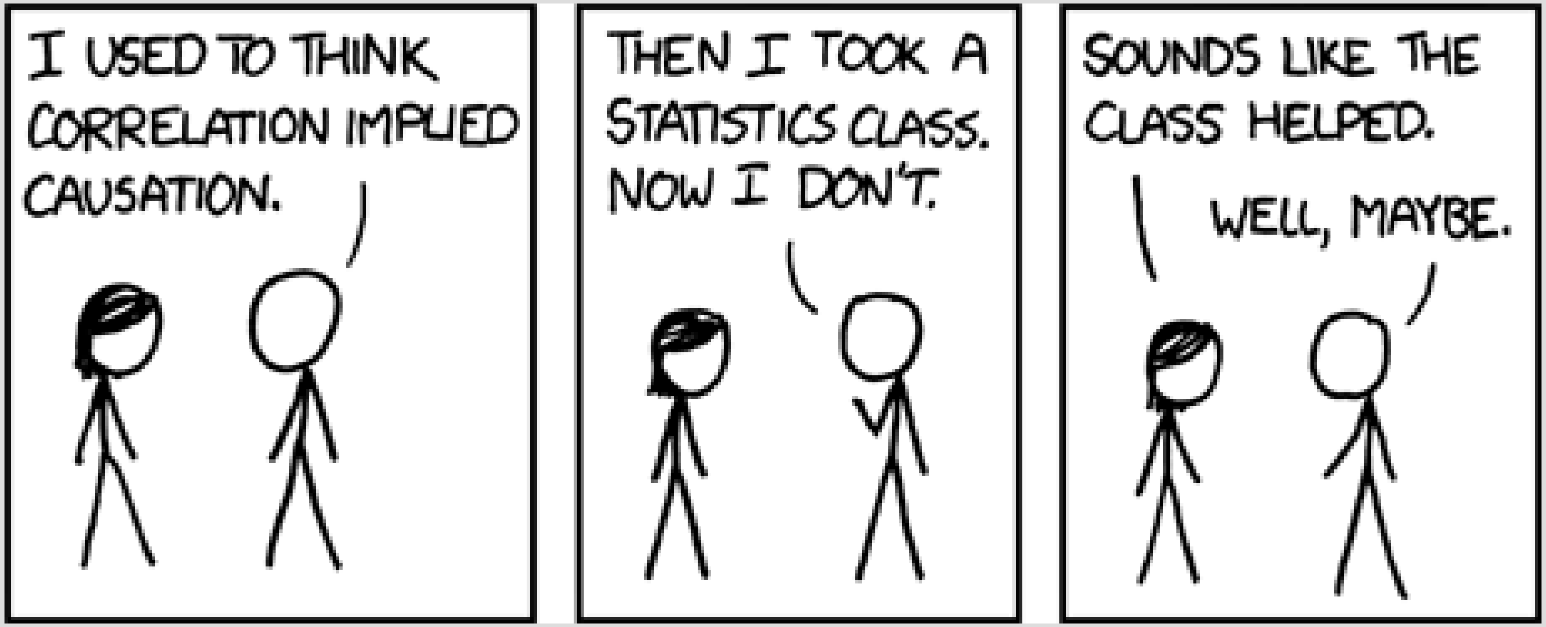
Bayes Nets (Ch. 14)


I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Independence

Independence between events can be thought of as the probabilities not effecting each other

$$P(A|B) = P(A)$$

A common example is flipping coins: if you have two coins, the outcome of the first coin flip does not effect the outcome of the second

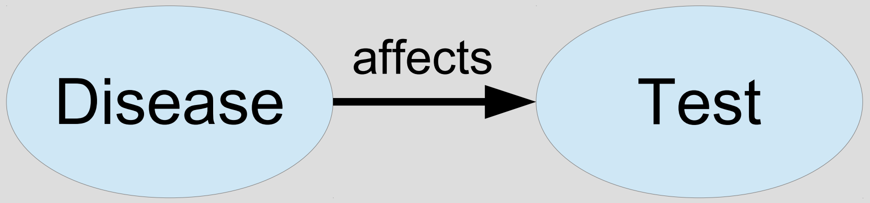
This is quite often represented as:

$$P(A,B) = P(A)P(B)$$

Independence

Independence is useful as it can let you “drop” irrelevant information (easier computation)

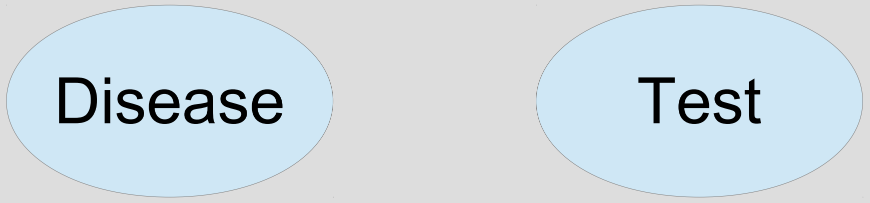
In our disease/test problem we had:



... so when computing P(d|t) we had to involve both variables (d ant t) to solve

Independence

If the test was independent of the disease (i.e. the doctor just flips a coin)



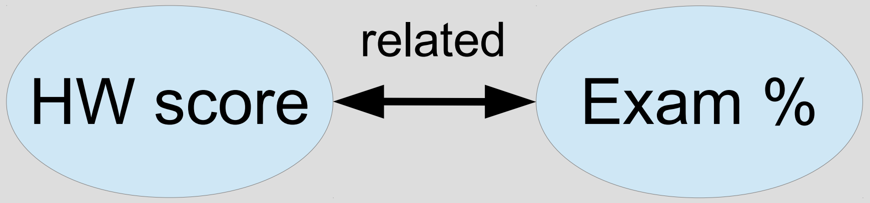
Then P(d|t) = P(d), which does not involve the right "Test" variable at all

(Note: often people think: $\text{independent} \Rightarrow P(A,B) = P(A)P(B)$
but actually also: $P(A,B) = P(A)P(B) \Rightarrow \text{independent}$)

Conditional Independence

If things are dependent, not all hope is lost

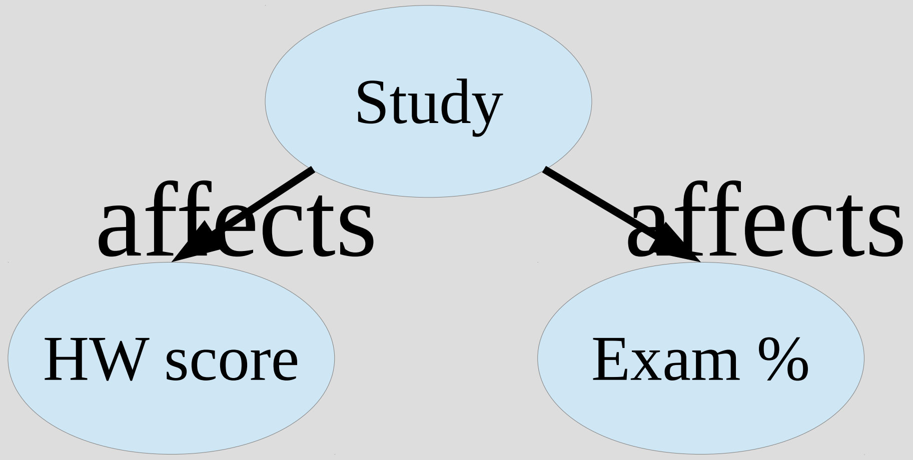
Consider a school example:



Hopefully your homework score should be some indication on how you do on an exam

Conditional Independence

But it would be more appropriate to probably have a third variable:



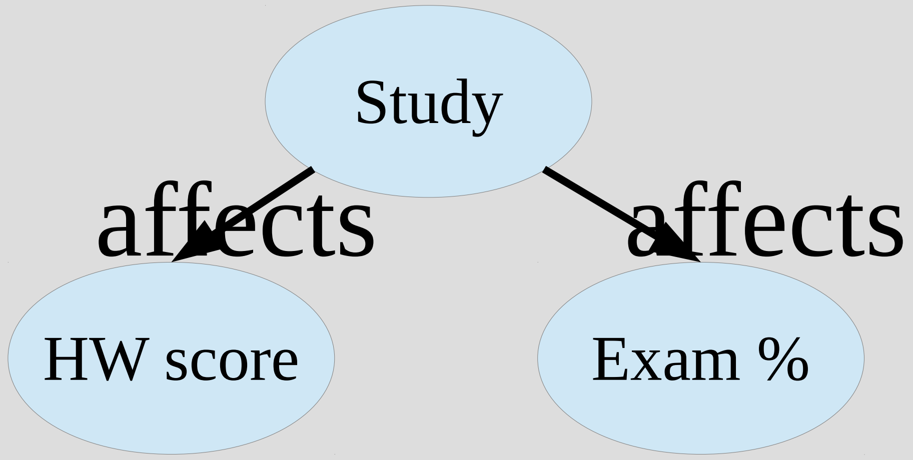
Here we would **not** say HW and Exam are independent, as $P(h,e) \neq P(h)P(e)$

However, as both homework and exams are driven by study...

Conditional Independence

We call this <u>conditional independence</u>, as

$P(h, e|s) = P(h|s)P(e|s)$



We will go over some general rules for conditionally independent in a bit

For now, the benefits are the same as normal independence: less computation

General Independence

If we have n variables and want to store a table of $P(x_1, x_2, \ldots x_n)$, there would be 2^n entries in the table (assuming x is T/F)

Without independence, we would have to store 2^n-1 of those (last skipped since sum to 1)

If all n variables are independent, you only need to store 1 value per variable

$O(n)$ is much better than $O(2^n)$

Bayesian Network

A Bayesian network (Bayes net) is:

(1) a directed graph
(2) acyclic

Additionally, Bayesian networks are assumed to be defined by conditional probability tables

(3) P(x | Parents(x))

We have actually used one of these before...

Bayesian Network

We had the following info (originally in paragraph form rather than table)

P(d)	d	¬d
	0.001	0.999

P(t|d)	t	¬t
d	1	0
¬d	0.01	0.99

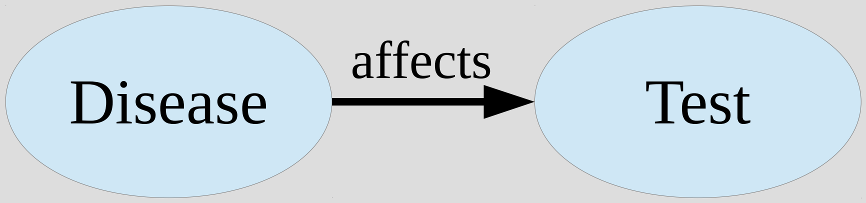
Bayesian Network

We had the following info (originally in paragraph form rather than table)

P(d)	d	¬d
	0.001	0.999

P(t|d)	t	¬t
d	1	0
¬d	0.01	0.99

If you remember the cause/effect relationship:



Bayesian Network

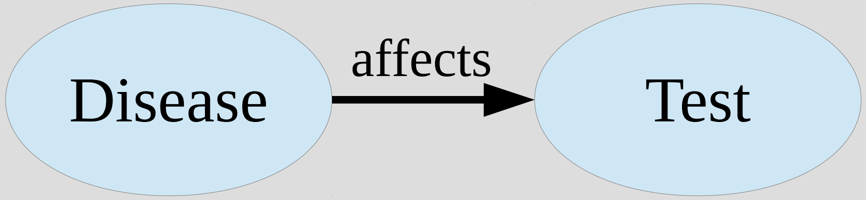
We had the following info (originally in paragraph form rather than table)

P(d)	d	¬d
	0.001	0.999

P(t|d)	t	¬t
d	1	0
¬d	0.01	0.99

(3) Test's parent is Disease in graph

If you remember the cause/effect relationship:



(1) directed

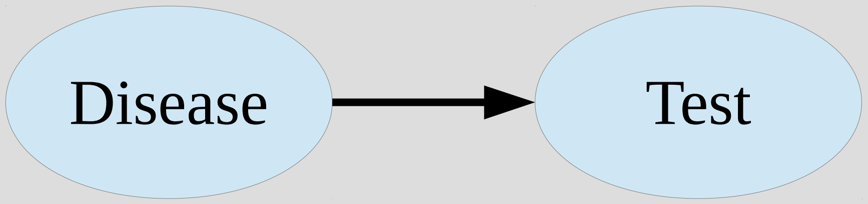
(2) acyclic

... this is, in fact, a Bayesian Network

Bayesian Network

P(d)	d	¬d
	0.001	0.999

P(t|d)	t	¬t
d	1	0
¬d	0.01	0.99



Using these tables we can manipulate the probability to find whatever we want:

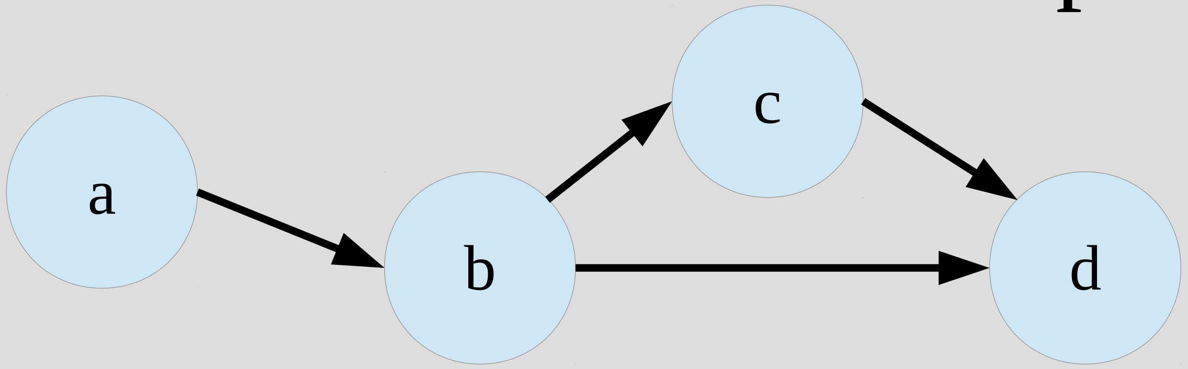
P(have disease and test would find)

$$P(d,t) = P(t|d)P(d) = 1 \cdot 0.001 = 0.001$$

$$P(d,\neg t) = P(\neg t|d)P(d) = 0 \cdot 0.001 = 0$$

Chain Rule

You are probably sick of the last example, so let’s look at a more complex one:


$$\begin{tikzcd} & & c \arrow[rd] & \\ a \arrow[r] & b \arrow[ru] \arrow[rr] & & d \end{tikzcd}$$

Using the rules of conditional probability:

$$\begin{aligned} P(a, \neg b, c, \neg d) &= P(\neg b, c, \neg d | a) P(a) \\ &= P(c, \neg d | a, \neg b) P(\neg b | a) P(a) \\ &= P(\neg d | a, \neg b, c) P(c | \neg b, a) P(\neg b | a) P(a) \end{aligned}$$

Chain Rule

Breaking down in this fashion:

$$\begin{aligned} P(a, \neg b, c, \neg d) &= P(\neg b, c, \neg d | a) P(a) \\ &= P(c, \neg d | a, \neg b) P(\neg b | a) P(a) \\ &= P(\neg d | a, \neg b, c) P(c | \neg b, a) P(\neg b | a) P(a) \end{aligned}$$

... is called the <u>chain rule</u>, written generally as:

$$P(x_1, x_2, \cdots x_n) = \prod_{i=1}^{n} P(x_i | x_{i-1}, x_{i-2}, \cdots x_1)$$

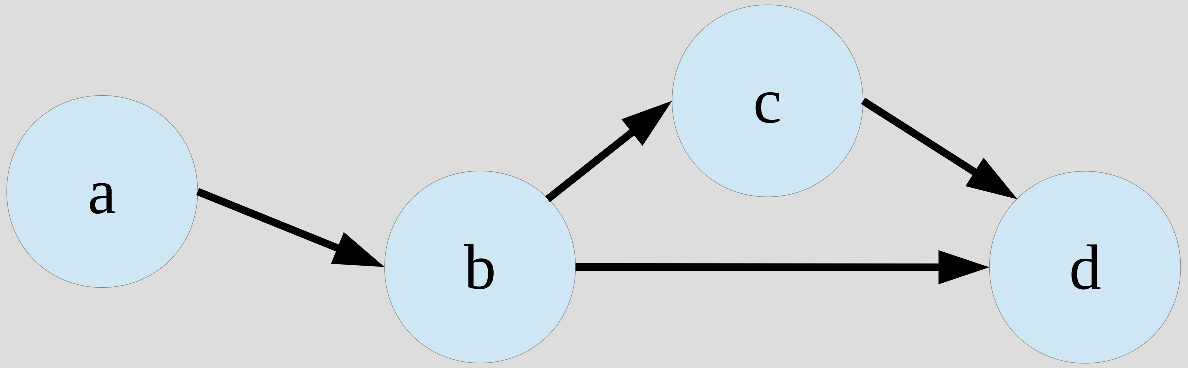
at end, p = P(x_1, x_2, ... x_n)

As code:
(no Greek)

def prob(of, given_from, given_to):

```
p = 1
for i in range(1, n):
  p *= prob(i, i-1, 1)
```

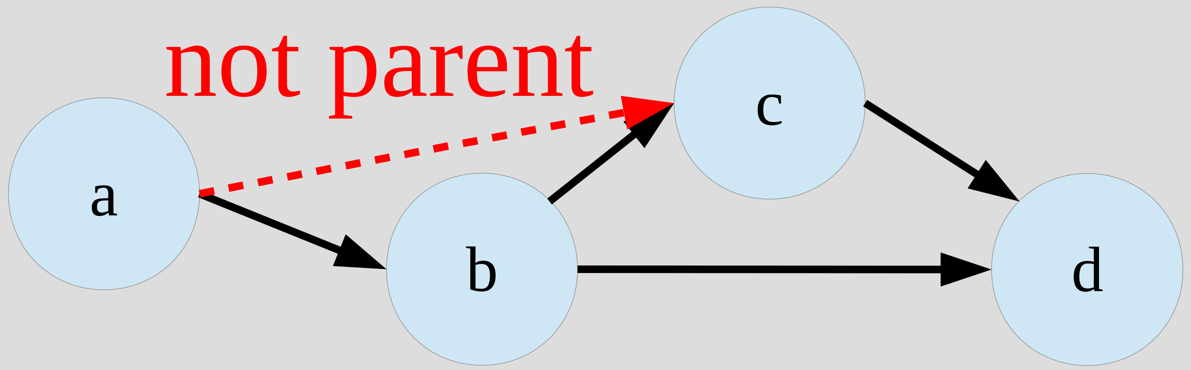
Conditional Independence



$$P(a, \neg b, c, \neg d) = P(\neg d | a, \neg b, c) P(c | \neg b, a) P(\neg b | a) P(a)$$

How is the above conflict with the definition of a Bayesian network?

Conditional Independence



$$P(a, \neg b, c, \neg d) = P(\neg d | a, \neg b, c) P(c | \neg b, a) P(\neg b | a) P(a)$$

How is the above conflict with the definition of a Bayesian network?

Bayesian networks only have tables for:

$P(x | Parents(x))$

So we only know stuff like: $P(c|\neg b)$ not $P(c|\neg b, a)$

Conditional Independence

It turns out (no coincidence), not an issue as:

$$P(c|\neg b, a) = P(c|\neg b)$$

... as ‘c’ and ‘a’ are conditionally independent given ‘b’, so the info from ‘a’ can be dropped

There are two powerful rules for conditional independence in Bayesian networks

... but the amount of given information differs:

(1) When ‘a’ is not a descendant of ‘c’

(2) No condition on ‘c’

child’s child’s child... in graph

Conditional Independence

Rule 1: When ‘a’ is not descendant of ‘c’
(Note: order matters between ‘a’ and ‘c’)

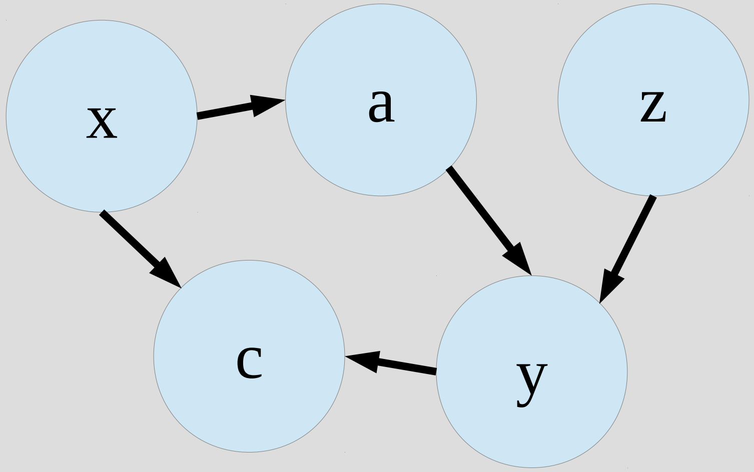
$$P(c|mustHave, a) = P(c|mustHave)$$

(redundant information side note: P(a|b) = P(a|b, b))

... when $mustHave = Parents(c)$

So in this network:

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



... but as ‘c’ is a descendant of ‘a’:

$$P(a|x, c) \neq P(a|x)$$

Conditional Independence

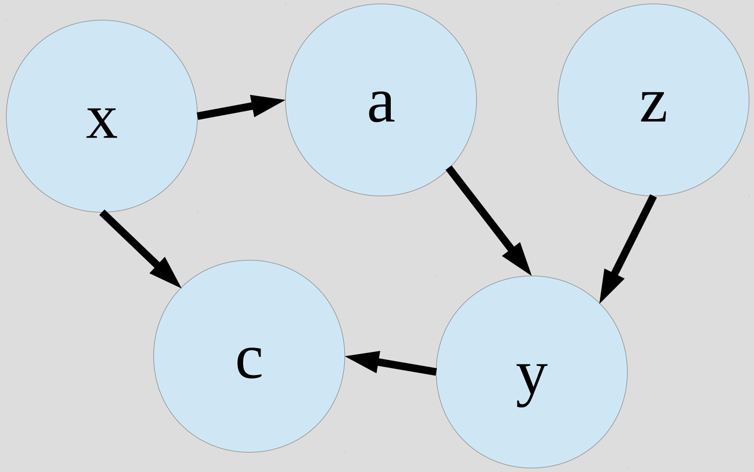
Rule 2: no restriction (called Markov blanket)

$$P(c|mustHave, a) = P(c|mustHave)$$

... when

$$\begin{aligned} mustHave =& Parents(c) + Children(c) \\ & + Parents(Children(c)) \end{aligned}$$

So in this network:



child(ren)

doesn't matter

$$P(a|x, y, z, c) = P(a|x, y, z)$$

parent(s)

child(ren)'s parent(s)

Conditional Independence

I have bad intuition on probability (in general), but let's try and see why the child's parent is needed in the Markov blanket

Blind eating network:

$$\begin{tikzcd}
\text{desert} \arrow[dr] & & \text{hungry} \arrow[dl] \\
& \text{over eat} &
\end{tikzcd}$$

You might consume too many calories when eating if you like the food or you were starved

$P(d|o) = P(d|o,h)$ ... or ... $P(d|o) \neq P(d|o,h)$?

Conditional Independence

I have bad intuition on probability (in general), but let’s try and see why the child’s parent is needed in the Markov blanket

Blind eating network:

$$\begin{tikzcd}
\text{desert} \arrow[dr] & & \text{hungry} \arrow[dl] \\
& \text{over eat} &
\end{tikzcd}$$

You might consume too many calories when eating if you like the food or you were starved

$\cancel{P(d|o) = P(d|o,h)}$ ... or ... $P(d|o) \neq P(d|o,h)$?

Conditional Independence

Assuming both liking the food and hunger increase the chance of over eating

Chance that you ate desert, knowing you overate but were full

$$P(d|o,h) < P(d|o,\neg h)$$

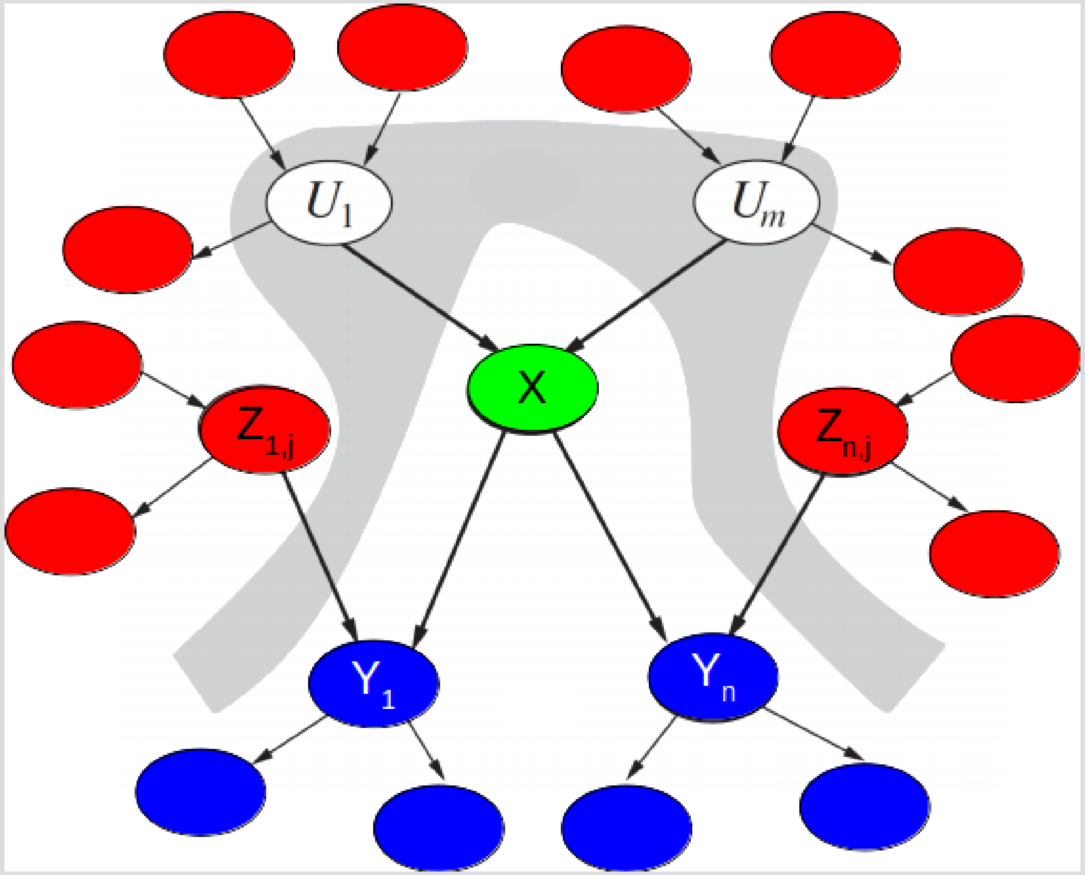
Chance that you ate desert, knowing you overate and were hungry

In both cases you know you over ate, so it is more likely you were eating desert if you were full than hungry (as hunger might be cause)

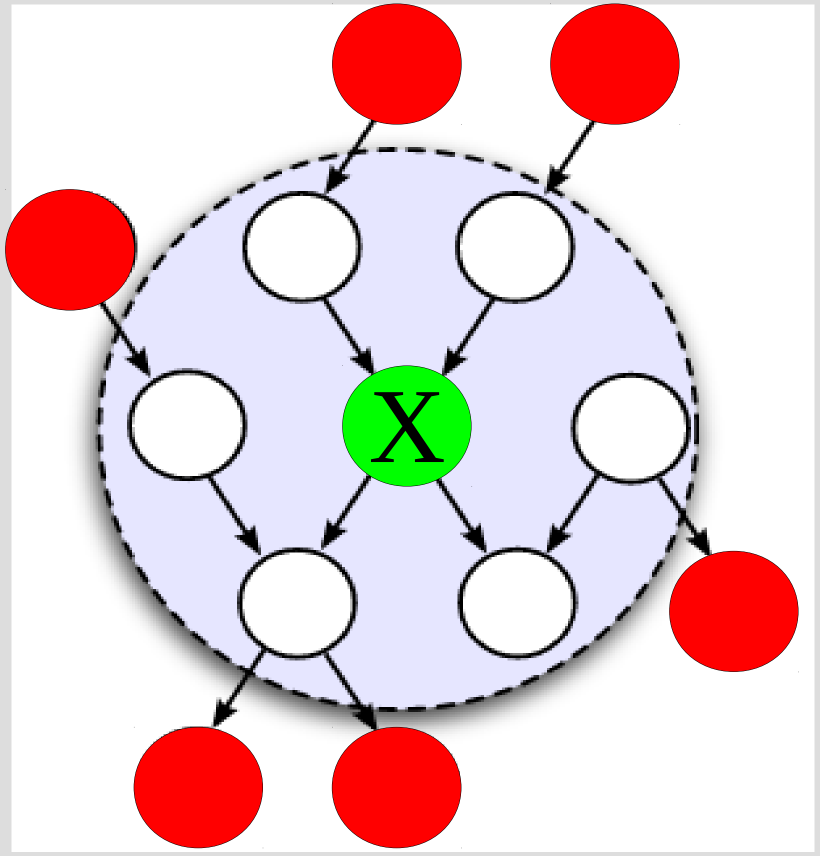
Conditional Independence

Book has a good picture of this:

Rule 1: Non-descendants



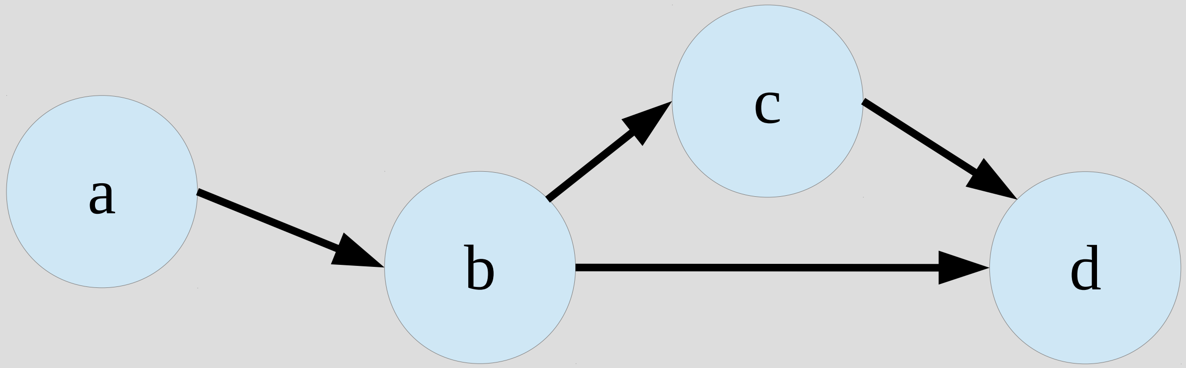
Rule 2: Markov blanket



Red = cond. independent, Blue = **not** cond. independent, White = given info, Green = P(x|stuff)

Conditional Independence

Coming back to this Bayesian network:



We only really need to use Rule #1 of conditional independence to get: (chain rule)

$$\begin{aligned} P(a,\neg b,c,\neg d) &= P(\neg d|a,\neg b,c)P(c|\neg b,a)P(\neg b|a)P(a) \\ &= P(\neg d|\neg b,c)P(c|\neg b)P(\neg b|a)P(a) \end{aligned}$$

... and we should have tables for each of these by definition (3) of Bayesian networks

Making Bayesian Networks

Thus you can get the probability P(a,b,c,d) fairly easily using the chain rule (right order) and conditional independence

Once you have P(a,b,c,d), it is pretty easy to compute any other probability you want, such as: P(a|b,c) or P(a, d)

Thus Bayesian networks store information about any probability you could want

Making Bayesian Networks

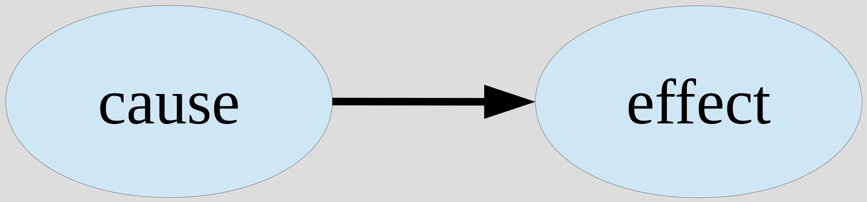
In fact, a Bayesian network is both fully expressive and does not contain redundancy

So you could put any numbers into the tables (assuming they add to 1 and $0 \leq P(x) \leq 1$) and all probability rules will be followed

Unlike how $P(a) = 0.2$, $P(b) = 0.3$, $P(a \text{ or } b) = 0.1$ does not follow the rules of probability

Making Bayesian Networks

So far, we have been building Bayesian networks as parent=cause & child=effect



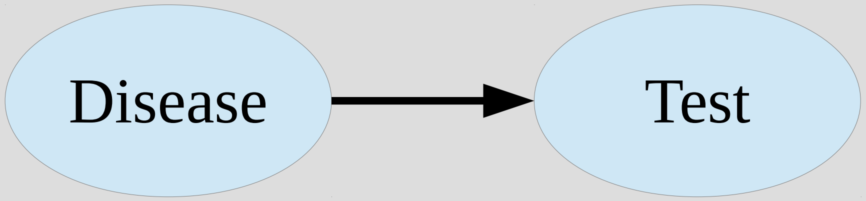
But this does not need to be the case
(it is just typically the best way)

In fact, there are multiple networks that can represent the same information

Making Bayesian Networks

P(d)	d	¬d
	0.001	0.999

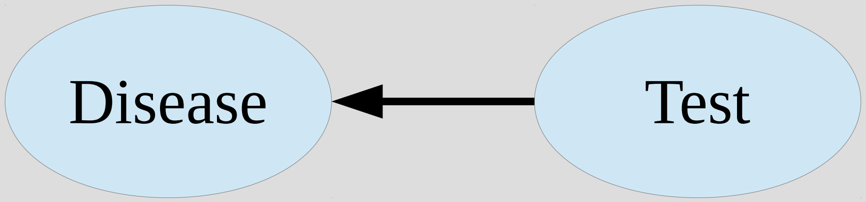
P(t|d)	t	¬t
d	1	0
¬d	0.01	0.99



... same as...

P(d|t)	d	¬d
t	0.090992	0.909008
¬t	0	1

P(t)	t	¬t
	0.01099	0.98901



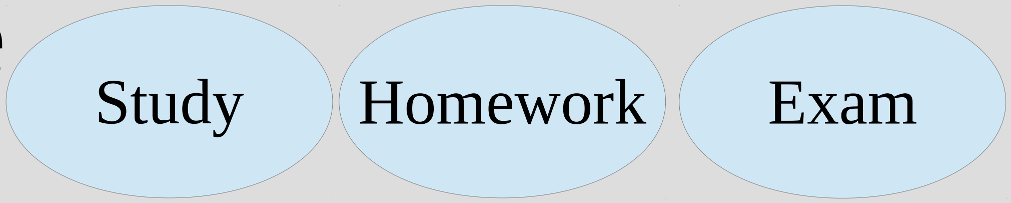
Making Bayesian Networks

If you have nodes/variables that you want to make into a Bayesian network, do this:

1. Assign variables to X_1, X_2, ... X_n (any order)
2. for i = 1 to n:
   - 2.1. Find minimal set of parents from X_{i-1}, X_{i-2}, ... X_1 such that $P(X_i|X_{i-1} \cdots X_1) = P(X_i|Parents(X_i))$ (i.e. non-descedent rule for cond. prob.)
   - 2.2. Make table & edges from Parents(X_i) to X_i

Making Bayesian Networks

Let's consider the Study, Homework, Exam situation from last time



Since we can choose these variables in any order, let X_1=Study, X_2=Homework, X_3=Exam

First loop iteration i=1, so we need to find:

$$P(X_i|X_{i-1}\cdots X_1) = P(X_i|Parents(X_i))$$

... but when i=1, $P(X_1|\{nothing\}) = P(X_1)$

... so no parents are needed to be found

Making Bayesian Networks

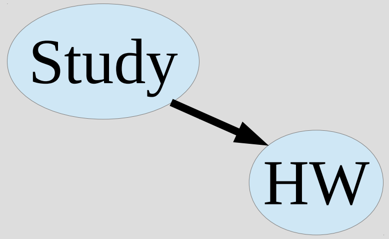
Next loop iteration i=2, and again we find:

$$P(X_i|X_{i-1}\cdots X_1) = P(X_i|Parents(X_i))$$

There are really two options: $P(X_2|X_1) = \begin{cases} P(X_2) \\ P(X_2|X_1) \end{cases}$

As X_1=Study and X_2=Homework are not independent, the first option is not possible

So we choose: Parents(X_2) = {X_1} and make a table for P(X_2|X_1) and update graph:



Making Bayesian Networks

Last iteration i=3, and again we find parents:

$$P(X_i|X_{i-1} \cdots X_1) = P(X_3|X_2, X_1) = P(X_i|Parents(X_i))$$

Parents(X_3) = {X_2, X_1} would work

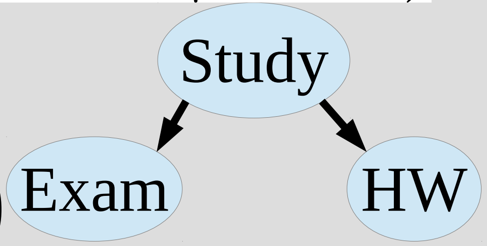
(this is the rule of conditional probability)

But this is not minimal, as Homework and Exam are conditionally independent

$$P(X_3|X_2, X_2) = P(Exam|HW, Study) = P(Exam|Study)$$

So the minimal parent set is {X_1}

... make a table for P(Exam|Study)

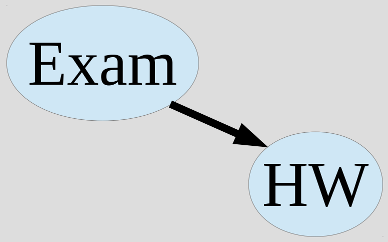


Making Bayesian Networks

Let’s do this again, but switch the order:
X_1 = Exam, X_2 = Homework, X_3=Study

i=1 loop iteration pretty trivial (no parents)
i=2 iteration finds minimal set of parents for Homework, which is {Exam}
(only other node and it does effect Homework)

Tables: P(e), P(h|e)



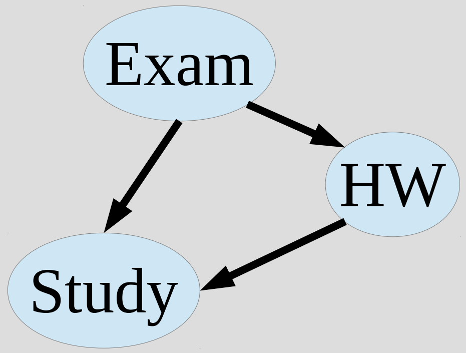
Making Bayesian Networks

When i=3, we add Study to the graph and see if we can find some conditional independence:

$P(Study|Exam, HW)$

However, both Exam and Homework affect the probability that we Studied, so Parents(Study) = {Exam, HW}

Tables: P(e), P(h|e), P(s|e,h)



$$\begin{tikzcd} & Exam \arrow[dl] \arrow[r] & HW \arrow[dll] \\ Study & & \end{tikzcd}$$

Making Bayesian Networks

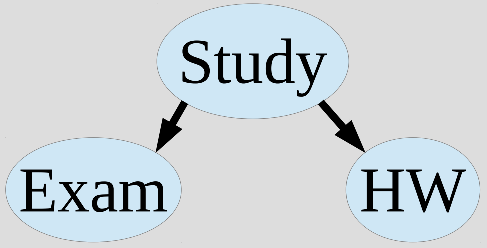
So depending on variable order we have:

P(s)	0.1

P(h|s)	0.2
P(h|¬s)	0.3

P(e|s)	0.4
P(e|¬s)	0.5

random numbers

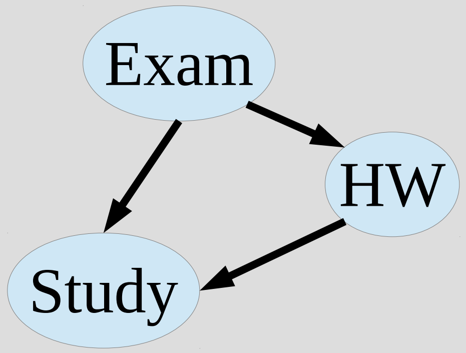


… or …

P(e)	0.49

P(h|e)	0.292
P(h|¬e)	0.288

P(s|e,h)	0.056944
P(s|¬e,h)	0.081633
P(s|e,¬h)	0.092219
P(s|¬e,¬h)	0.132231



Making Bayesian Networks

Like last time, the cause $\rightarrow$ effect direction is more stable to remember (changes less)

We mentioned last time that storing a table of P(a,b,c,d...) takes $O(2^n)$

If you have at most k parents on the Bayesian network, then it is actually $O(n*2^k)$ (previous slide was k=2, as study had two parents and thus required 4 entries in table)

Making Bayesian Networks

So choosing “cause” variables before “effect” ones, you get:

1. More stable probabilities
   (update fewer tables on changes)

2. Less memory used to store probabilities

Not to mention, finding P(effect|cause) is often much easier to compute in the real world