Directed Graphical Models

Felipe José Bravo Márquez

September 13, 2021

Directed Graphical Models

- Probabilistic graphical models (PGMs) provide a visual representation of the underlying structure of a joint probablity distribution [Ruozzi,].
- In this class we will focus on directed graphical models (DGMs), which are one type of PGM.
- Directed graphical models (DGMs) are a family of probability distributions that admit a compact parametrization that can be naturally described using a **directed acyclic graph**.
- DGMs were created by Turig prize awardee Judea Pearl under the name of **Bayesian networks**.
- We won't use that term much in this class because statistical inference for DGMs can be performed using frequentist or Bayesian methods, which makes the name misleading [Wasserman, 2013].

Judea Pearl

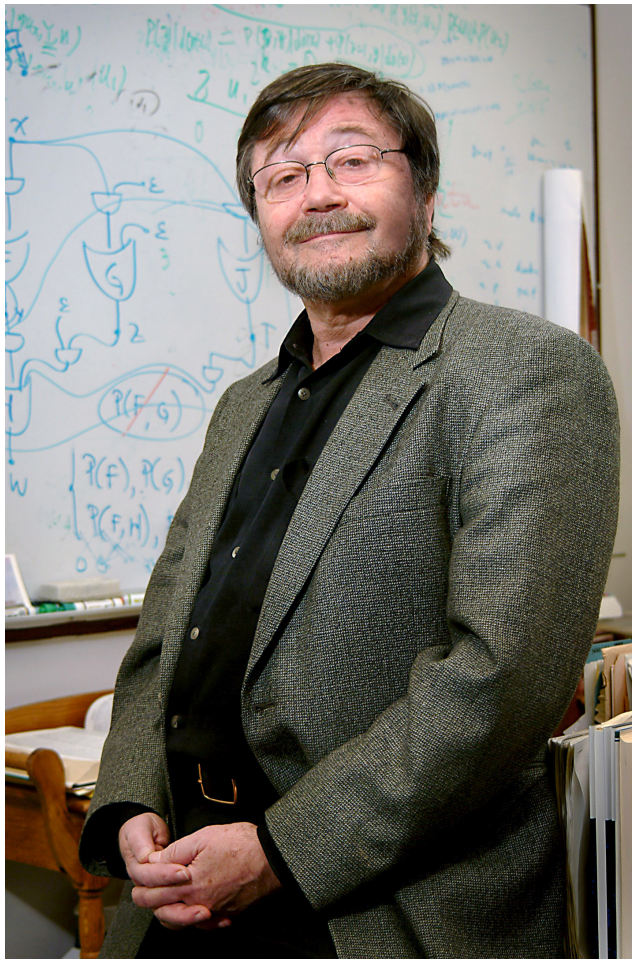


Figure: Judea Pearl, source:
https://www.amacad.org/person/judea-pearl

Directed Graphical Models

- DGMs link two very different branches of mathematics: probability and graph theory.
- They also have intriguing connections to philosophy, particularly the question of **causality**.
- At the same time, they are widely used in statistics and machine learning.
- DGMs can be used to solve problems in fields as diverse as medicine, language processing, vision, and many others [Ermon and Kuleshov,].
- But before introducing them more formally we need to learn the following mathematical concepts:
  1. Conditional independence
  2. The chain rule of probability
  3. Directed acyclical graphs (DAGs)

Conditional Independence

- Two random variables X and Y are independent, written $X \perp Y$ if $f(x, y) = f(x)f(y)$ for all values of x, y.
- This also implies that $f(x|y) = f(x)$ and $f(y|x) = f(y)$.
- Notice that f can be either a density function for continous random variables or a probability mass function for discrete random variables.
- In the same way $f(x|y)$ and $f(y|x)$ correspond to conditional densities or mass functions.
- Now, suppose we have three random variables A, B, C.
- A and B are conditionally independent given C, written $A \perp B|C$, if:

$$f(a|b, c) = f(a|c)$$

for all a, b and c.

Conditional Independence

- Intuitively, $A \perp B|C$ means that, once you know C, B provides no extra information about A.
- Notice that $A \perp B|C$ doesn't necessarily imply that $A \perp B$.

Example

- Let H, V, A be three random variables representing a person's height, vocabulary and age.
- H and V are dependent $f(v|h) \neq f(v)$ since very small people tend to be children, known for their more basic vocabularies.
- But knowing that two people are 19 years old (i.e., conditional on age) there is no reason to think that one person's vocabulary is larger if we are told that they are taller:

$$f(v|h,a) = f(v|a) \Leftrightarrow V \perp H|A$$

The chain rule of probability

- For a set of random variables $X_1, \ldots, X_n$, the chain rule of probability allow us to express the joint probability function $f(x_1, x_2, \ldots, x_n)$ as a product of n conditional probabilities:

$$f(x_1, x_2, \ldots, x_n) = f(x_1)f(x_2|x_1)f(x_3|x_2, x_1)\ldots f(x_n|x_{n-1}, \ldots, x_2, x_1).$$

- For example for the case of $n = 3$

$$f(x_1, x_2, x_3) = f(x_1)f(x_2|x_1)f(x_3|x_2, x_1)$$

using the definition of conditional probabilities we have that

$$f(x_2|x_1) = f(x_1, x_2)/f(x_1)$$

and

$$f(x_3|x_2, x_1) = f(x_1, x_2, x_3)/f(x_1, x_2)$$

- By replacing these expressions into the chain of products, many expressions cancel out and we obtain $f(x_1, x_2, x_3)$.

Joint Probabilities

- Expressing a joint probability function can be expensive.
- For example if there are m binary random variables, the complete distribution is specificied by $2^m - 1$ joint probabilities.
- For example, if we have two Boolean variables (A, B), we need the probabilities $\mathbb{P}(A, B)$, $\mathbb{P}(\neg A, B)$, $\mathbb{P}(A, \neg B)$, and $\mathbb{P}(\neg A, \neg B)$.
- A joint distribution for a set of random variables gives all the information there is about the distribution.
- Note that for m boolean variables, the joint distribution contains 2^m values.
- However, the sum of all the joint probabilities must be 1 because the probability of all possible outcomes must be 1.
- Thus, to specify the joint distribution, one needs to specify 2^{m-1} numbers.

Joint Probabilities

- The main idea of DGMs is that by assuming that some variables are conditionally independent we can use the probability chain rule to represent the joint distribution more compactly.
- For example in the previous example, a conditional independence assumption could be: $X_3 \perp X_2 | X_1$: $f(x_3|x_2, x_1) = f(x_3|x_2)$.
- Then our joint distribution would be reduced to:

$$f(x_1, x_2, x_3) = f(x_1)f(x_2|x_1)f(x_3|x_2)$$

- In the following slides we will clarify these concepts with the Student example given in [Koller and Friedman, 2009].

The Student Example

- Consider the problem faced by a company trying to hire a recent college graduate.
- The company's goal is to hire intelligent employees, but there is no way to test intelligence directly.
- However, the company has access to the student's SAT scores[1], which are informative but not fully indicative.
- Thus, our probability space is induced by the two random variables Intelligence (I) and SAT (S).
- For simplicity, we assume that each of these takes two values: $Val(I) = \{i_1, i_0\}$, which represent the values high intelligence (i_1) and low intelligence (i_0).

[1] The SAT is a standardized test widely used for college admissions in the United States.

The Student Example

- Similarly $Val(S) = \{s_1, s_0\}$, which also represent the values high (score) and low (score), respectively.
- Thus, our joint probability function in this case has four entries.
- For example, one possible joint probability function f would be:

I	S	$f(i,s)$
i^0	s^0	0.665
i^0	s^1	0.035
i^1	s^0	0.06
i^1	s^1	0.24

- Notice that we would need 3 parameters to specify this joint probability function (2^{m-1}).
- Alternatively, we could use the the chain rule of conditional probabilities to represent the joint probability function as follows:

$$f(i,s) = f(i)f(s|i)$$

The Student Example

- Other factorizations obtained by changing the order of the variables are also valid (e.g., $f(i,s) = f(s)f(i|s)$).
- However, it is convenient to represent the process in a way that is more compatible with causality.
- Various factors (e.g., genetics, upbringing) first determined (stochastically) the student's intelligence.
- Her/his performance on the SAT is determined (stochastically) by her/his intelligence.
- We note that the models we construct are not required to follow causal intuitions, but they often do.

The Student Example

- From a mathematical perspective, $f(i,s) = f(i)f(s|i)$ leads to an alternative way of representing the joint probability function.
- Instead of specifying the various joint entries $f(i,s)$, we would specify it in the form of $f(i)$ and $f(s|i)$.
- Thus, we could represent the joint probability function using two tables.
- The first one representing the marginal probability function of I.

i^0	i^1
0.7	0.3

- These numbers can be easily verified from the previous table. For example

$$f(i^0) = f(i^0, s^0) + f(i^0, s^1) = 0.665 + 0.035 = 0.7$$

The Student Example

- The second table represents the conditional probability function of S given I:

I	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

- which can also be verified from previous table using the Bayes theorem. For example:

$$f(s^0|i^0) = f(s^0, i^0)/f(i^0) = 0.665/0.7 = 0.95$$

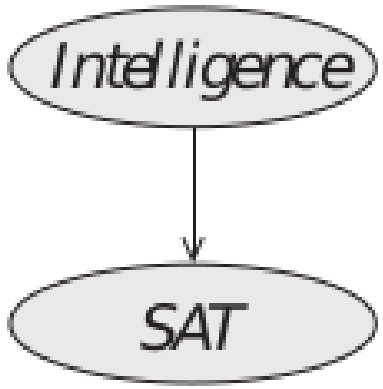
- The conditional probability function $f(s|i)$ represents the probability that the student will succeed on his SATs in the two possible cases:
  1. The case where the student's intelligence is low.
  2. The case where it is high.

The Student Example

- The conditional probability function asserts that a student of low intelligence is extremely unlikely to get a high SAT score ($f(s_1|i_0) = 0.05$).
- On the other hand, a student of high intelligence is likely, but far from certain, to get a high SAT score ($f(s_1|i_1) = 0.8$).
- It is instructive to consider how we could parameterize this alternative representation.
- Here, we are using three Bernoulli distributions, one for $f(i)$, and two for $f(s|i_0)$ and $f(s|i_1)$.
- Hence, we can parameterize this representation using three independent parameters, say θ_{i^1} ,$\theta_{s^1|i^1}$, and $\theta_{s^1|i^0}$
- Recall that the original representation also required 3 parameters.

The Student Example

- Thus, although the conditional representation is more natural than the explicit representation of the joint, it is not more compact.
- However, as we will soon see, the conditional parameterization provides a basis for our compact representations of more complex distributions.
- Although we haven't defined directed acyclical graphs (DAGs) yet, it is instructive to see how this example would be represented as one.



- The DAG from above has a node for each of the two random variables I and S, with an edge from I to S representing the direction of the dependence in this model.

The Student Example

- Let's assume now that the company also has access to the student's grade G in some course.
- In this case, our probability space is the joint probability function over the three relevant random variables I, S, and G.
- We will assume that I and S are as before, and that G takes on three values g^1, g^2, g^3, representing the grades A, B, and C, respectively.
- So, the joint probability function has twelve entries.
- We can see that for any reasonable joint probability function f, there are no independencies that hold.
- The student's intelligence is clearly correlated both with SAT score and grade.

The Student Example

- The SAT score and grade are also not independent.
- A high SAT score increases the chances of getting a high grade.
- Thus, we may assume that, for our particular probability function f, $f(g^1|s^1) > f(g^1|s^0)$.
- However, it is quite plausible that our probability function f in this case satisfies a conditional independence property.
- If we know that the student has high intelligence, a high grade on the SAT no longer gives us information about the student's performance in the class.
- More formally: $f(g|s,i) = f(g|i)$ or $G \perp S|I$
- So the joint probability function can be reduced from

$$f(i,s,g) = f(i)f(s|i)f(g|i,s)$$

to

$$f(i,s,g) = f(i)f(s|i)f(g|i)$$

The Student Example

- Now our joint probability function is specified by $f(i)$, $f(s|i)$, and $f(g|i)$.
- While $f(i)$, $f(s|i)$ might be the same as before, and $f(g|i)$ might be:

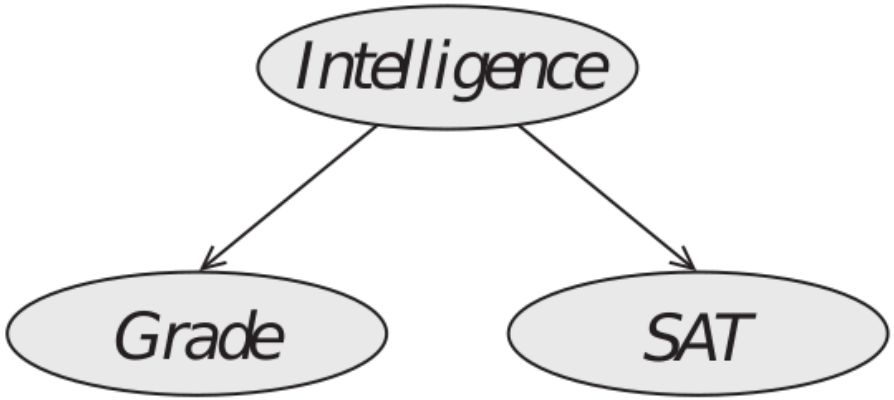
I	g^1	g^2	g^3
i^0	0.2	0.34	0.46
i^1	0.74	0.17	0.09

- Now we can calculate the joint probability of any combination of inputs.
- For example:

$$
\begin{aligned}
f(i^1, s^1, g^2) &= f(i^1) f(s^1|i^1) f(g^2|i^1) && (1)\\
&= 0.3 * 0.8 * 0.17 && (2)\\
&= 0.0408 && (3)
\end{aligned}
$$

The Student Example

- The corresponding DAG would be as follows:



- In this case, the alternative parameterization is more compact than the joint.
- We now have three Bernoulli distributions $f(i)$, $f(s|i^1)$ and $f(s|i^0)$, and two three-valued categorical distributions $f(g|i_1)$ and $f(g|i^0)$.
- A categorical distribution is discrete probability distribution that describes the possible results of a random variable that can take on one of K possible categories, with the probability of each category separately specified[2].

[2] https://en.wikipedia.org/wiki/Categorical_distribution

The Student Example

- Each of the Bernoullis requires one independent parameter, and each three-valued categorical requires two independent parameters, for a total of seven.
- By contrast, our joint probability function has twelve entries, so that eleven independent parameters are required to specify an arbitrary joint probability function over these three variables.
- It is important to note another advantage of this way of representing the joint: modularity.
- When we added the new variable G, the joint probability function changed entirely.
- Had we used the explicit representation of the joint, we would have had to write down twelve new numbers.
- In the factored representation, we could reuse our local probability models for the variables I and S, and specify only the probability model for G, the conditional probability function $f(G|I)$.
- This property will turn out to be invaluable in modeling real-world systems.

Directed Acyclic Graphs (DAGs)

- We just learned how the chain rule of probability and conditional independence assumptions enable us to obtain a compact representation of a joint probability function.
- Now we are in a position to introduce DAGs.
- A directed graph consists of a set of nodes with arrows between some nodes.
- More formally, a directed graph G consists of a set of vertices V and an edge set E of ordered pairs of vertices.
- If $(Y, X) \in E$ then there is an arrow pointing from Y to X.

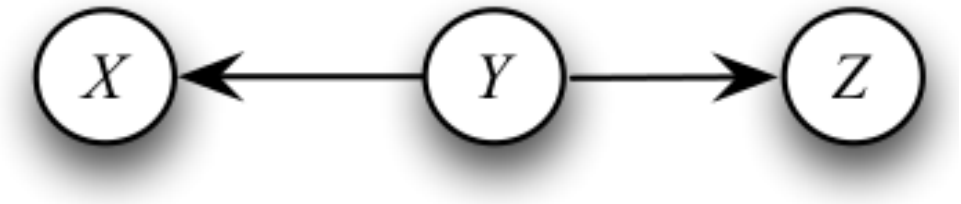


Figure: A directed graph with vertices $V = \{X, Y, Z\}$ and edges $E = \{(Y, X), (Y, Z)\}$.

Directed Acyclic Graphs (DAGs)

- If an arrow connects two variables X and Y (in either direction) we say that X and Y are **adjacent**.
- If there is an arrow from X to Y then X is a **parent** of Y and Y is a **child** of X.
- The set of all parents of X is denoted by π_X or $\pi(X)$.
- A **directed path** between two variables is a set of arrows all pointing in the same direction linking one variable to the other such as the chain shown below:

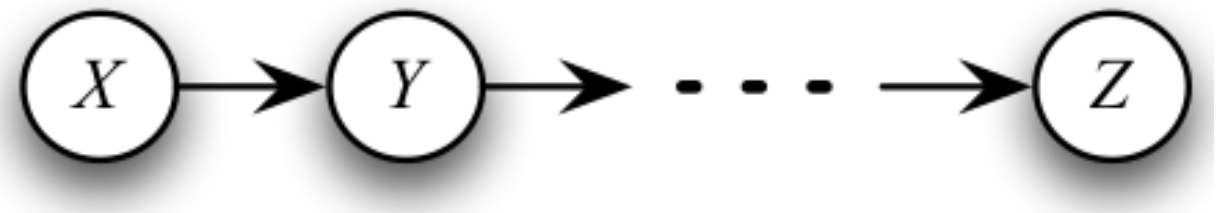


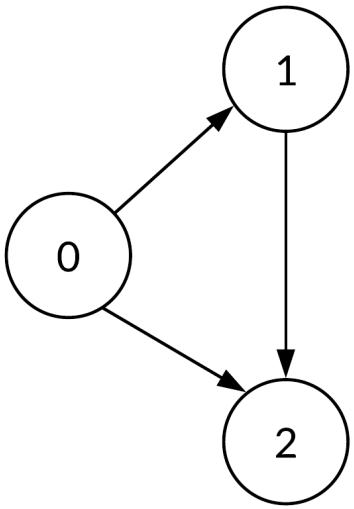
Figure: A chain graph with a directed path.

- A sequence of adjacent vertices starting with X and ending with Y but ignoring the direction of the arrows is called an **undirected path**.
- X is an **ancestor** of Y if there is a directed path from X to Y (or X = Y).
- We also say that Y is a **descendant** of X.

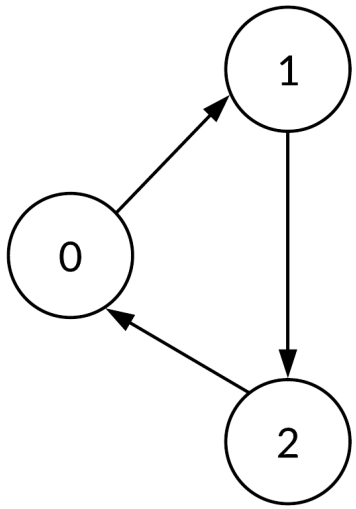
Directed Acyclic Graphs (DAGs)

- A directed path that starts and ends at the same variable is called a cycle.
- A directed graph is acyclic if it has no cycles.
- In this case we say that the graph is a directed acyclic graph or DAG.

Acyclic Graph



Cyclic Graph



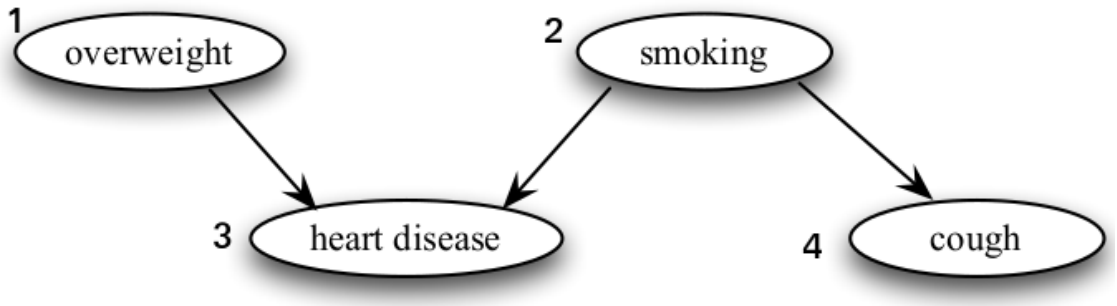
- From now on, we only deal with directed acyclic graphs since it is very difficult to provide a coherent probability semantics over graphs with directed cycles.

Probability and DAGs

- When a DAG is used to encode a joint probability function f (density or mass) with given conditional independence relations, we have a Directed Graphical Model or Bayesian Network.
- Warning: It is common to find terms DAG, DGM or Bayesian network used interchangeably in the literature.
- Let G be a DAG with vertices $V = (X_1, \ldots, X_i, \ldots, X_m)$.
- Each vertex X_i is random variable
- The edges of G represent (conditional) independence relations between variables.
- The order of vertices $(1, \ldots, i, \ldots, m)$ forms a topological sort on the random variables.
- This means that every variable comes before all its descendants in the graph.

Probability and DAGs

- For example, the next figure shows a DAG with four variables, the numbers indicate the topological order.



- It is clear from the figure, that no vertex comes before any of its descendant in the DAG.
- Let f be probability function for V, $f(x_1, \ldots, x_d)$, we say that G represents f, if

$$f(x) = \prod_{j=1}^{d} f(x_j|\pi_{x_j}) \tag{4}$$

where π_{x_j} is the set of parent nodes of X_j

Probability and DAGs

- Based on the equation above, probability function f of the previous graph takes the following decomposition:

$$f(o, s, h, c) = f(o)f(s)f(h|o, s)f(c|s).$$

- If no conditional independence assumptions were made, the joint probability function f would the following:

$$f(o, s, h, c) = f(o)f(s|o)f(h|o, s)f(c|o, s, h)$$

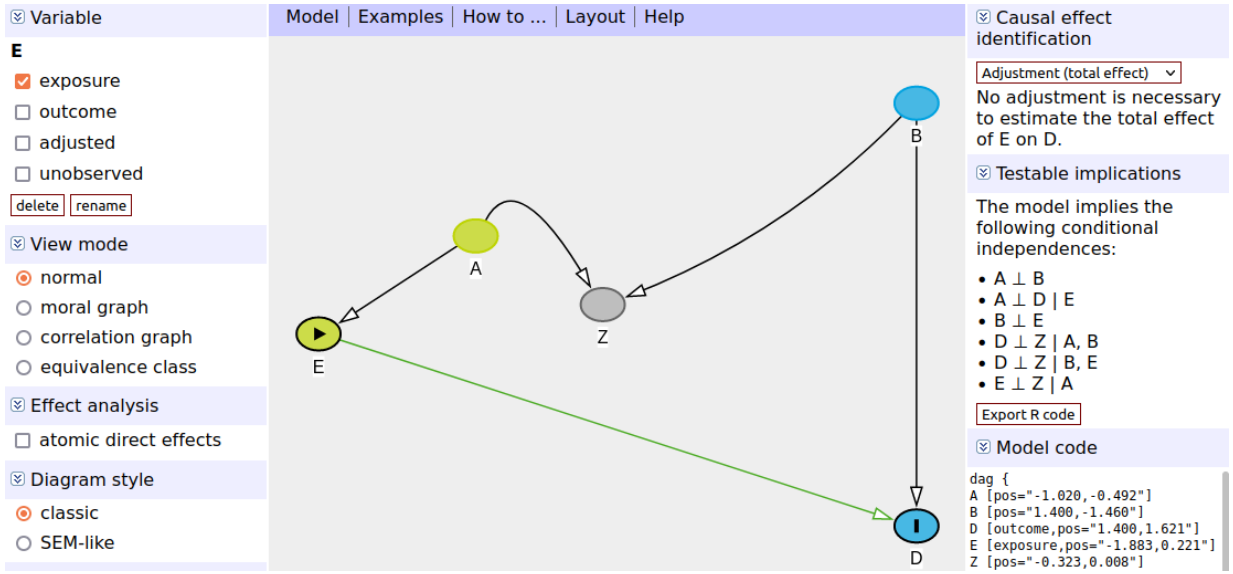
- That means that the graph above encodes the following independence asssumptions:
  1. $f(s|o) = f(s)$
  2. $f(c|o, s, h) = f(c|s)$

Probability and DAGs

- How can we know all the independence assumptions encoded by a DAG?
- The answer is **d-separation**.
- D-separation is a criterion for deciding, whether a set X of variables is independent of another set Y, given a third set Z.
- The idea is to associate “dependence” with “connectedness” (i.e., the existence of a connecting path) and “independence” with “unconnectedness” or “separation”.
- Warning: D-separation is hard to understand.
- Before trying to understand it we are going to introduce an R library for playing with DAGs: **Dagitty**.

Dagitty

- DAGitty is a browser-based environment for creating, editing, and analyzing DAGs: `http://dagitty.net/`.



- The R package 'dagitty' provides access to all of the capabilities of the DAGitty web application within R.

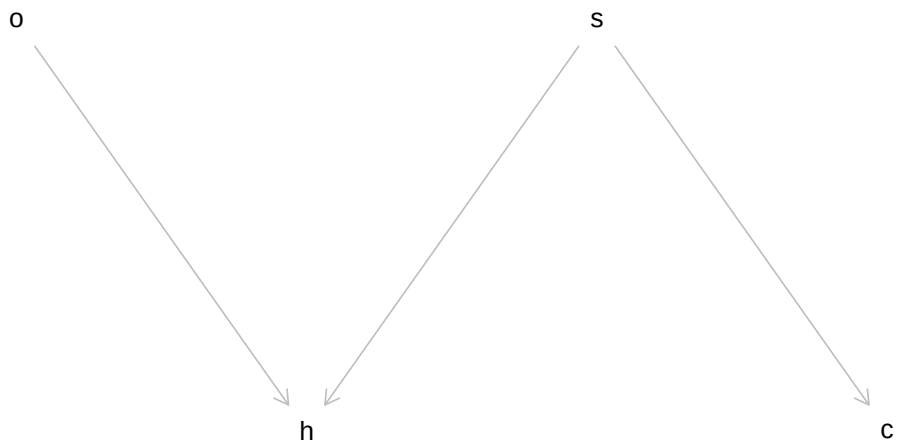
Dagitty

- We can use Dagitty to create the DAG from before with the command “dag”:

```
> library(dagitty)
>
> over <- dagitty("dag{ o -> h; s -> h; s ->c }")
```

- Now, in order to visualize the DAG we need to specify the coordinates of each vertex:

```
> coordinates(over) <-
list( x=c(o=0,h=1,s=2,c=3) , y=c(o=0,h=1,s=0,c=1) )
> plot(over)
```



Dagitty

- We can obtain the children of a node:

```
> children(over,"s")
[1] "c" "h"
```

- Or the parents:

```
> children(over,"s")
[1] "c" "h"
```

- Or the undirected paths between two nodes:

```
> paths( over, "o", "c" )$path
[1] "o -> h <- s -> c"
```

- Or just the directed paths between two nodes:

```
> paths( over, "s", "c",directed = T )$path
[1] "s -> c"
```

- Finally, we can obtain all the conditional independencies encoded by the DAG:

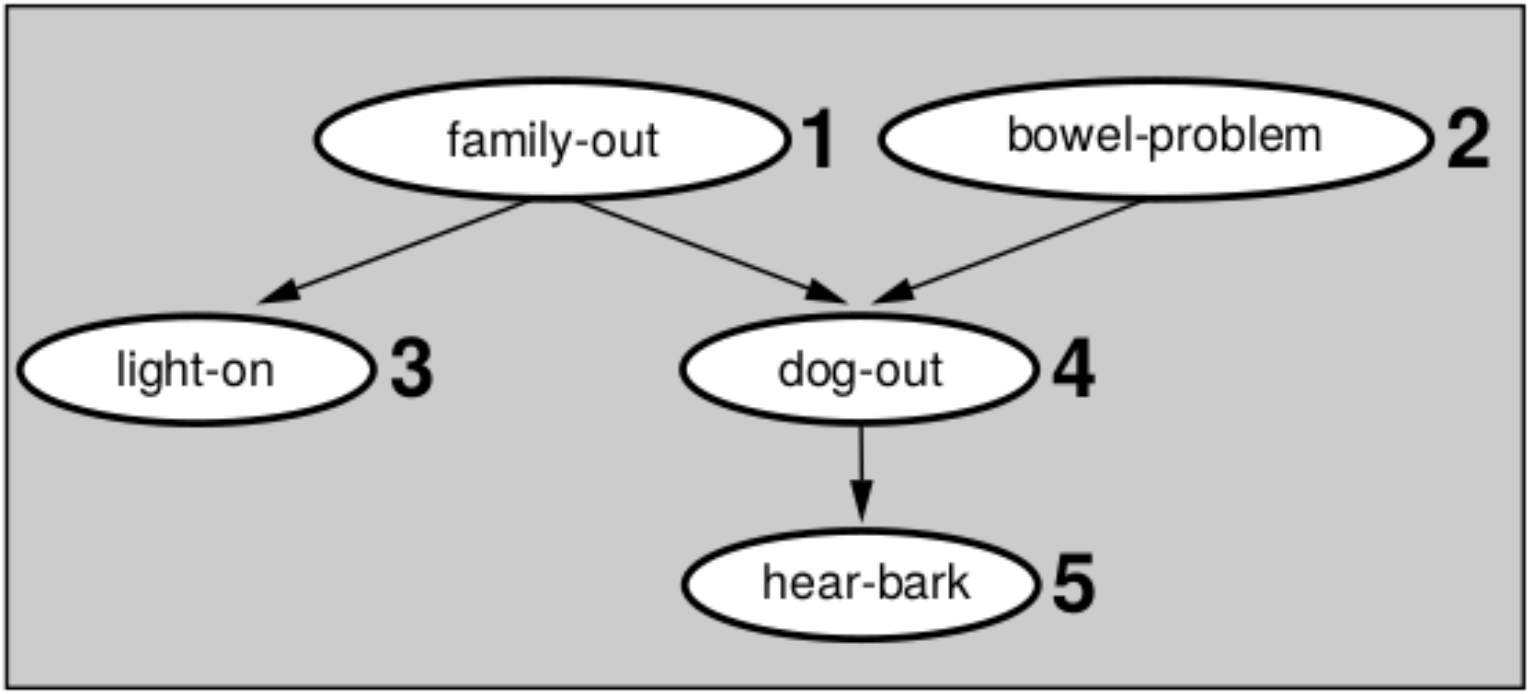
```
> impliedConditionalIndependencies(over)
c _||_ h | s
c _||_ o
o _||_ s
```

The Family Out Problem

- We are going to use the “family out” example proposed by Eugene Charniak in [Charniak, 1991] to understand d-separation.
- The situation is as follows:
- When Eugene goes home at night, he wants to know if his family is home before trying the doors.
- Often when his wife leaves the house, she turns on an outdoor light.
- Eugene’s wife can also turn on the outdoor light if she is expecting a guest.
- Also, they have a female dog.
- When nobody is home, the dog is put in the back yard.
- The same is true if the dog has bowel troubles.
- Finally, if the dog is in the backyard, Eugene’s will probably hear her barking.

The Family Out Problem

- All these causal relationships are represented in the DAG below:

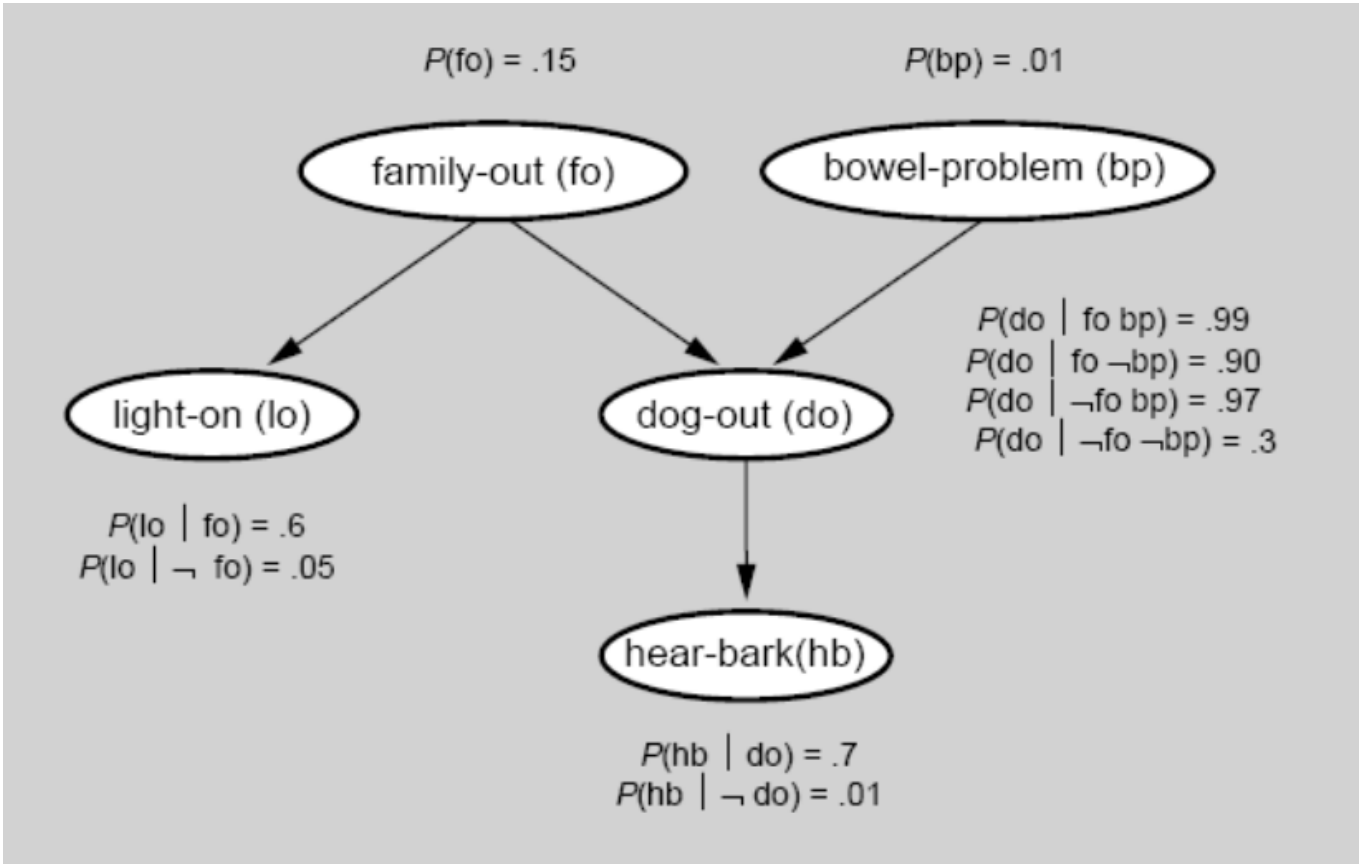


- The numbers show the corresponding position of each vertex in the topological sort.
- So, the joint probability function would be as follows:

$$f(fo, bp, lo, do, hb) = f(fo)f(bp)f(lo|fo)f(do|fo, bp)f(hb|do)$$

The Family Out Problem

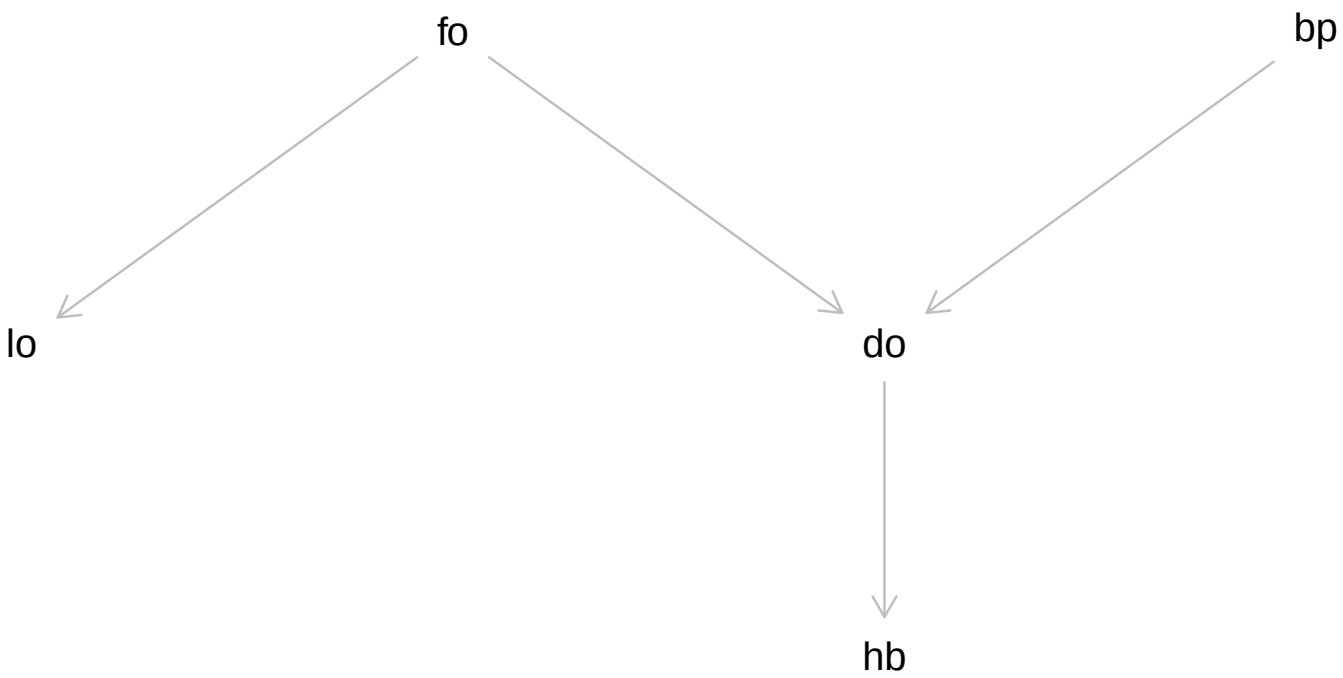
- Taking each random variable as binary, the marginal and conditional probability functions representing our DAG are shown below:



The Family Out Problem

- Let's build the DAG using dagitty:

```
fo <- dagitty("dag{ fo -> lo; fo -> do; bp -> do;do->hb }")
coordinates(fo) <- list( x=c(lo=0,fo=1,do=2,hb=2,bp=3)
                       , y=c(fo=0,bp=0,lo=1,do=1,hb=2) )
plot(fo)
```



The Family Out Problem

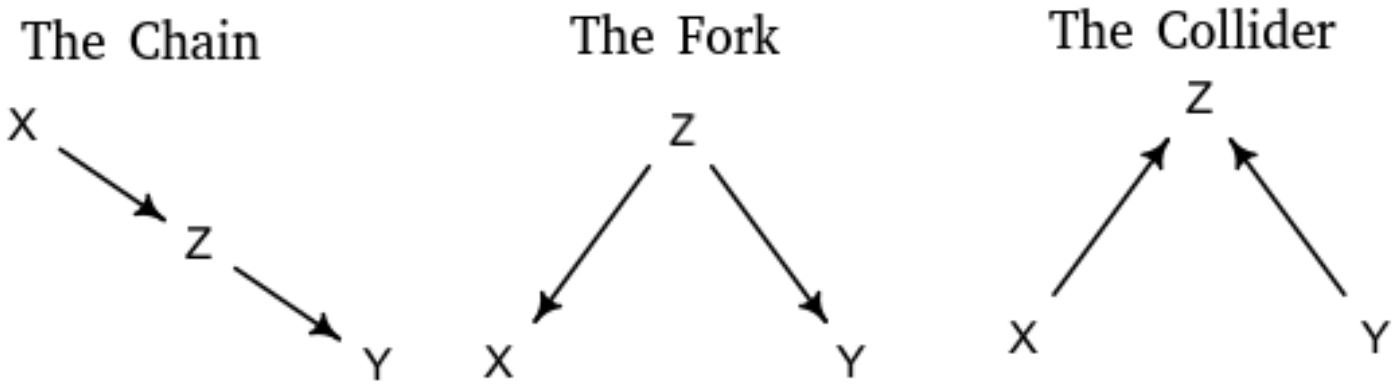
- Let's try to understand now the built-in independence assumptions of this DAG.
- Consider the random variables family-out *fo* and hear-bark *hb*.
- Are these variables independent?
- Intuitively not, because if his family leaves home, then the dog is more likely to be out.
- Thus, he is more likely to hear it bark.
- However, what if he knows that dog-out *do* is true or false?
- Now *hb* is independent of *fo* given *do*, $f(hb|fo, do) = f(hb|do)$.
- Hearing the dog bark was dependent on her being in or out.
- Once he knows whether she is in or out, then where the family is is of no concern.

The Family Out Problem

- Note that if we wanted to say that the location of the family is directly relevant to hearing the dog bark, then we would have to put another arc directly between the two variables.
- Direct relevance would occur, say, if the dog is more likely to bark when the family is away than when it is at home.
- This is not the case here.
- Now we can give the rule specifying dependence and independence in DAGs.
- In a DAG, a variable A is dependent on a variable B given evidence $E = \{E_1, \ldots, E_m\}$ if there is a d-connecting path from A to B given E.

D-separation

- We need to keep in mind that there are only three types of variable relations that combine to form all possible undirected paths in a DAG [McElreath, 2020].
- For a random variable Z and its two immediate neighbors in the path X and Y, the 3 possible relations are:



D-separation

Definition

- An undirected path from A to B is d-connecting with respect to the evidence nodes E if **every** interior node Z_i in the path satisfies one of these two rules:
  1. Z_i is a chain or a fork and not a member of E.
  2. Z_i is a collider, and either Z_i or one of its descendants is in E.

- In the literature, the term d-separation is more common.
- Two nodes are d-separated if there is no d-connecting path between them.
- So, if A and B are d-separated given E, they are conditionally independent $A \perp B|E$.

Rule 1

- The rule 1 tells us that a d-connecting path must not be **blocked** by evidence.
- This means that once we know about a middle node, we do not need to know about anything further away.
- For example, the path from *fo* to *hb*, $fo \rightarrow do \rightarrow hb$ forms a chain.
- We already discussed that *fo* and *hb* are dependent from each other, but conditionally independent given *do*.
- Now we can use the first rule of d-separation to answer these questions about dependency.
- Let's use the commands **dconnected** and **dseparated** from daggity.

```
> dconnected(fo,"fo","hb",c())
[1] TRUE
> dseparated(fo,"fo","hb",c())
[1] FALSE
> dconnected(fo,"fo","hb",c("do"))
[1] FALSE
> dseparated(fo,"fo","hb",c("do"))
[1] TRUE
```

- So, *do* blocks the path between *fo* and *hb*.

Rule 1

- Let's consider the path with a fork: *lo* $\leftarrow$ *fo* $\rightarrow$ *do*.
- By rule 1, *lo* and *do* are d-connected but d-separated give *fo*.

```
> dconnected(fo,"lo","do",c())
[1] TRUE
> dconnected(fo,"lo","do",c("fo"))
[1] FALSE
```

- So *fo* blocks the path between *lo* and *do*.
- From a causal point of view, this is the classical problem of confounding.
- Light-on and dog-out are both caused by family-out, hence they are dependent.
- However, once the cause if known, they become independent.

Rule 1

- Rule 1 also tell us that there can be no colliding interior nodes in the path.
- Let's consider the path from *fo* to *bp*, $fo \rightarrow do \leftarrow bo$.
- Here we have a collider (not a fork nor a chain), so by rule one *fo* and *bp* are marginally independent.

```
> dconnected(fo,"fo","bp",c())
[1] FALSE
```

- If two things can cause the same state of affairs and have no other connections, then the two things are independent.

Rule 2

- Suppose we know that the dog is out (i.e., dog-out is a member of E)
- Now, are family-out and bowel-problem conditionally independent given dog-out?
- By rule number 2 we know that fo and bp given do are d-connected (a collider node in the evidence).

```
> dconnected(fo,"fo","bp",c("do"))
[1] TRUE
```

- So, they became dependent given the evidence.
- How do we explain this?
- For example, if we know that the dog is out, knowing that the family is at home should raise (slightly) the probability that the dog has bowel problems.
- Because we eliminated the most likely explanation for the dog being out, less likely explanations become more likely.
- This phenomenon is also called “collider bias” or the “explain-away” effect [Pearl and Mackenzie, 2018].

Rule 2

- We would have a similar situation if we did not know that the dog was out but heard the barking.
- Since *hb* is a descendant of the collider node *do*, *fo* and *bp* are d-connected with respect to *hb*:

```
> dconnected(fo,"fo","bp",c("hb"))
[1] TRUE
```

- In this case, we would not be sure the dog was out, but we do have relevant evidence (which raises the probability), so hear-bark, in effect, connects the two nodes above the collider.
- Intuitively, rule 2 means that a d-connecting path can only go through a collider if we are conditioning on any of its potential effects.
- By “any effect” we mean the collider node itself or any of its descendants.

Rule 2

- We have finished the explanation of d-separation.
- This is undoubtedly the most difficult part of DAGs to understand.
- Exercise: explain all the conditional independencies of the family out DAG using the rules of d-separation.

```
> impliedConditionalIndependencies(fo)
bp _||_ fo
bp _||_ hb | do
bp _||_ lo
do _||_ lo | fo
fo _||_ hb | do
hb _||_ lo | fo
hb _||_ lo | do
```

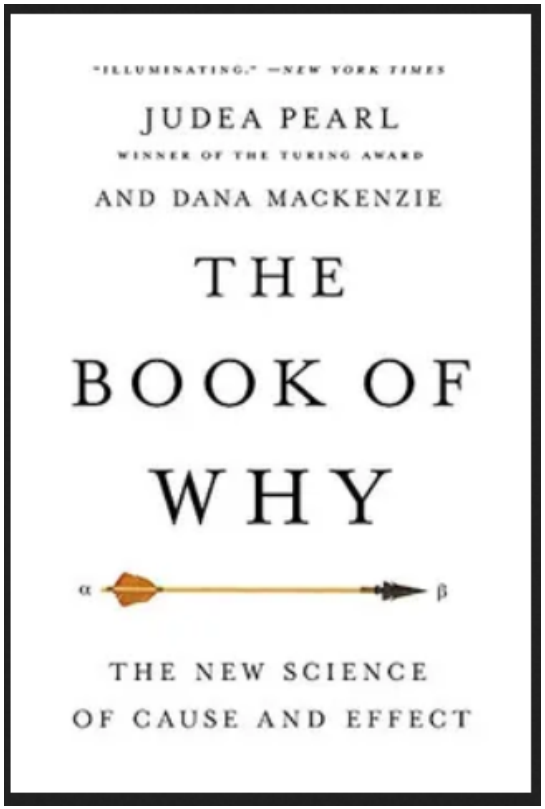
Causal DAGs

- The interpretation of DAGs as carriers of independence assumptions does not necessarily imply **causation**.
- For example, for a DAG consisting only of chains, we could reverse the order of all the arrows while retaining all the encoded independence assumptions.
- From a causal point of view, such a change would completely change the causal order of things.
- **Causal inference** is a huge topic and goes beyond the scope of this course.
- When DAGs are given a causal interpretation, they are usually called **causal graphs** or **causal diagrams**.
- This causal interpretation allows us to infer causes from observed effects (if the light is on and the dog is out, then his family is probably out).
- It is this causal interpretation that explains why DAG models are rarely used in any variable ordering other than those which respect the direction of **time** and **causation**. [Pearl, 2009]

Causal Inference

Causal Inference

- DGMs or Bayesian Networks are very fundamental in causal inference.
- Judea Pearl, the creator of them is also a foundational figure in causal inference.
- His book, “The Book of Why” [Pearl and Mackenzie, 2018] is a highly recommended introduction to the field.



Estimation for DAGs

- Two estimation questions arise in the context of DAGs.
- First, given a DAG $\mathcal{G}$ and data $d_1, \ldots, d_n$ from a distribution f consistent with $\mathcal{G}$, how do we estimate f?
- Second, given data $d_1, \ldots, d_n$ how do we estimate $\mathcal{G}$?
- The first question is pure estimation while the second involves model selection.
- These are very involved topics and are beyond the scope of this course.
- We will just briefly mention the main ideas.

Estimation for DAGs

- If we are doing frequentist inference, we typically use some parametric model $f(x|\pi_x;\theta_x)$ for each conditional density.
- The likelihood function is then

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f(d_i;\theta)) = \prod_{i=1}^{n}\prod_{j=1}^{m} f(X_{ij}|\pi_j;\theta_j)$$

- where X_{ij} is the value of X_j for the ith data point and j are the parameters for the-jth conditional density.
- We can then estimate the parameters by maximum likelihood.
- On the other hand, if we want to perform Bayesian inference we must set priors for all our variables $X_1,\ldots,X_m$ and estimate the posterior accordingly (using MCMC for example).

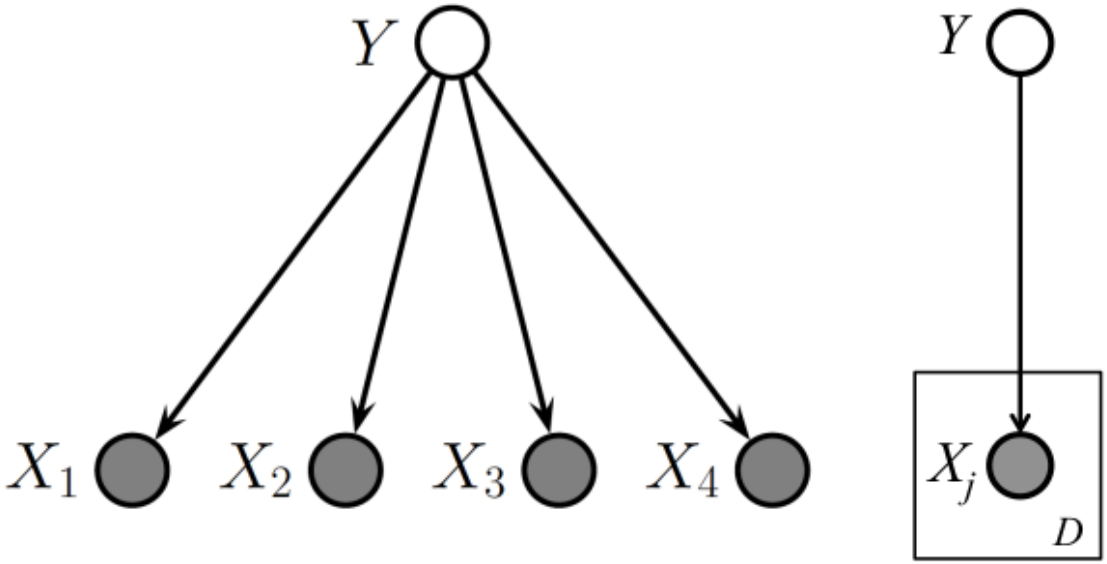
Estimation for DAGs

- To estimate the structure of the DAG itself, we could fit every possible DAG using maximum likelihood and use AIC (or some other method) to choose a DAG.
- However, there are many possible DAGs so we would need much data for such a method to be reliable.
- Also, searching through all possible DAGs is a serious computational challenge.
- Producing a valid, accurate confidence set for the DAG structure would require astronomical sample sizes.
- If prior information is available about part of the DAG structure, the computational and statistical problems are at least partly ameliorated [Wasserman, 2013].

Plate Notation

- DGM are widely used to represent probabilistic machine learning models.
- For example: Naive Bayes, Latent Dirichlet Allocation, Hidden Markov Models.
- However, when the number of variables is large, the standard DAG notation is not useful.
- Plate notation is a method of representing variables that repeat in a DAG.
- A plate or rectangle is used to group variables into a subgraph that repeat together.
- A number is drawn on the plate to represent the number of repetitions of the subgraph in the plate.

Plate Notation



- Left: variables X_j are conditionally independent given Y.
- Right: Same model, using plate notation.
- This represents the same model as the one on the left, except the repeated X_j nodes are inside a box, known as a plate.
- The number in the lower right hand corner, $D = 4$, specifies the number of repetitions of the X_j node.
- Shaded nodes indicate observed data.
- Open nodes are latent/hidden (e.g., a parameter in Bayesian inference).

A real-world application

- The Netherlands Forensic Institute uses the **Bonaparte DNA-matching software** every day, mostly for missing-persons cases, criminal investigations, and immigration cases [Pearl and Mackenzie, 2018].
- Applicants for asylum must prove that they have fifteen family members in the Netherlands.
- The idea behind Bonaparte is to make it possible to use DNA information from more distant relatives or from multiple relatives.
- Bonaparte does this by converting the pedigree of the family into a DAG.
- The central problem is that the genotype of an individual, detected in a DNA test, contains a contribution from both the father and the mother, but we cannot tell which part is which.
- Thus these two contributions (called “alleles”) have to be treated as hidden, unmeasurable variables in the DAG.

A real-world application

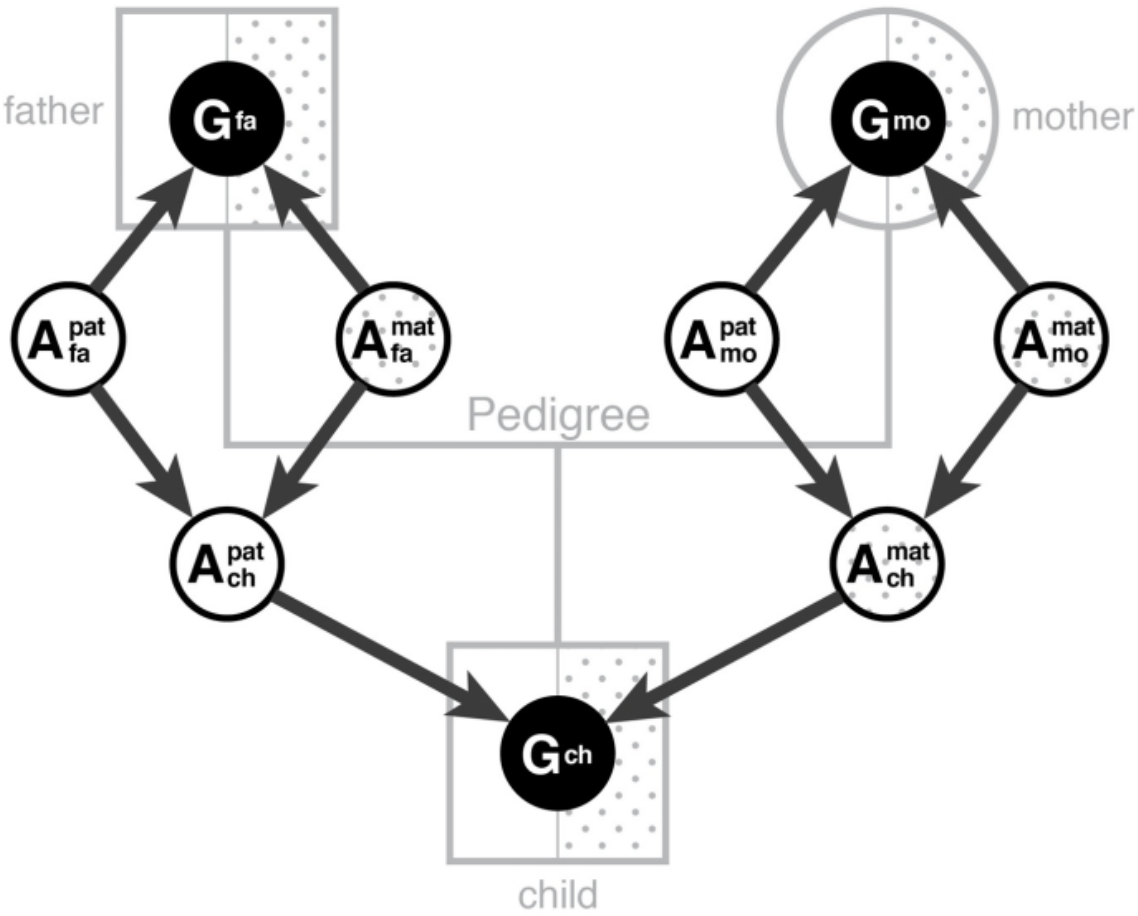


Figure: The figure shows how Bonaparte converts one small piece of a pedigree to a (causal) DAG [Pearl and Mackenzie, 2018]

A real-world application

- The system did it most impressive work after a massive disaster, such as the crash of Malaysia Airlines Flight 17.
- Few, if any, of the victims of the plane crash could be identified by comparing DNA from the wreckage to DNA in a central database.
- The next best thing to do was to ask family members to provide DNA swabs and look for partial matches to the DNA of the victims.
- Part of Bonaparte's job is to infer the probability of the cause from the evidence.
- Cause example: the victim's gene for blue eyes came from his father.
- Evidence examples: he has a blue-eyed gene and a black-eyed gene, his cousins on the father's side have blue eyes, but his cousins on the mother's side have black eyes.

A real-world application

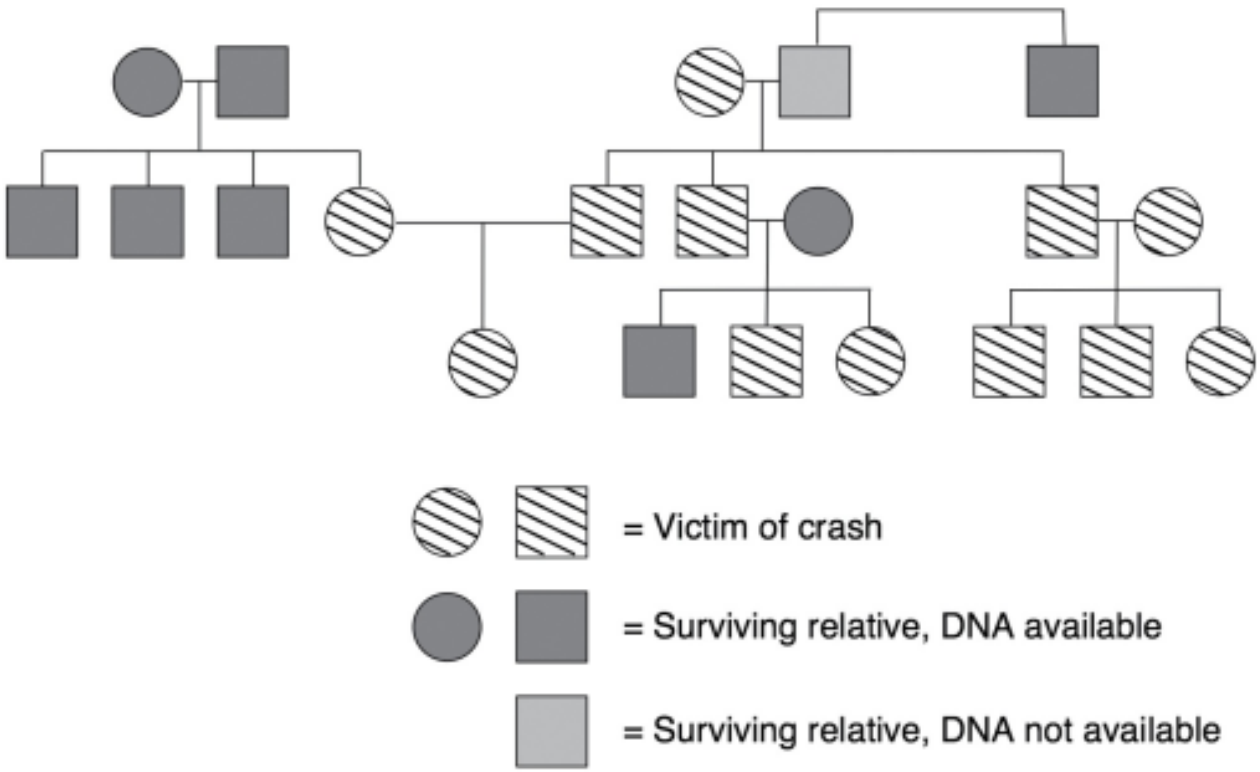


Figure: Actual pedigree of a family with multiple victims in the Malaysia Airlines crash [Pearl and Mackenzie, 2018]

A real-world application

- Once the DAG is set up, the final step is to input the victim's DNA and compute the likelihood that it fits into a specific slot in the pedigree.
- This is done by belief propagation with Bayes's rule.
- The DAG begins with a particular degree of belief in each possible statement about the nodes in the network, such as "this person's paternal allele for eye color is blue."
- As new evidence is entered into the network—at any place in the network—the degrees of belief at every node, up and down the network, will change in a cascading fashion.
- Thus, for example, once we find out that a given sample is a likely match for one person in the pedigree, we can propagate that information up and down the network.
- In this way, Bonaparte not only learns from the living family members' DNA but also from the identifications it has already made.

Conclusions

- This class introduced Directed Graphical Models (aka Bayesian Networks), a probabilistic framework for encoding conditional independence relation using graphs.
- We have introduced the three main mathematical ideas behind them: conditional independence, the chain rule of probability, and DAGs.
- D-separation is a rule for establishing whether two variables are conditionally independent given another in a DAG.
- DAGs play an important role in both machine learning and causal inference.

References I


Charniak, E. (1991).
Bayesian networks without tears.
AI magazine, 12(4):50–50.


Ermon, S. and Kuleshov, V.
Cs228 notes.


Koller, D. and Friedman, N. (2009).
Probabilistic graphical models: principles and techniques.
MIT press.


McElreath, R. (2020).
Statistical rethinking: A Bayesian course with examples in R and Stan.
CRC press.


Pearl, J. (2009).
Causality.
Cambridge university press.


Pearl, J. and Mackenzie, D. (2018).
The book of why: the new science of cause and effect.
Basic books.


Ruozzi, N.
Cs 6347: Statistical methods in artificial intelligence and machine learning.

References II


Wasserman, L. (2013).
All of statistics: a concise course in statistical inference.
Springer Science & Business Media.