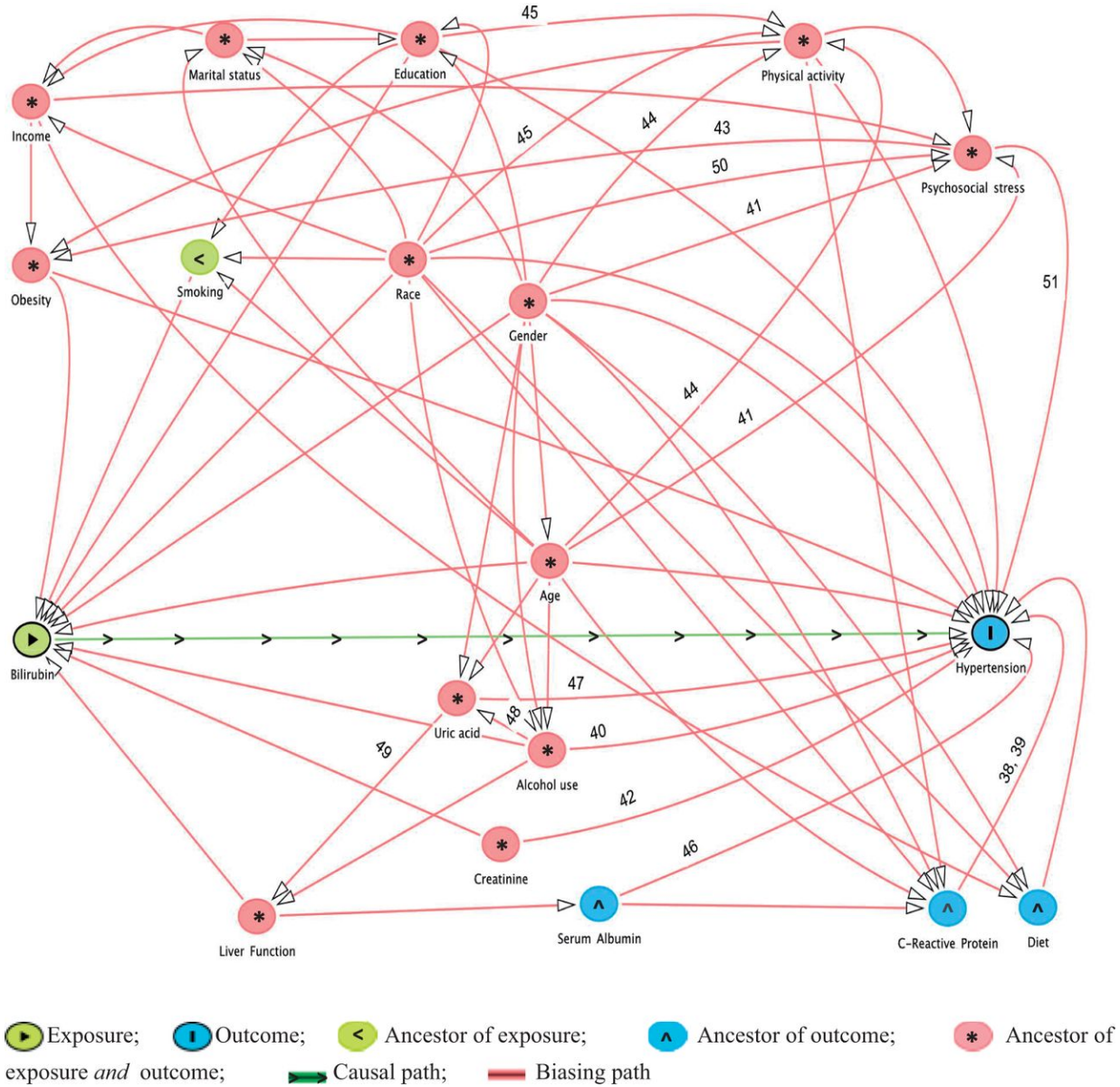

```
isAdjustmentSet(g,Z=c())

## [1] TRUE

isAdjustmentSet(g,Z="M")

## [1] FALSE
```

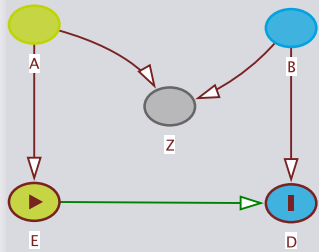
Famous exchange between Donald Rubin and Judea Pearl in “Statistics in Medicine” (2009)



Covariate Adjustment in Practice

Wang and Bautista, Int. J. Epidemiol. (2014)



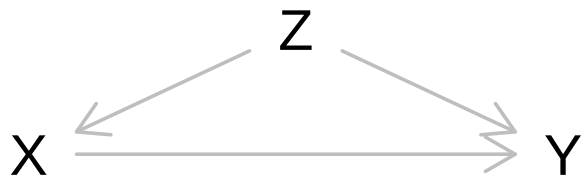


Robust Covariate Adjustment

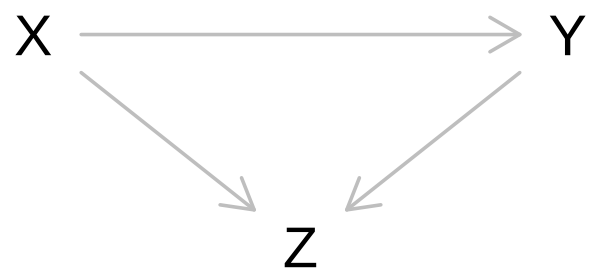
Validity of covariate adjustment depends on correctness of the input DAG.

Hopefully, the DAG has been tested by evaluating its conditional independencies. But even then, the problem of statistical equivalence remains.

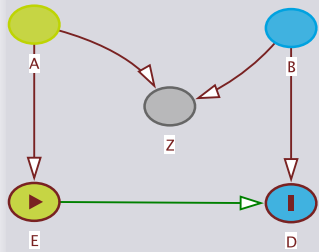
The DAGs below are statistically equivalent, but they imply very different adjustment sets.




```
g <- "dag {Z -> {X->Y}}"
adjustmentSets(g,"X","Y")

## { Z }
```

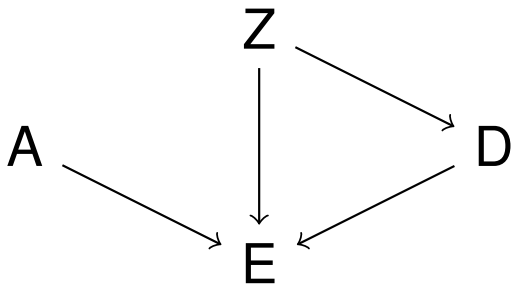



```
g <- "dag{Z <- {X->Y}}"
adjustmentSets(g,"X","Y")

##  {}
```

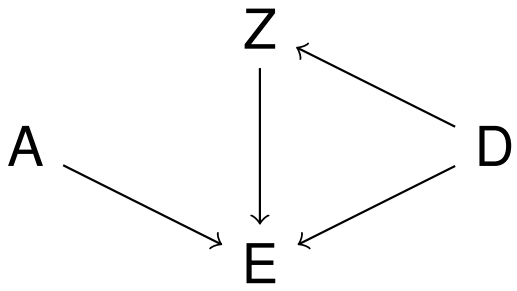



Recap: Markov Equivalence

- DAGs that imply the same *d*-separation relations are called Markov equivalent.

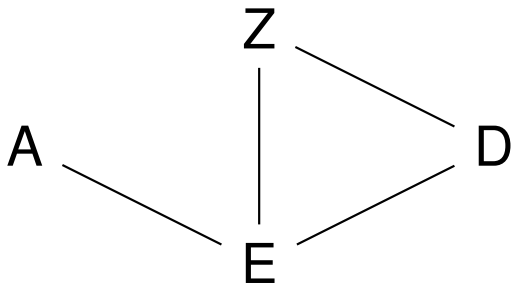


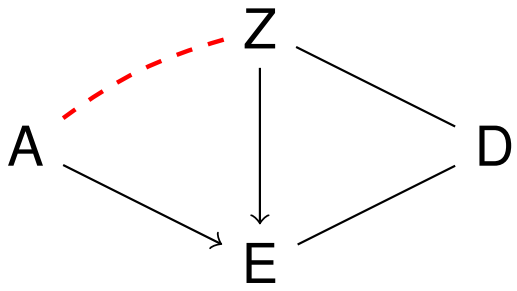
$D \perp\!\!\!\perp A \quad ; \quad A \perp\!\!\!\perp Z$

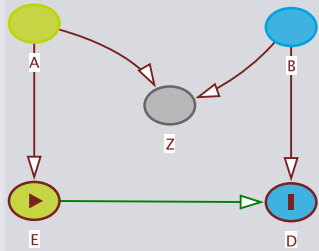


$D \perp\!\!\!\perp A \quad ; \quad A \perp\!\!\!\perp Z$

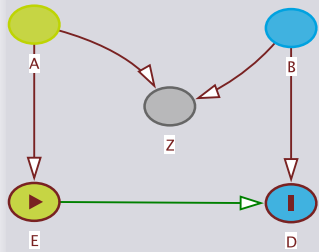
- Two models are Markov equivalent iff they have the same "skeleton" (edges w/o arrowheads) and the same "immoralities" (children of "unmarried" = unlinked parents).







Adjustment for Markov Equivalence Classes



It isn't always necessary to know the whole DAG for causal inference.

Generalized adjustment criterion (Perkovic et al, 2015)

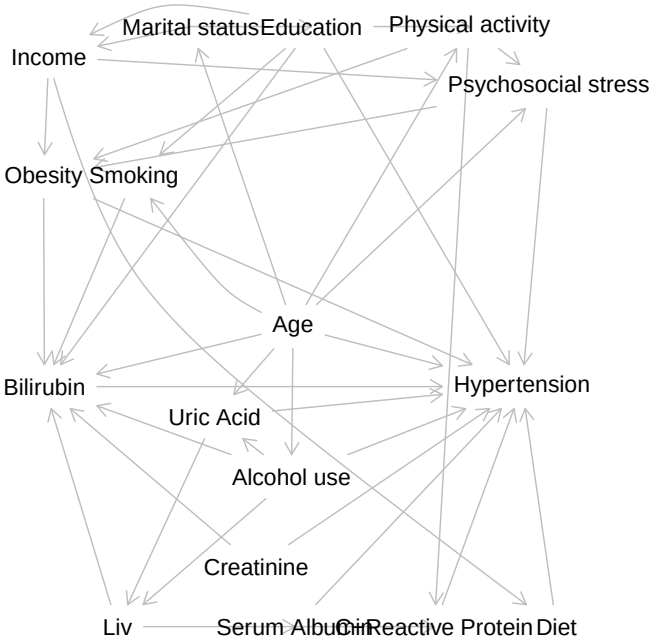
Given a CPDAG (equivalence class) $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, let $\mathbf{Z}$ be a set of covariates such that

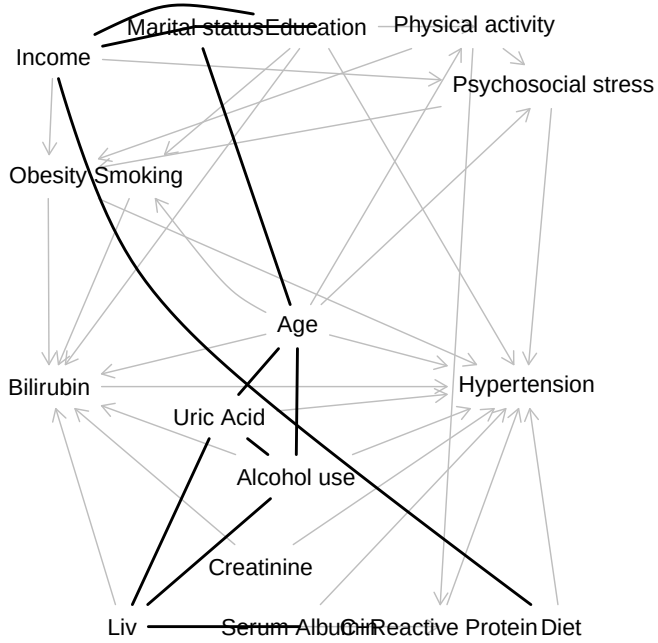
- No possible directed path from X to Y starts with an undirected edge;
- $\mathbf{Z}$ does not contain any possible descendants of a node $W \in \mathbf{V} \setminus X$ on a proper causal path from X to Y;
- $\mathbf{Z}$ blocks all proper non-causal paths from X to Y.

Then $\mathbf{Z}$ is an adjustment set for the causal effect X on Y.

directed: $X \to W \to Y$; possibly directed: $X - W \to Y$

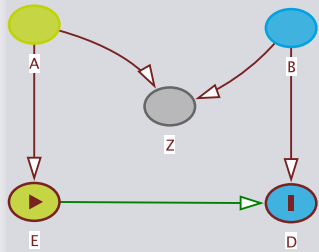
Adjustment for Markov Equivalence Classes






```
adjustmentSets(equivalenceClass(g))
```

```
## Age ; Alcohol use ; Creatinine ; Education ;
## Obesity ; Serum Albumin ; Uric Acid ;
```

Summary

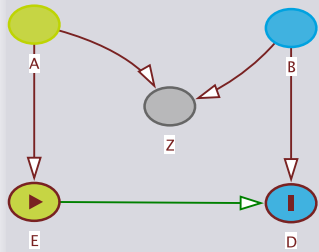
The DAG approach to covariate adjustment:

- Draw DAG based on the best domain knowledge (and validate it!)
- Compute adjustment sets from DAG.
- Collect data.
- If DAG is correct, then adjustment sets will remove confounding bias.

Model misspecifications are not always critical

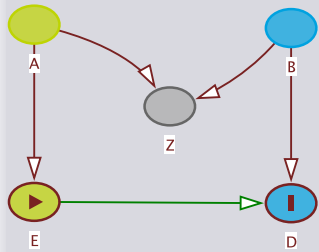
Benito van der Zander, Maciej Liskiewicz, Johannes Textor:
Constructing Separators and Adjustment Sets in Ancestral Graphs.
In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI 2014)*, pp. 907–916. AUAI Press, 2014.

Emilija Perkovic, Johannes Textor, Markus Kalisch, Marloes Maathuis:
A Complete Generalized Adjustment Criterion.
In Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI 2015), pp. 682-691. AUAI Press, 2015.



Summary

- The DAG language allows us to precisely define the difference between observational and interventional regimes.
- This allows us to determine how (if at all) causal effects can be inferred from observational data.
- Once a DAG is specified, such inferences become relatively easy (back-door criterion). The hard part is building the DAG!

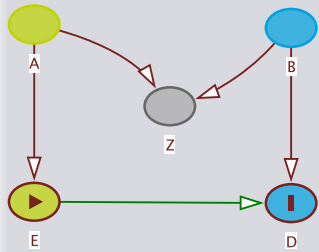


Lecture 9

Instrumental Variables

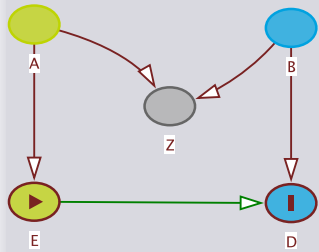
Course Bayesian Networks, December 3rd, 2018

Johannes Textor
Institute for Computing and Information Sciences



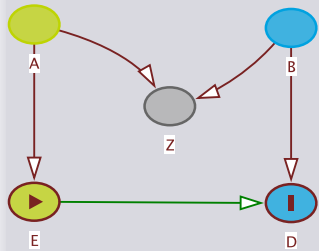
Learning Objectives 9

1. Understand the idea of instrumental variable (IV) estimation.
2. Learn about mendelian randomization, a practically relevant type of IV-based study.
3. Be able to identify both simple and conditional IVs in DAGs.



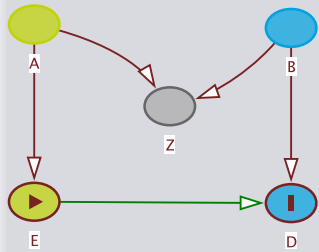
Lecture Outline 9

1. Residual confounding
2. Instrumental Variables (IVs)
3. Conditional IVs



Residual Confounding

- Causal estimates from observational studies are affected by confounding (and many other problems).
- The back-door criterion provides a theoretical answer to confounding.
- But: applying the back-door criterion requires us to know and measure the variables in the adjustment set.
- Often, this may be too difficult in practice.



Do Vitamin C Supplements Keep You Healthy?

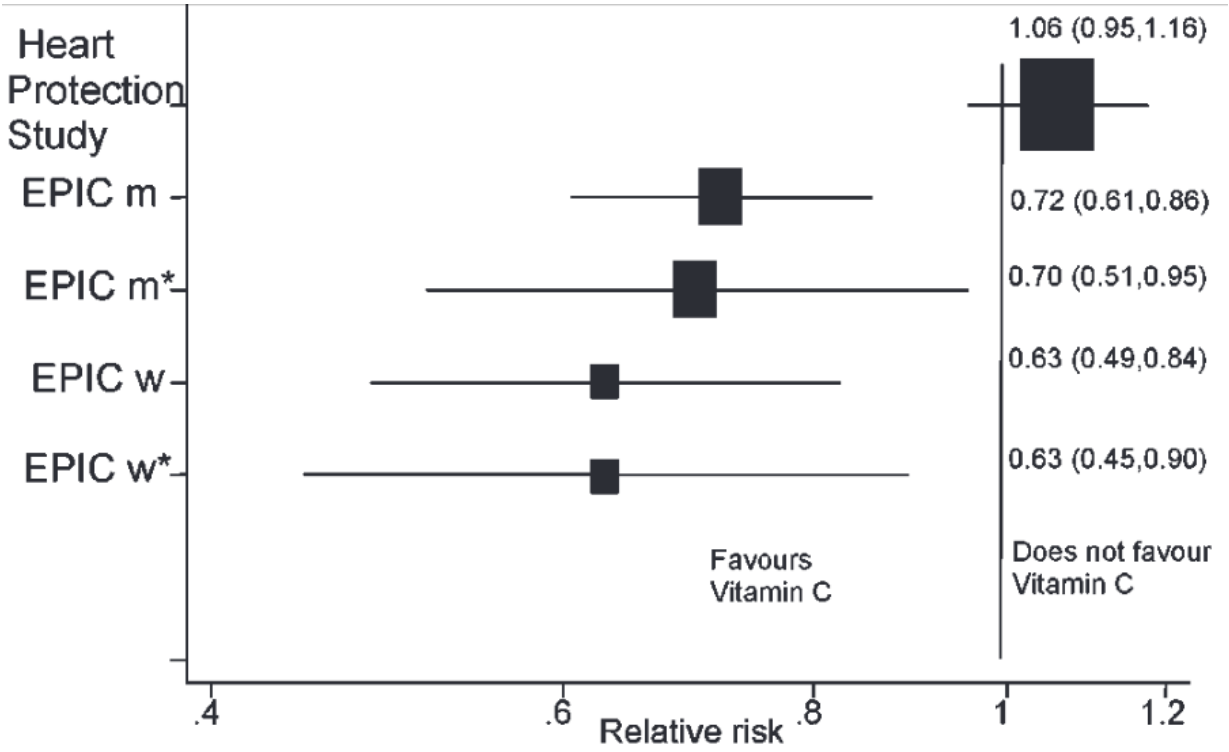
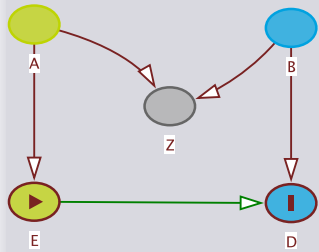


Figure 1 Estimates of the effects of an increase of 15.7 µmol/l plasma vitamin C on coronary heart disease 5-year mortality estimated from observational epidemiological EPIC study[27] and randomized controlled Heart Protection Study.[28] (EPIC m = men, age-adjusted; EPIC m* = men, adjusted for systolic blood pressure, cholesterol, body mass index, smoking, diabetes, and vitamin supplement use; EPIC w = women, age-adjusted; EPIC w* = women, adjusted for systolic blood pressure, cholesterol, body mass index, smoking, diabetes, and vitamin supplement use)

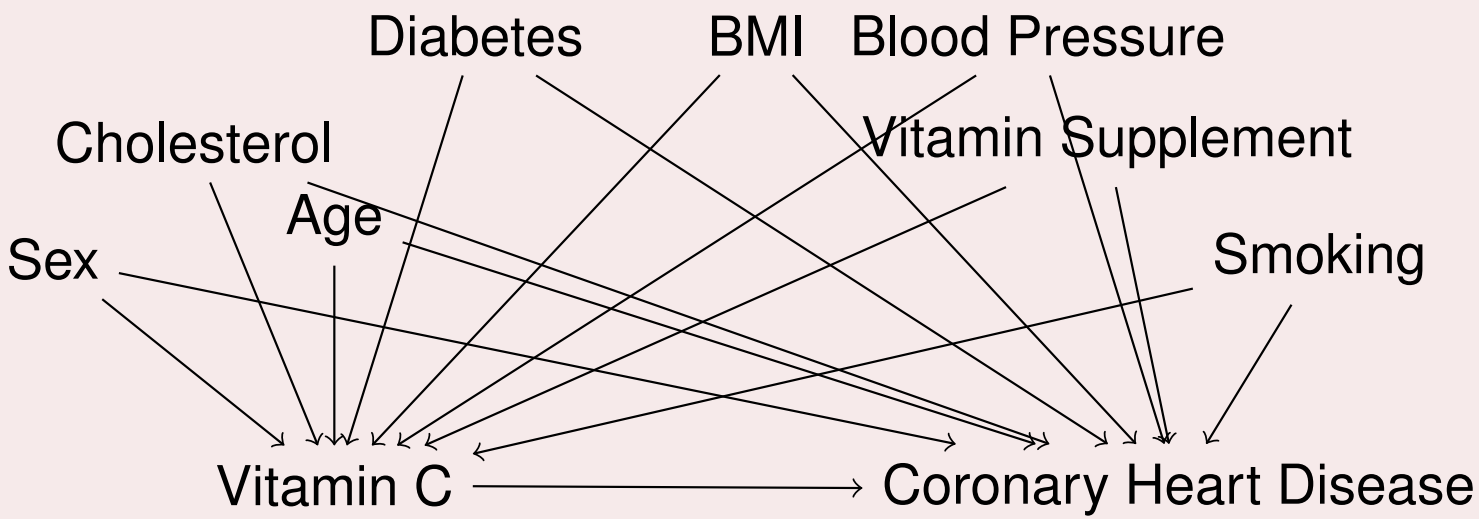
Source: doi 10.1093/ije/dyg070



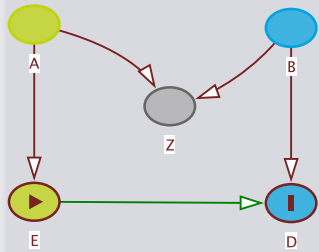
5-Minute Exercise

Exercise

The observational design of the vitamin C study corresponds to the following DAG:

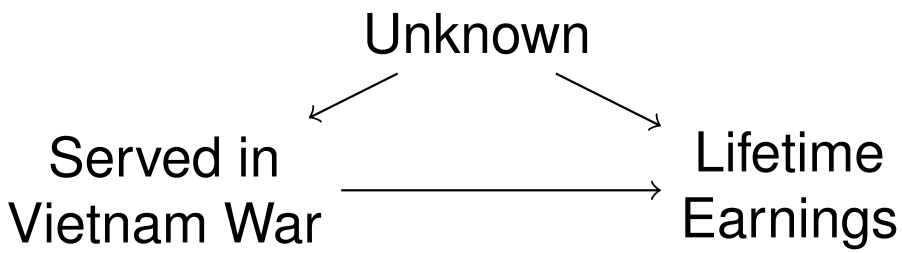


Can you think of any confounders they may have left out?



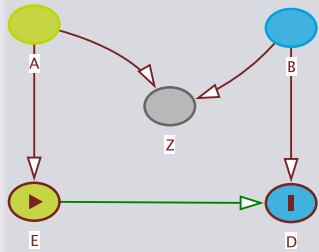
Residual Confounding: War Veterans

There is much evidence that Vietnam veterans are worse off later in life. It is tempting to conclude that these problems are due to their experiences in the war. But various unknown factors could have caused both the decision to join the military and the later earnings.



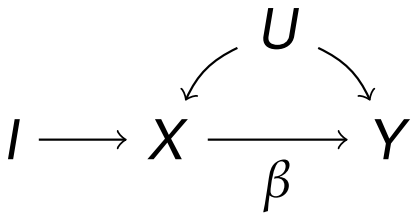
We discussed before how covariate adjustment can be used to control for these confounding factors, but that requires that we know and measure the important sources of confounding.

Joshua D. Angrist:
Estimating the Labor Market Impact of Voluntary Military Service Using Social Security Data on Military Applicants.
Econometrica 66(2):249–288, 1998



Instrumental Variables (IVs)

- Goal: estimate causal effect of X on Y.
- Problem: an unobserved confounder U.
- Solution: two-stages least squares using an instrumental variable (IV).

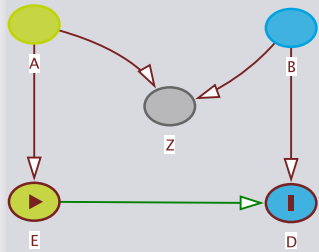


Exogeneity: $I \perp\!\!\!\perp U$.
Exclusion restriction:
I only affects Y through X, but not directly.

```
m1 <- lm( Y ~ X )
library( AER ); m2 <- ivreg( Y ~ X | I )
```

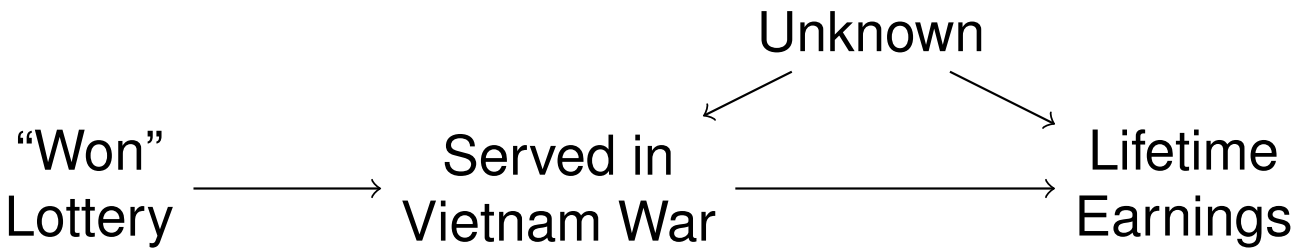
Two-stages least squares

1. Regress X on I and obtain the prediction $\hat{X}$.
2. Regress Y on $\hat{X}$.



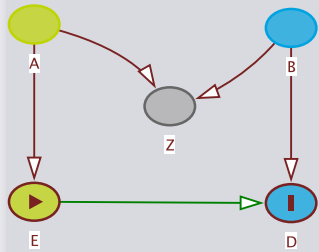
The Vietnam Draft Lottery

In 1969, men were called in a random order determined by their birthdays, and asked to serve in the war. 195 out of 366 possible birthdays were "drafted". For example, men born on a September 14th were drafted, but men born on a June 20th were not. Not every drafted person enlisted.



Angrist used "draft" as an IV to determine the effect of serving in the war on lifetime earnings.

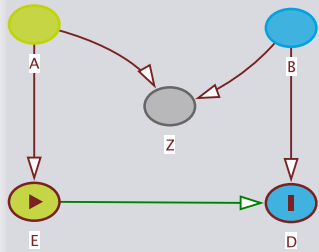
```
m1 <- lm( earn ~ enlistment )
m2 <- ivreg( earn ~ served | won )
m3 <- ivreg( earn ~ enlist + yearofbirth | served + yearofbirth )
```

Genetic IVs: Mendelian Randomization

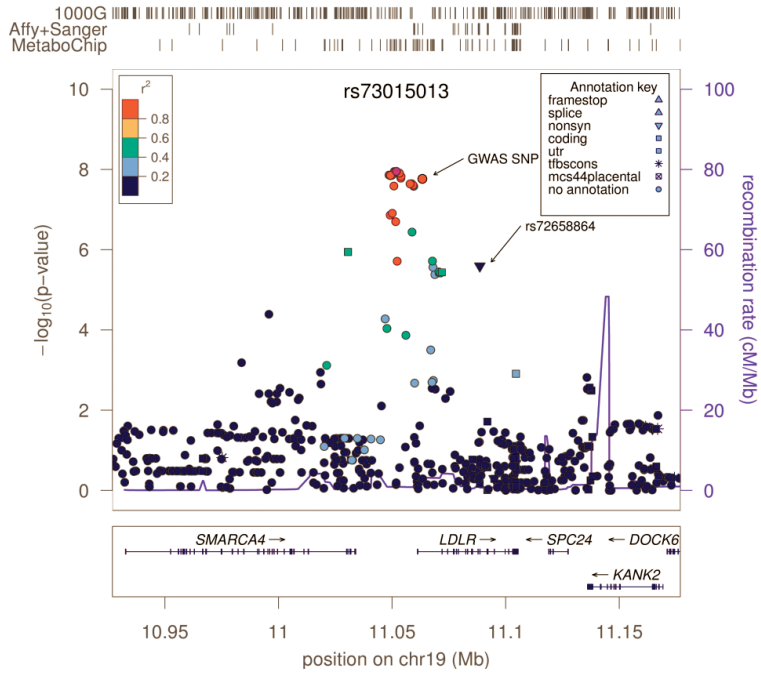
- Despite the elegance of the Vietnam war lottery, one wonders how often we are lucky to find variables like this.
- Indeed, IVs are sometimes described as “unicorns”.
- In recent years, one specific form of IVs has gained enormous traction: genetic instruments.
- This specific form of IV analysis is also called Mendelian Randomization.

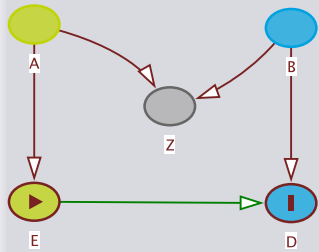
George Davey Smith, Shah Ebrahim:
‘Mendelian randomization’: can genetic epidemiology contribute to understanding environmental determinants of disease?
International Journal of Epidemiology, 32(1):1–22, 2003



Single Nucleotide Polymorphisms

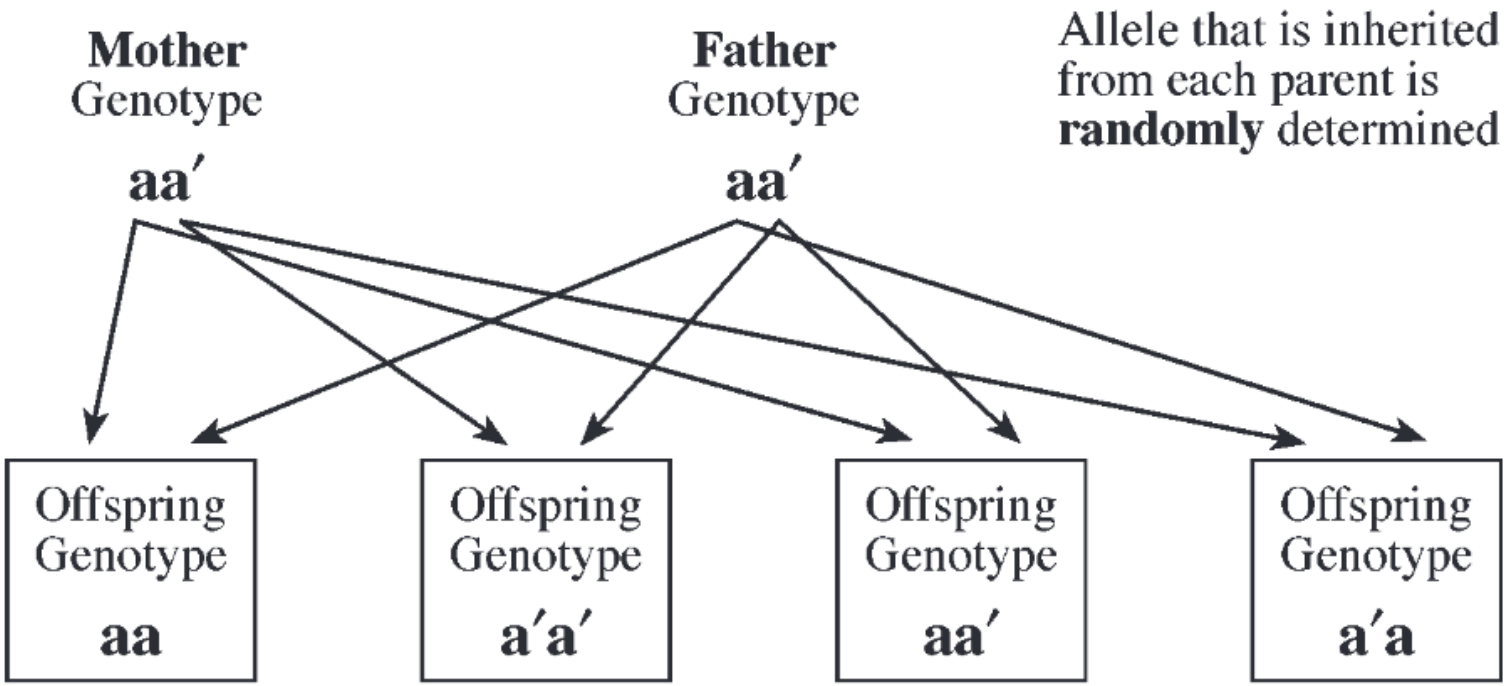
A single nucleotide polymorphism (SNP) is a single-basepair variation in the genome. In a genome-wide association study (GWAS), we seek statistical associations between SNPs and diseases or traits such as obesity.

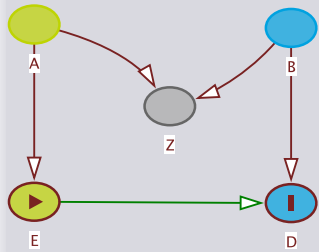




Random Assignment of SNPs

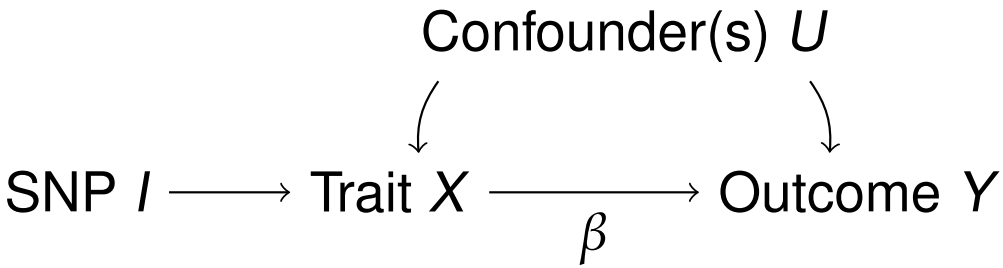
Suppose a diploid cell carries two possible variations of a SNP. During meiosis, one of these two SNPs is selected for the sperm or egg cell. This selection occurs randomly. The two SNP versions of sperm and egg cell determine the phenotype of the offspring.





Using SNPs as Instruments

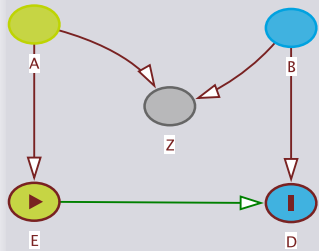
Mendelian randomization (MR) refers to the use of SNPs as instruments.



Let's check the assumptions:

- Exogeneity: Because I is randomly assigned, there can be no arrow going into I. But there may still be an arrow from I to U.
- Exclusion restriction: There may still be direct arrows from I to Y.

These assumptions remain fundamentally untestable (as they have to be), but many types of sanity checks and sensitivity analyses have been developed for MR.



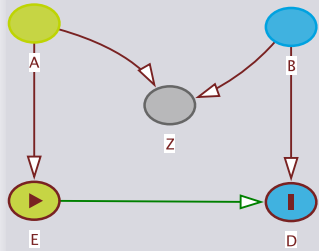
An Example MR Study: Vitamin D Supplements

Author summary

Why was this study done?

- Observational epidemiological studies have reported associations of low vitamin D levels with risk of asthma, atopic dermatitis, and elevated immunoglobulin E (IgE) levels. However, these studies are susceptible to confounding and reverse causation, and thus it remains unclear whether these associations are genuine.
- The randomized controlled trials published to date on this topic have been inconclusive.
- If vitamin D insufficiency did cause atopic diseases, this would be of clinical relevance, since vitamin D insufficiency is common and safely correctable.

Despoina Manousaki *et al.*:
Vitamin D levels and susceptibility to asthma, elevated immunoglobulin E levels, and atopic dermatitis: A Mendelian randomization study.
PLoS Medicine, 10.1371/journal.pmed.1002294 2017



An Example MR Study: Vitamin D Supplements

Vitamin D (25OHD) results								Asthma results			Childhood asthma results§			Atopic dermatitis results∞			IgE results§		
Locus	**25OHD associated SNP**	**EA**	**EAF**	**Effect on 25OHD***	***p*#**	**F-Statistic‡**	**Variance in 25OHD explained by each SNP (%)**	**OR (95% CI)**	***p***	***n***	**OR (95% CI)**	***p***	***n***	**OR (95% CI)**	***p***	***n***	**Beta (95% CI)**	***p***	***n***
CYP2R1	rs10741657	C	0.62	−0.052	3.3×10^{-20}	18.78	0.13	0.99 (0.97–1.01)	0.54	142,551	1.02 (0.96–1.07)	0.56	15,008	1.02 (0.99–1.05)	0.27	40,834	−0.02 (−0.23 to 0.19)	0.86	12,853
DHCR7	rs12785878	G	0.27	−0.056	2.1×10^{-27}	18.29	0.12	1.01 (0.98–1.03)	0.64	142,551	0.95 (0.90–1.01)	0.11	15,008	1.02 (0.98–1.06)	0.32	40,834	−0.15 (−0.36 to 0.06)	0.20	12,853
GC	rs2282679	C	0.3	−0.047	1.9×10^{-109}	13.38	0.09	1.01 (0.99–1.04)	0.31	144,243	1.00 (0.95–1.06)	0.96	15,008	0.98 (0.94–1.02)	0.32	40,531	0.06 (−0.17 to 0.29)	0.60	12,853
CYP24A1	rs6013897	A	0.19	−0.027	6.0×10^{-10}	3.13	0.02	1.02 (0.99–1.05)	0.14	144,243	1.03 (0.97–1.10)	0.38	15,008	1.03 (0.99–1.07)	0.22	40,529	0.02 (−0.25 to 0.29)	0.90	12,853

250HD, 25-hydroxyvitamin D; 95% CI, 95% confidence interval; EA, effect allele; EAF, effect allele frequency.

*Effect on natural log-transformed 25OHD levels in the Canadian Multicentre Osteoporosis Study (CaMos) Cohort, adjusted for age, age^2, sex, season of blood draw, and body mass index (BMI).

#*p*-Values derived from the Study of Underlying Genetic Determinants of Vitamin D and Highly Related Traits (SUNLIGHT) Consortium.

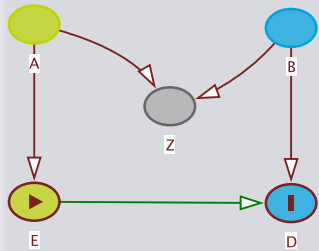
‡F-Statistic derived from multiply adjusted natural log-transformed 25OHD levels in the CaMos Cohort.

ᵇResults are derived from the meta-analysis of the UK Biobank study and the GABRIEL asthma consortium.

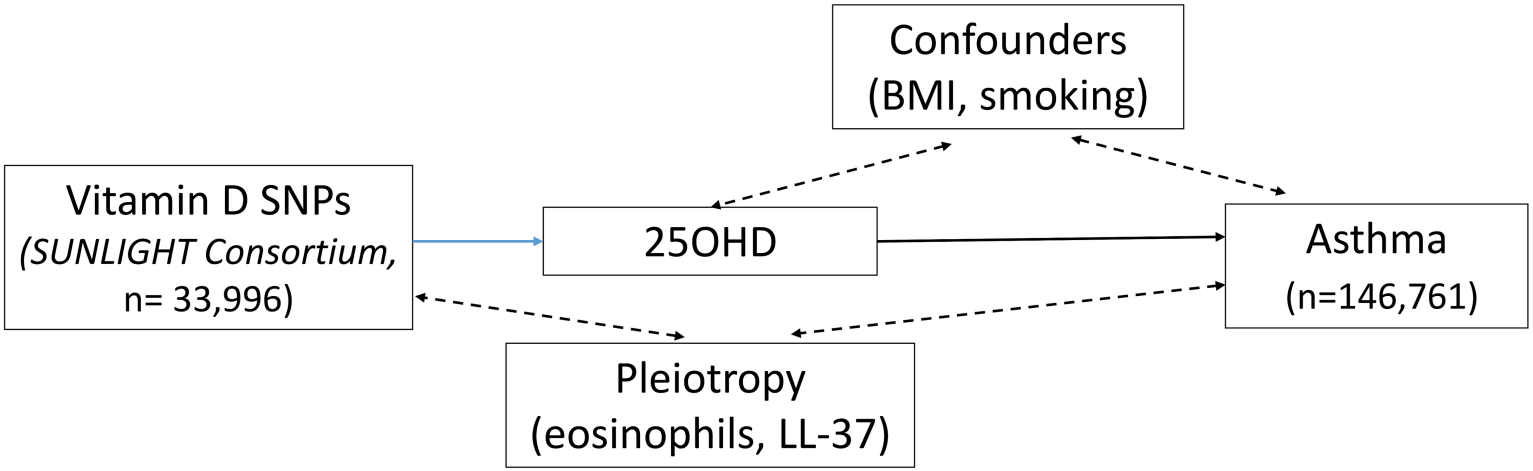
§ Results are derived from the GABRIEL asthma consortium.

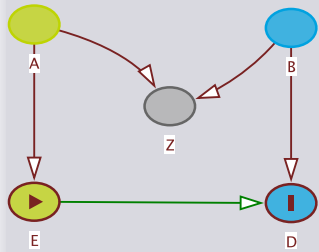
∞ Results are derived from the Early Genetics and Lifecourse Epidemiology (EAGLE) Eczema Consortium.

https://doi.org/10.1371/journal.pmed.1002294.t001



An Example MR Study: Vitamin D Supplements





An Example MR Study: Vitamin D Supplements

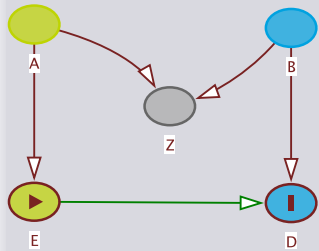
Outcome	MR estimate odds ratio or beta (95% CI)	*p*	I^2 (95% CI)
Asthma	1.03 (0.90–1.19)*	0.63	0% (0%–85%)
Childhood asthma	0.95 (0.69–1.31)*	0.76	0% (0%–85%)
Atopic dermatitis	1.12 (0.92–1.37)*	0.27	15% (0%–87%)
IgE levels	−0.40 (−1.65 to 0.85)**	0.54	0% (0%–85%)

95% CI, 95% confidence interval.

*Odds ratio (OR) is expressed as the odds of asthma or atopic dermatitis per standard deviation decrease in natural log-transformed 25OHD levels.

**Beta is the effect per standard deviation decrease in natural log-transformed 25OHD levels on natural log-transformed total IgE levels.

https://doi.org/10.1371/journal.pmed.1002294.t002



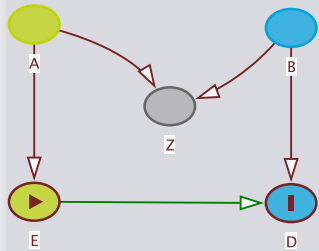
Conclusion of the Vitamin D Study

What did the researchers do and find?

- We applied a Mendelian randomization study design, which greatly limits bias due to confounding and prevents bias due to reverse causation, to understand if vitamin D levels are associated with a higher risk of adult and pediatric asthma, atopic dermatitis, or elevated IgE levels.
- Despite high statistical power, our study showed no evidence of an unconfounded association between vitamin D and the studied outcomes.

What do these findings mean?

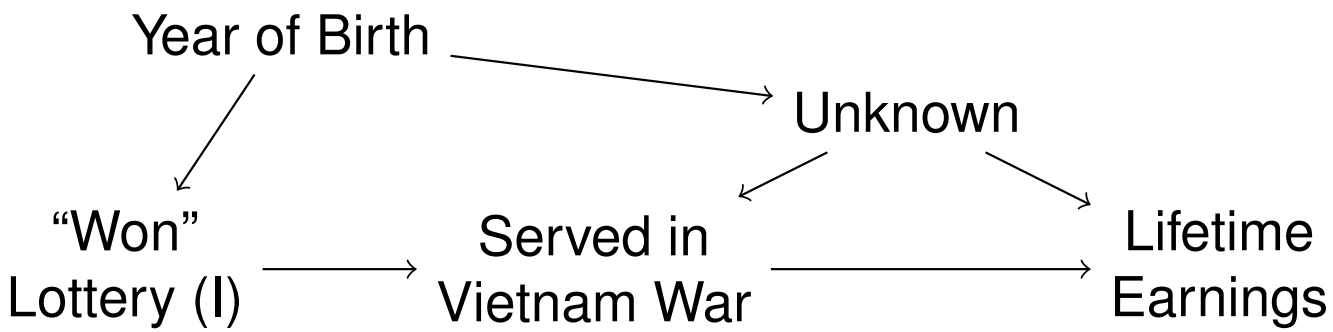
- Our findings suggest that the previous epidemiological associations between vitamin D and atopic diseases could be due to confounding.
- Efforts to increase vitamin D levels will probably not result in decreased risk of adult and pediatric asthma, atopic dermatitis, or elevated IgE levels.



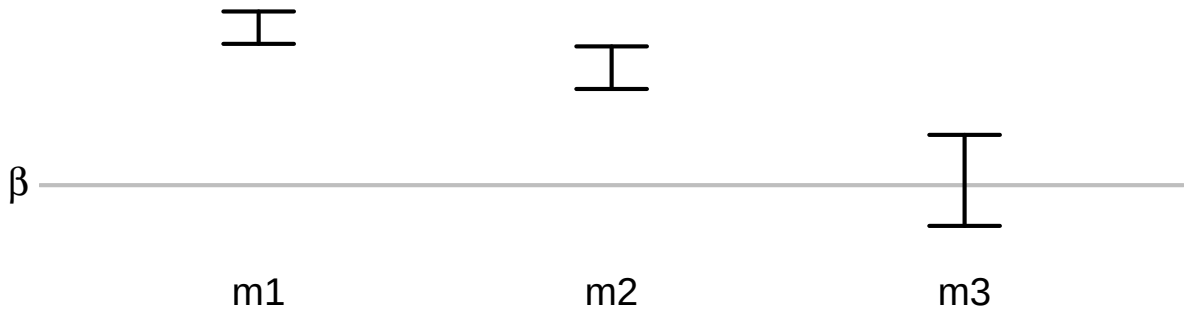
Motivation for Conditional IVs

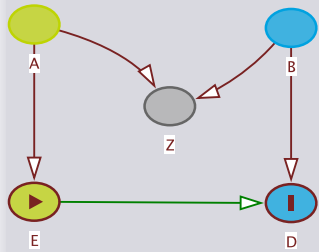
Despite the clear success of MR, IVs remain hard to find in many if not most cases. It would be useful if we could apply IV methods in more general cases.

The Vietnam Draft Lottery was repeated a few times, but with slightly different procedures. When combining these data, the "exogeneity" criterion $I \perp\!\!\!\perp U$ is violated.

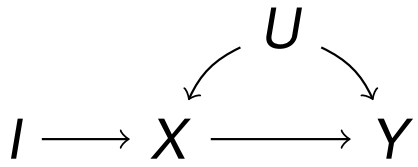


However, we can condition on the year of birth to to d-separate I from U. We say that I is a conditional IV.

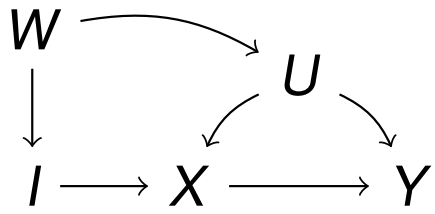




Conditional IVs in Bayesian Networks



The classic IV model.



I is a conditional IV given *W*.

Graphical Definition (Brito and Pearl)

I is a *conditional IV* if for some observed variables **W**,

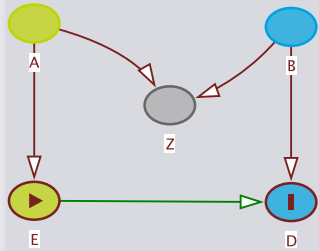
*(a) *I* correlates with *X* conditional on **W**,

(b) **W** *d*-separates *I* and *Y* when $X \to Y$ is removed,

(c) **W** contains no descendants of *Y*.

Instrumentalization: Given net *G* and $X \to Y$, *I*, find **W** that

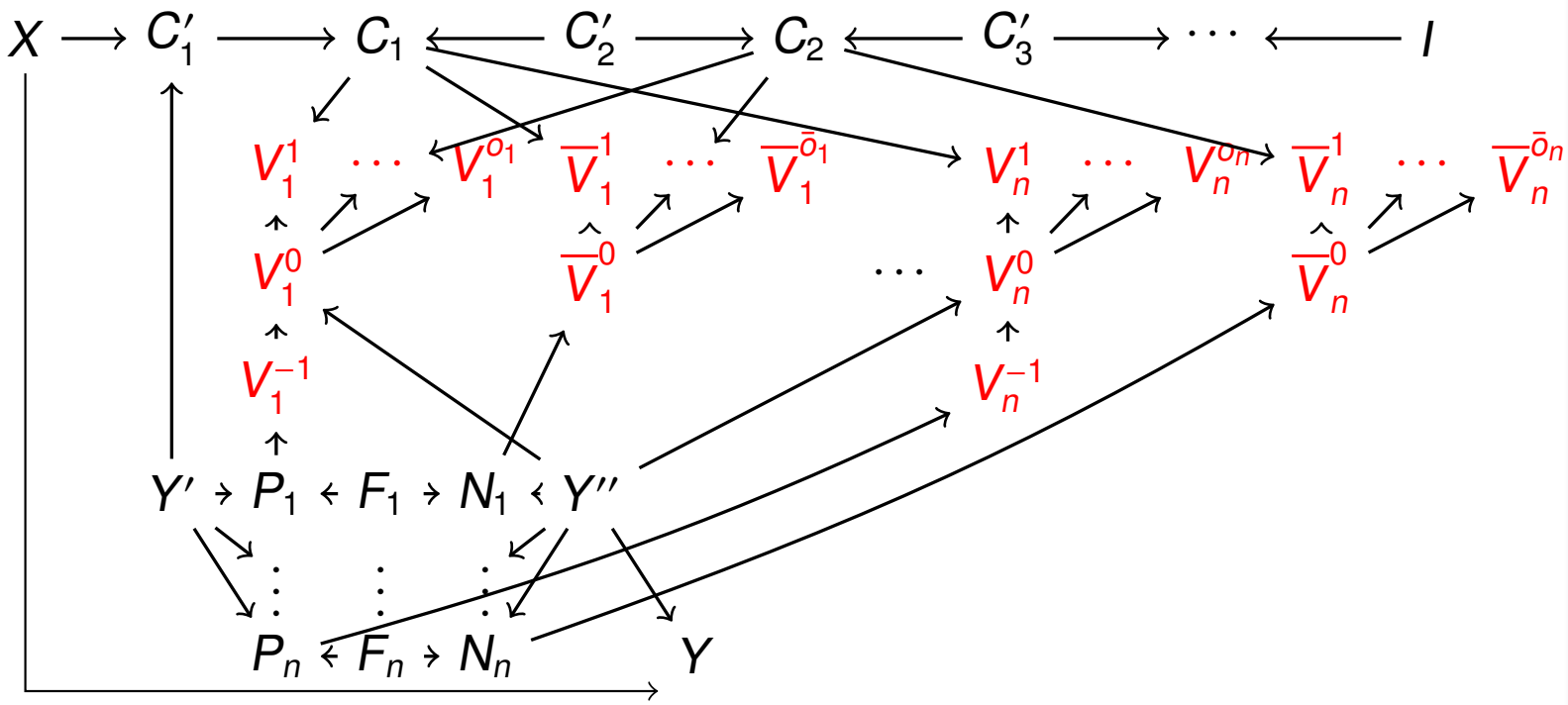
- d-separates *I* and *Y* but
- does not d-separate *I* and *X*.



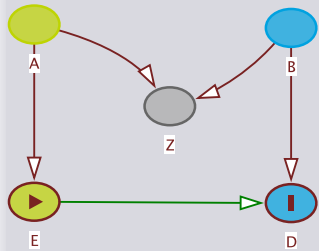
Instrumentalization is NP-hard

Theorem

Determining whether I is a conditional IV relative to $X \to Y$ is an NP-hard problem.

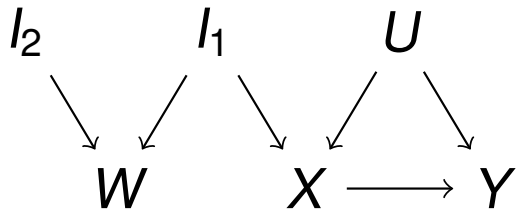


Proof by reduction from 3-SAT (C_i: clauses in formula). Task: Choose P_n, F_n to Connect X to I but keep it separate from Y.

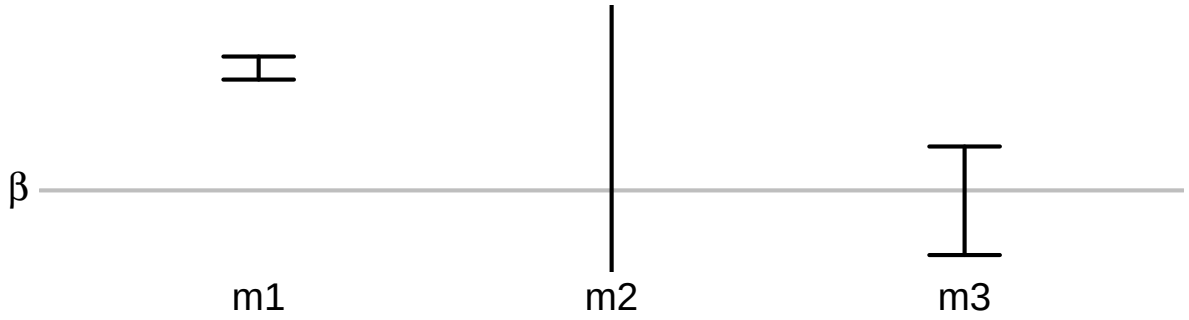


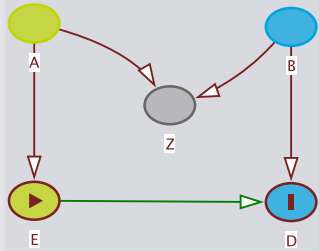
Conditional IVs and Spurious Correlations

Conditioning on **W** is normally meant to remove confounding from an IV. But we can also use it to create an IV.




```
m1 <- lm( Y ~ X )
m2 <- ivreg( Y ~ X | I2 )
m3 <- ivreg( Y ~ X + W | I2 + W )
```





Ancestral IVs

We introduce a restricted definition that is more in line with the "remove confounding" interpretation of conditional IVs.

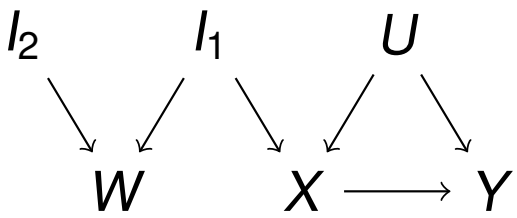
Definition (Ancestral IVs)

I is an *ancestral IV* if for some observed variables $\mathbf{W}$,

*(a) I correlates with X conditional on $\mathbf{W}$,

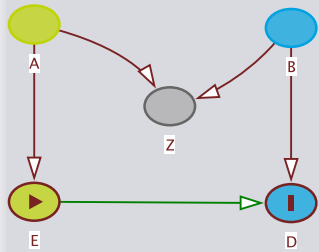
(b) $\mathbf{W}$ d-separates I and Y when $X \to Y$ is removed,

(c) $\mathbf{W}$ contains only ancestors of Y, I and no descendants of Y.



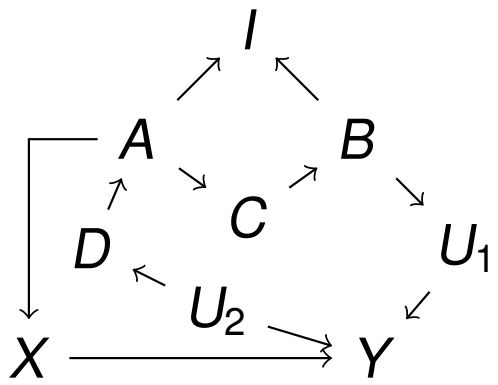
I_1: ancestral IV

I_2: non-ancestral IV conditional on W

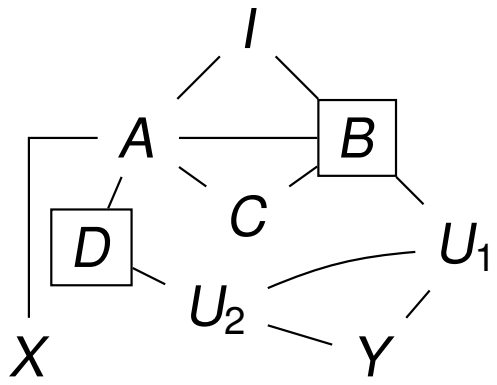


Instrumentalization for Ancestral IVs: Easy

function ANCESTRAL-INSTRUMENT($\mathcal{G}, X, Y, Z$)
  $\mathcal{G}_c := \mathcal{G}$ with edge $X \to Y$ removed
  $\mathbf{W} :=$ NEAREST-SEPARATOR($\mathcal{G}_c, Y, Z$)
  if $(\mathbf{W} = \perp) \vee (\mathbf{W} \cap De(Y) \neq \emptyset) \vee (X \in \mathbf{W})$ **then**
    return $\perp$
  if $(Z \not\perp\!\!\!\perp X \mid \mathbf{W})$ in $\mathcal{G}_c$ **then return W else return** $\perp$

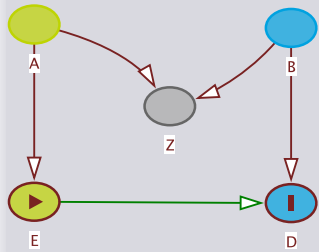


$\mathcal{G}$



$\mathcal{G}_c$ (moral graph)

NEAREST-SEPARATOR($\mathcal{G}_c, Y, Z$) $= \{D, B\}$ (but not $\{A, B\}$)
Blocks only those paths necessary to separate I and Y but not more



Putting it Together

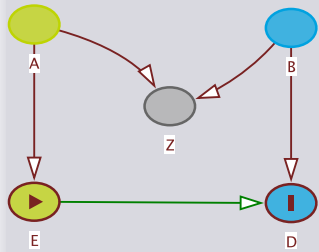
- We can not (efficiently) instrumentalize IVs, but
- We can (efficiently) instrumentalize ancestral IVs.

Theorem

For a given DAG $\mathcal{G}$ and variables X and Y, a conditional IV I relative to $X \to Y$ exists if and only if an ancestral IV I' relative to $X \to Y$ exists.

Whenever there exists a conditional IV in a given graphical model, we can find one efficiently.

B. van der Zander, J. Textor, and M. Liśkiewicz:
Efficiently finding conditional instruments for causal inference.
Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), pages 3243–3249. AAAI Press, 2015.

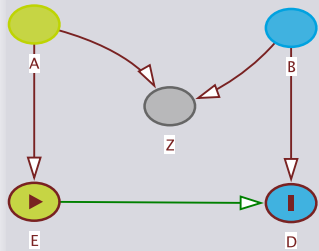


Implementation in R

The R package `dagitty` provides a command to list IVs, including conditional IVs.

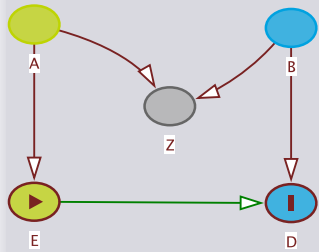
```
library(dagitty)
d <- dagitty("dag{ U -> X ; U -> Y ; I -> X -> Y }")
exposures(d) <- "X" ; outcomes(d) <- "Y"
instrumentalVariables( d )

##  I |  U
```

```
d <- getExample("Shrier") # The contact sport DAG
instrumentalVariables( d )

##  Coach |  ContactSport, NeuromuscularFatigue, TissueWeakness
##  ConnectiveTissueDisorder |  ContactSport, NeuromuscularFatigue, TissueWeakness
##  FitnessLevel |  ContactSport, NeuromuscularFatigue, TissueWeakness
##  Genetics |  ContactSport, NeuromuscularFatigue, TissueWeakness
##  PreGameProprioception |  ContactSport, NeuromuscularFatigue, TissueWeakness
##  PreviousInjury |  ContactSport, NeuromuscularFatigue, TissueWeakness
##  TeamMotivation |  ContactSport, NeuromuscularFatigue, TissueWeakness
```

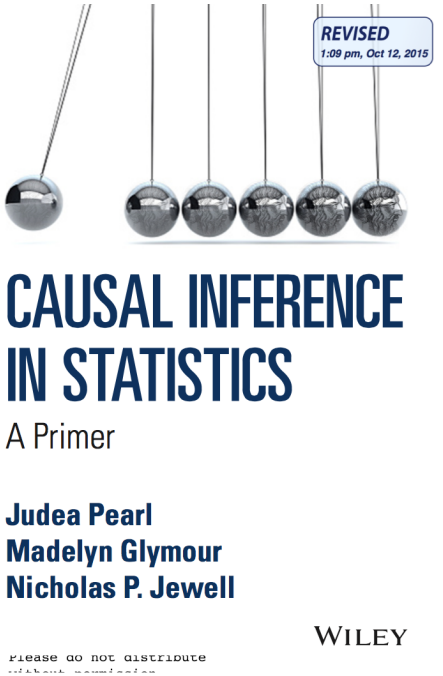
Summary

- Conditional IVs are a natural generalization of IVs, but they are hard to find.
- It suffices to consider ancestral conditional IVs
- Ancestral conditional IVs can be found quickly in Bayesian networks.

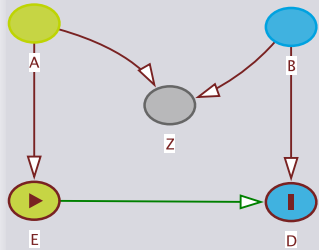


Conclusion

- Bayesian networks can be used to encode causal assumptions and derive their implications.
- This is especially useful when attempting to devise causal effects from observational data.
- Two ways to do that are covariate adjustment (previous lecture) and instrumental variables (this lecture).



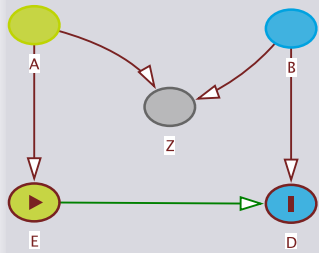
dagitty.net/primer



Lecture 10
Exact Inference

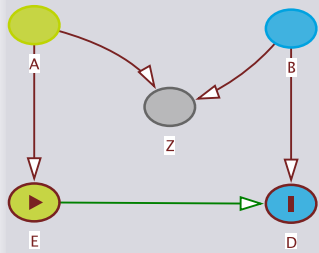
Course Bayesian Networks, December 10th, 2018

Johannes Textor
Institute for Computing and Information Sciences

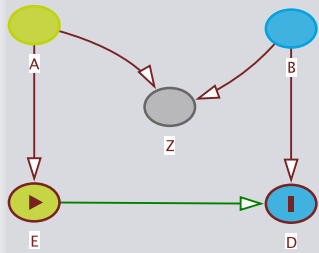


Learning Objectives 10

1. Understand why exact inference in Bayesian networks is a hard problem.
2. Know what factor graphs are and how to derive them from Bayesian networks (next week).
3. Be able to apply the message passing algorithm to factor graphs (next week).



Lecture Outline 10



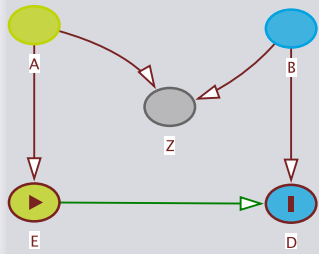
Recap: Inference

We said that inference in a Bayes net means computing marginal or conditional probabilities – that is, expressions of the form $P(x_i)$ or $P(x_i \mid e)$.

We can restrict our attention to marginal probabilities because

$$P(x_i \mid e) = \frac{P(x_i, e)}{P(e)} .$$

In this lecture, we first focus on a subclass of inference problems: compute marginal probabilities $P(x_i)$ for a single variable X_i.



Recap: NP-hardness

Definition

A decision problem is a set $S \subseteq \mathbf{N}$.

Definition

NP is the set of decision problems whose membership function can be computed in polynomial time by a non-deterministic Turing machine.

Reduction

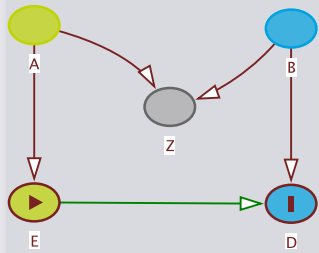
We say that a polynomial time reduction from P to S exists

$$P \leq_P S$$

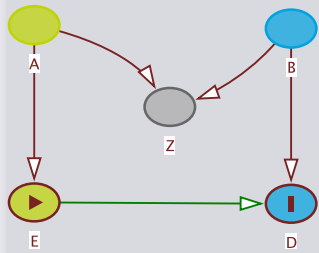
if there is a polynomial-time computable function r such that

$$x \in P \Leftrightarrow r(x) \in S \ .$$

(S can simulate P with polynomial-time overhead.)



3-SAT



Definition

A decision problem S is called NP-hard if, for every other decision problem $P \in \mathrm{NP}$, $P \leq_P S$.

Boolean satisfiability is the "canonical" NP-hard problem:

$$\mathrm{SAT} = \{\varphi \mid \varphi \text{ is a satisfiable Boolean formula}\}$$

It suffices to consider only formulas of a certain normal form:

$$\text{3-SAT} = \{\varphi \mid \varphi \text{ is a satisfiable Boolean formula in 3-CNF}\}$$

3-CNF: conjunction of disjunctions with ≤ 3 literals per clause.

$$\underbrace{(X_1 \vee X_3)}_{\text{clause}} \wedge (\neg X_1 \vee \neg X_2 \vee \overbrace{\neg X_3}^{\text{literal}}) \wedge (\neg X_1 \vee X_2 \vee X_4)$$

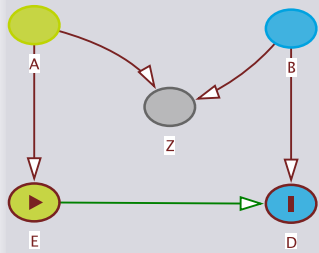
Inference as a Decision Problem

Our goal is to show that exact inference in Bayes nets is NP-hard. Our first step is to formulate it as a decision problem.

$$\text{IBN} = \{(G, x) \mid G = (\mathbf{V}, \mathbf{E}, P) \text{ is a Bayes net}, x \in \mathbf{V}, P(x) > 0\}$$

We will build a function r that takes as input a 3-CNF formula φ and outputs a Bayesian network G_φ and a variable (name) x_ϕ such that

$$\varphi \in \text{3-SAT} \Leftrightarrow r(\varphi) = (G_\varphi, x_\varphi) \in \text{IBN}$$



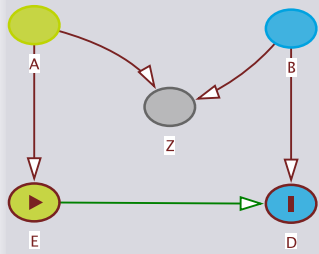
Translating the Formula into a Network

Our network will consist of three layers:

1. The variable layer (one node per variable).
2. The clause layer (one node per clause).
3. The formula layer (one node).

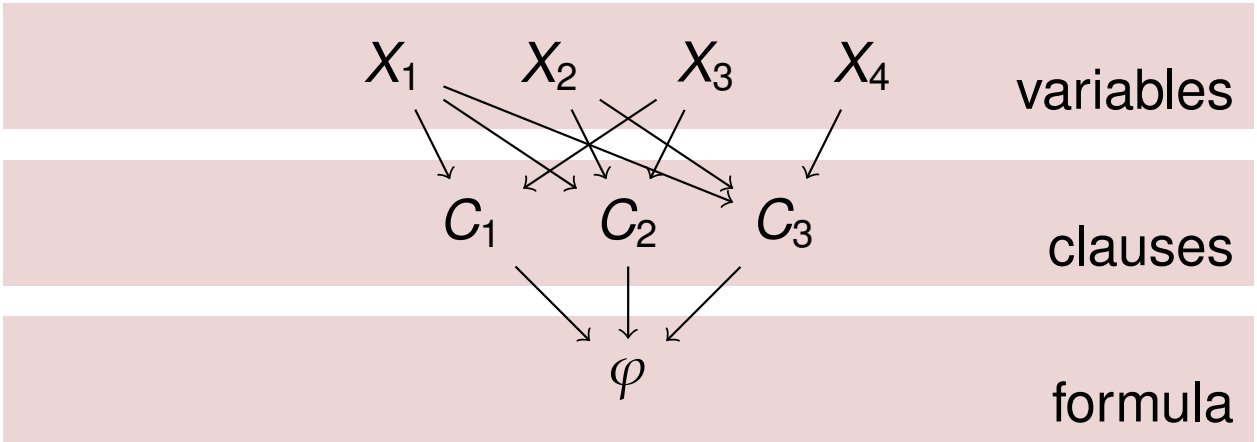
$X_1 \quad X_2 \quad X_3 \quad X_4$	variables
$C_1 \quad C_2 \quad C_3$	clauses
φ	formula

$$\varphi = \underbrace{(X_1 \vee X_3)}_{C_1} \wedge \underbrace{(\neg X_1 \vee \neg X_2 \vee \neg X_3)}_{C_2} \wedge \underbrace{(\neg X_1 \vee X_2 \vee X_4)}_{C_3}$$

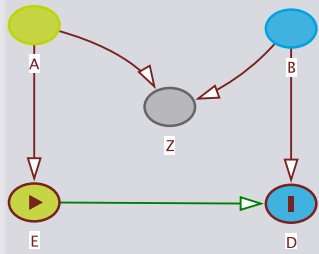


Connecting Nodes in the Formula Network

- Draw an edge from each variable to all clauses in which that variable appears.
- Draw an edge from each clause to the formula node.



$$\varphi = \underbrace{(X_1 \vee X_3)}_{C_1} \wedge \underbrace{(\neg X_1 \vee \neg X_2 \vee \neg X_3)}_{C_2} \wedge \underbrace{(\neg X_1 \vee X_2 \vee X_4)}_{C_3}$$



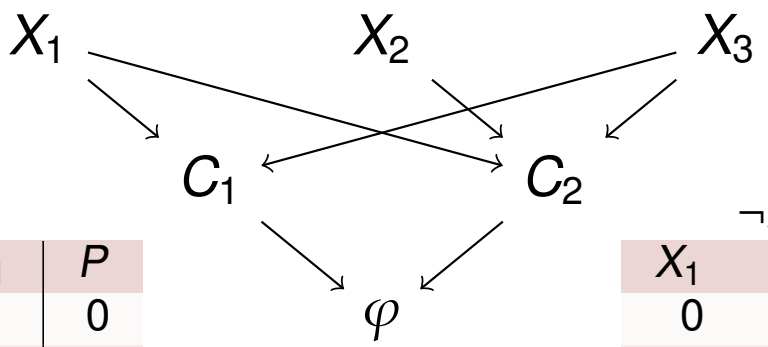
Probability Tables for the Formula Network

- Annotate each variable with probability 0.5.
- Annotate each clause node with a probability table that encodes this clause.
- Annotate the formula node with a probability table that encodes a logical “and”.

X_1	P
1	0.5

X_2	P
1	0.5

X_3	P
1	0.5



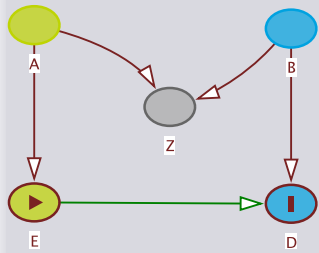
$X_1 \vee X_3$

X_1	X_3	C_1	P
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

C_1	C_2	φ	P
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

$\neg X_1 \vee \neg X_2 \vee \neg X_3$

X_1	X_2	X_3	C_1	P
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0



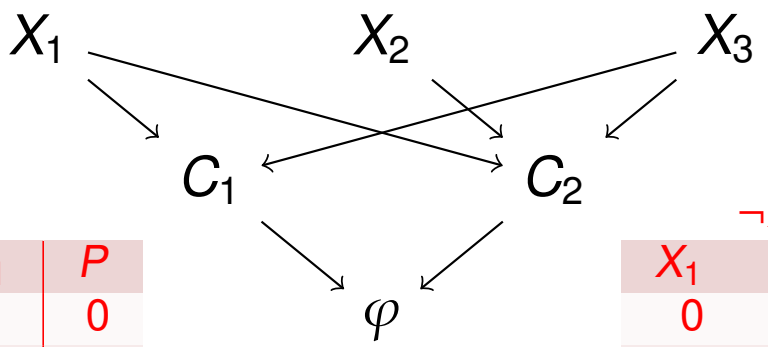
Probability Tables for the Formula Network

- Annotate each variable with probability 0.5.
- Annotate each clause node with a probability table that encodes this clause.
- Annotate the formula node with a probability table that encodes a logical “and”.

X_1	P
1	0.5

X_2	P
1	0.5

X_3	P
1	0.5



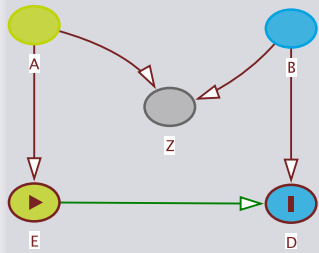
$X_1 \vee X_3$

X_1	X_3	C_1	P
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

C_1	C_2	φ	P
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

$\neg X_1 \vee \neg X_2 \vee \neg X_3$

X_1	X_2	X_3	C_1	P
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0



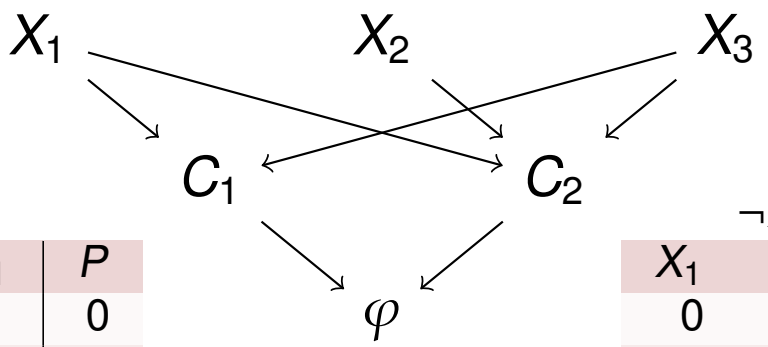
Probability Tables for the Formula Network

- Annotate each variable with probability 0.5.
- Annotate each clause node with a probability table that encodes this clause.
- Annotate the formula node with a probability table that encodes a logical “and”.

X_1	P
1	0.5

X_2	P
1	0.5

X_3	P
1	0.5



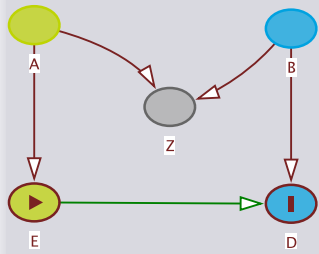
$X_1 \vee X_3$

X_1	X_3	C_1	P
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

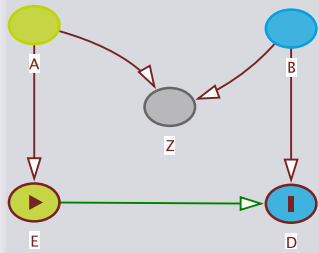
C_1	C_2	φ	P
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

$\neg X_1 \vee \neg X_2 \vee \neg X_3$

X_1	X_2	X_3	C_1	P
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	0



5-Minute Exercise



Exercise

Draw the network $r(\varphi)$ for the following 2-SAT formula:

$$\varphi = (X_1 \vee X_2) \wedge (X_1 \vee \neg X_2)$$

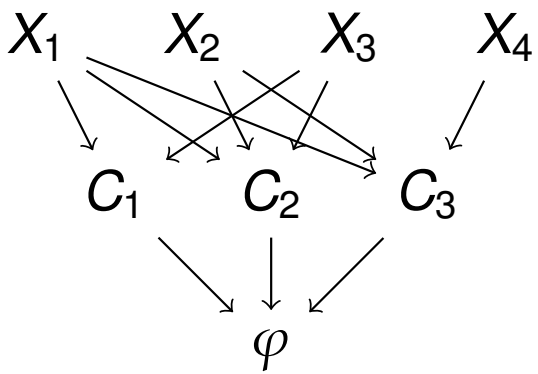
Compute the associated marginal probability $P(\varphi)$.

Probability Tables for the Formula Network

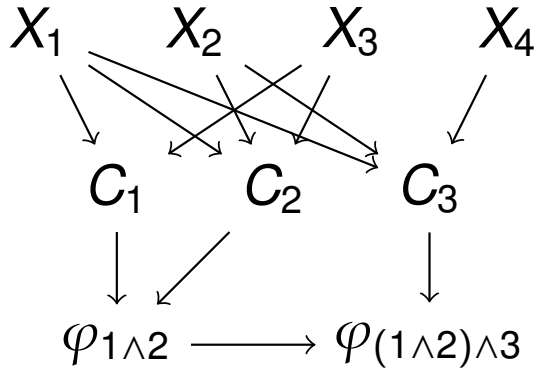
Network size for formula φ with n variables and k clauses:

- n variable nodes $\Rightarrow$ n parameters.
- k clause nodes, ≤ 3 parents
  $\Rightarrow \leq 8k$ parameters.
- 1 formula nodes, k parents $\Rightarrow 2^k$ parameters.

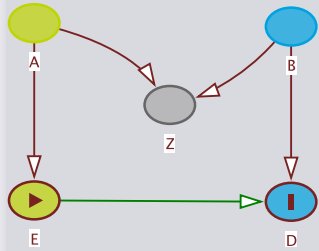
The network still has exponential size! Solution: compute the "and" function in $k-1$ steps instead of a single step. Each step is a function with 2 parameters.



$n + 8k + 2^k$



$n + 8k + 4(k-1)$



Completing the Proof

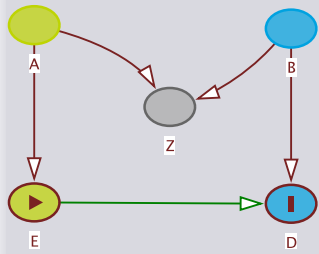
Given a 3CNF-formula φ with n variables and k clauses, we can run the following algorithm to determine if ϕ is satisfiable:

1. Construct the Bayesian network $r(\varphi)$, which has $n + k + (k - 1)$ nodes, $3k + 2(k - 1)$ edges, and $n + 8k + 4(k - 1)$ probability table entries.
2. Determine whether $r(\varphi) \in \text{IBN}$.
3. φ is satisfiable $\Leftrightarrow$ $r(\varphi) \in \text{IBN}$.

Therefore, exact inference in Bayesian networks can simulate Boolean satisfiability with polynomial time overhead.

$$\text{3-SAT} \leq_P \text{IBN}$$

Since 3-SAT is NP-hard, this means that IBN is also NP-hard.



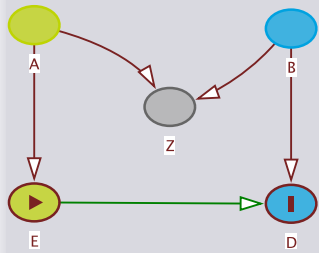
Conclusions

Take-home message from complexity analysis

Even for "sparse" Bayes nets ($\leq$ 3 parents), computing marginal probabilities remains a hard problem – no polynomial time algorithm is known, and perhaps none exists.

This leaves us with 3 options to approach inference:

- Use brute-force (expontential time) algorithms.
- Try to find efficient algorithms for retricted subclasses of Bayes nets.
- Use approximation algorithms.



Naive Marginalization

Consider a Markov chain with n binary variables (below, $n = 4$):

$$X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_4$$

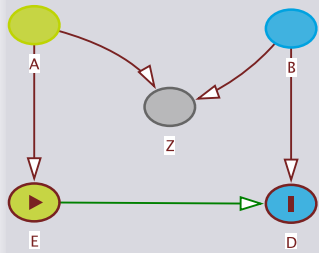
To compute $P(x_n)$, we need to sum over all $x_1, \ldots, x_{n-1}$:

$$P(x_4) = \sum_{x_1, x_2, x_3} P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_2)P(x_4 \mid x_3)$$

For example,

$$\begin{aligned}
P(X_4 = 1) = &P(X_1 = 0)P(X_2 = 0 \mid X_1 = 0)P(X_3 = 0 \mid X_2 = 0)P(X_4 = 1 \mid X_3 = 0) \\
&+P(X_1 = 1)P(X_2 = 0 \mid X_1 = 1)P(X_3 = 0 \mid X_2 = 0)P(X_4 = 1 \mid X_3 = 0) \\
&+P(X_1 = 0)P(X_2 = 1 \mid X_1 = 0)P(X_3 = 0 \mid X_2 = 1)P(X_4 = 1 \mid X_3 = 0) \\
&+P(X_1 = 1)P(X_2 = 1 \mid X_1 = 1)P(X_3 = 0 \mid X_2 = 1)P(X_4 = 1 \mid X_3 = 0) \\
&+P(X_1 = 0)P(X_2 = 0 \mid X_1 = 0)P(X_3 = 1 \mid X_2 = 0)P(X_4 = 1 \mid X_3 = 1) \\
&+P(X_1 = 1)P(X_2 = 0 \mid X_1 = 1)P(X_3 = 1 \mid X_2 = 0)P(X_4 = 1 \mid X_3 = 1) \\
&+P(X_1 = 0)P(X_2 = 1 \mid X_1 = 0)P(X_3 = 1 \mid X_2 = 1)P(X_4 = 1 \mid X_3 = 1) \\
&+P(X_1 = 1)P(X_2 = 1 \mid X_1 = 1)P(X_3 = 1 \mid X_2 = 1)P(X_4 = 1 \mid X_3 = 1)
\end{aligned}$$

A naive evaluation like this requires adding 2^{n-1} terms!



The Distributive Law

By using the distributive law, we can save a lot of work.

$$P(x_4) = \sum_{x_3} P(x_4 \mid x_3) \left(\sum_{x_2} P(x_3 \mid x_2) \left(\sum_{x_1} P(x_2 \mid x_1) P(x_1) \right) \right)$$

$$X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_4$$

X_1	P
1	0.5

X_1	X_2	P
0	1	0.4
1	1	0.4

X_2	X_3	P
0	1	0.8
1	1	0.6

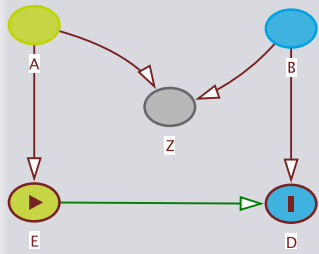
X_3	X_4	P
0	1	0.2
1	1	0.5

Consider the innermost term:

$$\sum_{x_1} P(x_2 \mid x_1) P(x_1) = P(x_2 \mid X_1 = 0) P(X_1 = 0) + P(x_2 \mid X_1 = 1) P(X_1 = 1)$$

Each summand is a function of X_2, can be written as vector:

$$\sum_{x_1} P(x_2 \mid x_1) P(x_1) = 0.5 \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}_{X_2} + 0.5 \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}_{X_2} = \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}_{X_2}$$



The Distributive Law

Now let us go one step further:

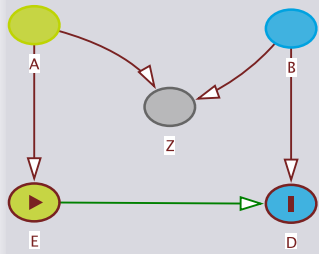
$$\sum_{X_2} \underbrace{P(x_3 \mid x_2)}_{\text{function of } X_2, X_3} \underbrace{\left(\sum_{x_1} P(x_2 \mid x_1) P(x_1) \right)}_{\text{function of } X_2}$$

If we represent $P(x_3 \mid x_2)$ as a matrix

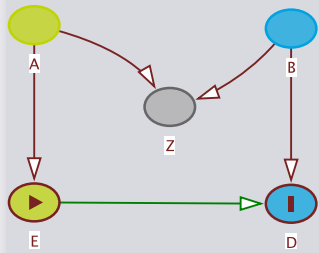
$$P(x_3 \mid x_2) = \begin{pmatrix} P(X_3 = 0 \mid X_2 = 0) & P(X_3 = 0 \mid X_2 = 1) \\ P(X_3 = 1 \mid X_2 = 0) & P(X_3 = 1 \mid X_2 = 1) \end{pmatrix}$$

then we can write the term above as a matrix-vector-product:

$$\sum_{X_2} P(x_3 \mid x_2) \left(\sum_{x_1} P(x_2 \mid x_1) P(x_1) \right) = \begin{pmatrix} 0.2 & 0.4 \\ 0.8 & 0.6 \end{pmatrix}_{X_3, X_2} \times \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}_{X_2} = \begin{pmatrix} 0.32 \\ 0.68 \end{pmatrix}_{X_3}$$



The Distributive Law



Now let us apply this repeatedly to our Markov chain:

$$X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_4$$

X_1	P
1	0.5

X_1	X_2	P
0	1	0.4
1	1	0.4

X_2	X_3	P
0	1	0.8
1	1	0.6

X_3	X_4	P
0	1	0.2
1	1	0.5

We get:

$$P(x_4) = \sum_{x_3} P(x_4 \mid x_3) \left(\sum_{x_2} P(x_3 \mid x_2) \left(\sum_{x_1} P(x_2 \mid x_1) P(x_1) \right) \right)$$

$$= \begin{pmatrix} 0.8 & 0.5 \\ 0.2 & 0.5 \end{pmatrix}_{X_4,X_3} \times \left(\begin{pmatrix} 0.2 & 0.4 \\ 0.8 & 0.6 \end{pmatrix}_{X_3,X_2} \times \begin{pmatrix} 0.4 \\ 0.6 \end{pmatrix}_{X_2} \right)$$

This gives us $n-1$ matrix-vector multiplications instead of 2^n additions.

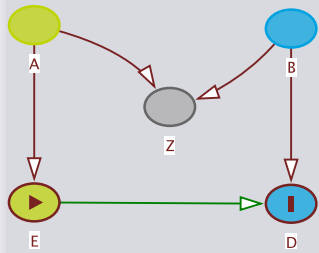
Elimination Orderings

What did we just do and why did this work? Let us look again at our expression:

$$P(x_4) = \underbrace{\sum_{x_3}}_{\substack{\text{eliminate } X_3 \\ \text{keep } X_4}} P(x_4 \mid x_3) \underbrace{\left(\sum_{x_2}}_{\substack{\text{eliminate } X_2 \\ \text{keep } X_3}} P(x_3 \mid x_2) \underbrace{\left(\sum_{x_1}}_{\substack{\text{eliminate } X_1 \\ \text{keep } X_2}} P(x_2 \mid x_1) P(x_1)\right)\right)$$

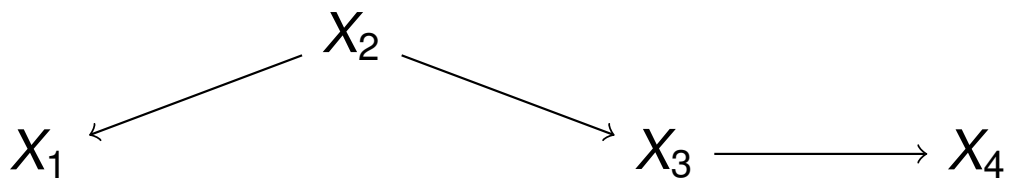
We ordered our factorization such that in each step, we eliminate one variable and keep a function of only one variable. This requires that the factors be ordered such that, for instance X_1 does not occur "before" the expression $\sum_{X_1}$. Such an ordering of a factorization is called an elimination ordering.

How can we find elimination orderings for other networks than Markov chains?



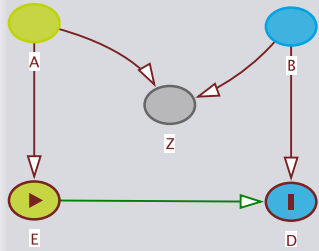
Elimination Ordering of a Fork

Let us now look at a slightly different network:

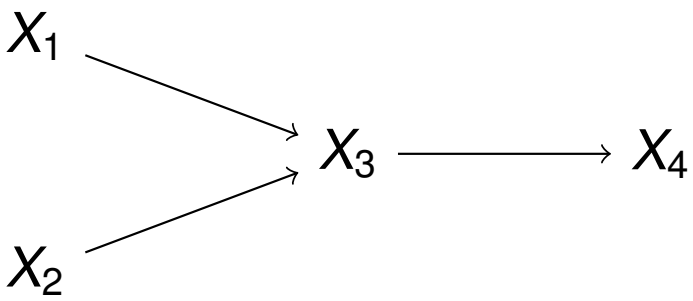


$$
\begin{aligned}
P(x_4) &= \sum_{x_1,x_2,x_3} P(x_1 \mid x_2)P(x_2)P(x_3 \mid x_2)P(x_4 \mid x_3) \\
&= \sum_{x_3} \left(P(x_4 \mid x_3) \left(\sum_{x_2} P(x_2)P(x_3 \mid x_2) \left(\sum_{x_1} P(x_1 \mid x_2) \right) \right) \right) \\
&= \sum_{x_3} \left(P(x_4 \mid x_3) \left(\sum_{x_2} P(x_2)P(x_3 \mid x_2) \times \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{X_2} \right) \right)
\end{aligned}
$$

Importantly, the term $P(x_2)P(x_3 \mid x_2)$ can again be represented as a 2×2 matrix with X_2 in the columns and X_3 in the rows.



Elimination Orderings with Multiple Parents

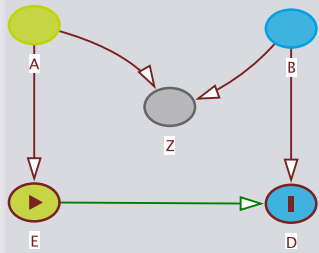


$$P(x_4) = \sum_{x_1,x_2,x_3} P(x_1)P(x_2)P(x_3 \mid x_1, x_2)P(x_4 \mid x_3)$$
$$= \sum_{x_3} P(x_4 \mid x_3) \sum_{x_1,x_2} P(x_1)P(x_2)P(x_3 \mid x_1, x_2)$$

The inner term cannot be simplified further. It can be written as a product $M \times v_1 \times v_2$, where M is a 3D Matrix and v_1 and v_2 are column vectors. The result is again a column vector.

Parent complexity

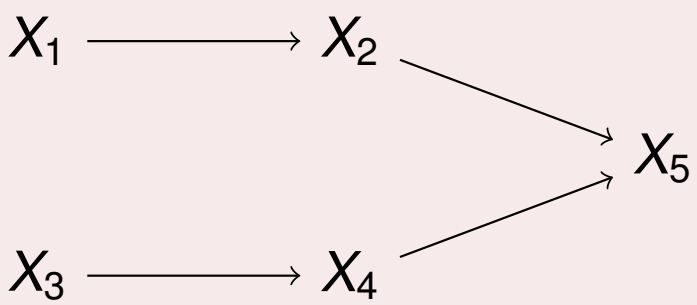
For nodes with k parents, we need to eliminate $k - 1$ variables at the same time. For binary variables, this requires a k-dimensional matrix with 2^k entries.

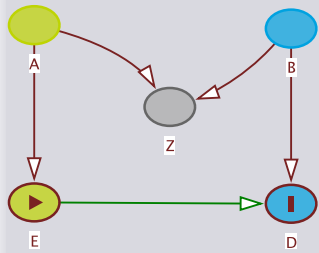


5-Minute Exercise

Exercise

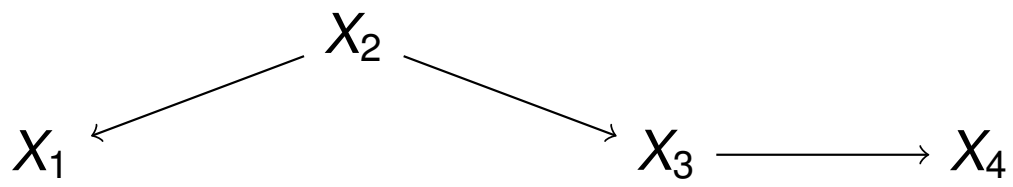
Find an elimination ordering for $P(x_5)$ in the following network:





Elimination Orderings and *d*-Separation

Let us revisit the fork elimination ordering:

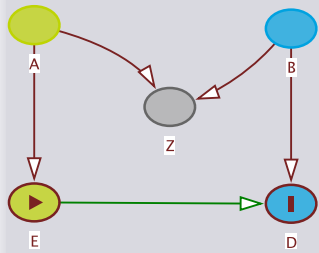


$$P(x_4) = \underbrace{\sum_{x_3}}_{X_4 \perp\!\!\!\perp X_1, X_2 | X_3} \Big(P(x_4 \mid x_3) \Big(\underbrace{\sum_{x_2}}_{X_3, X_4 \perp\!\!\!\perp X_1 | X_2} P(x_2) P(x_3 \mid x_2) \Big(\sum_{x_1} P(x_1 \mid x_2) \Big) \Big) \Big)$$

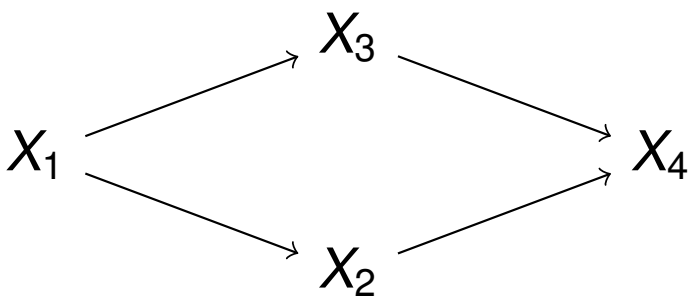
In each step, the network is cut into three parts:

- An inner part X_i.
- An outer part X_o.
- A variable X such that $P(x_o \mid x, x_i) = P(x_o \mid x)$

This is only possible if X *d*-separates X_i and X_o.

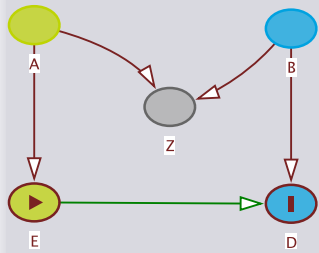


Elimination Orderings and Loops



$$P(x_4) = \sum_{x_1,x_2,x_3} P(x_1)P(x_2 \mid x_1)P(x_3 \mid x_1)P(x_4 \mid x_3, x_2)$$

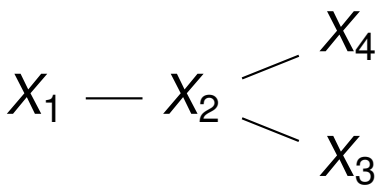
No elimination ordering can be found because there is no variable that cuts the network in two different parts!



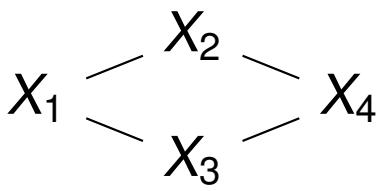
Trees and Polytrees

Definition

A tree is an undirected graph in which there is one, and only one, path between any pair of nodes.



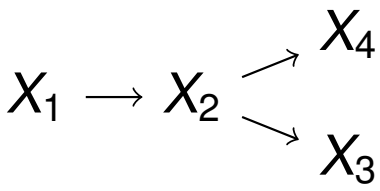
a tree

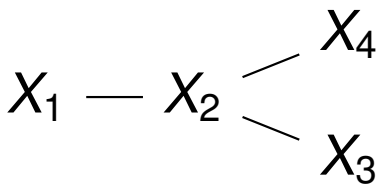


not a tree

Definition

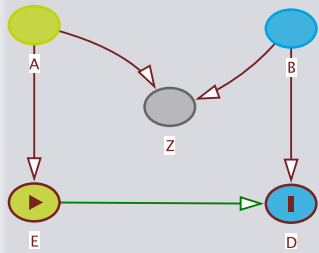
The skeleton of a directed graph is the same graph in which all directed edges are replaced by undirected edges.





Definition

A polytree is a directed graph whose skeleton is a tree.



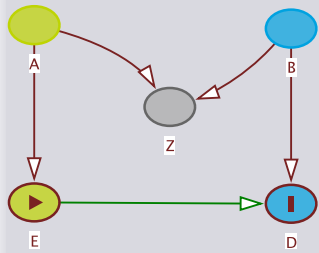
Elimination Orderings in Polytrees

For polytree networks, an efficient elimination ordering is guaranteed to exist: At every node X, we have that

$$\mathrm{Ch}^*(X) \perp\!\!\!\perp \mathrm{Pa}^*(X) \mid X \,,$$

where $\mathrm{Ch}^*(X)$ is the component of the network containing the children of X, and $\mathrm{Pa}^*(X)$ is the component of the network containing the parents of X.

(There can be no path from an ancestor of X to a descendant of X that does not pass through X).



Summary

1. Exact inference in Bayesian networks is NP-hard.
2. But for some subclasses of Bayesian networks, efficient exact inference algorithms do exist.
3. Elimination orderings are one approach for efficient exact inference, and they work in polytrees.