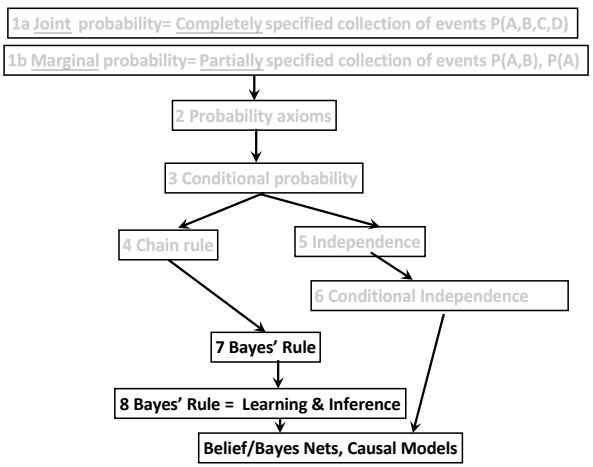
6.034 Artificial Intelligence: Lecture 11
Probabilistic reasoning 2

Robert C. Berwick

September 28, 2020

Where we have been



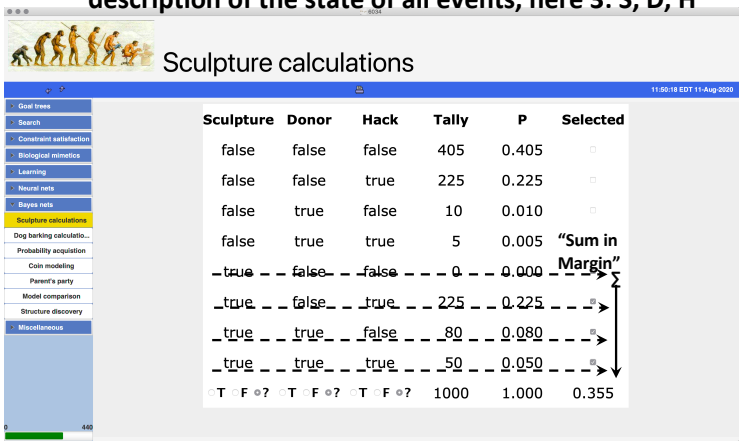
Menu for today

- ❑ Where we have been: probability as a language for representing uncertainty
- ❑ Where we are going:
- ❑ Naïve Bayes (aka "Idiot Bayes") – the simplest "inference net"
- ❑ Graphical probabilistic models aka "Bayes nets": focused & efficient reasoning about causality in an uncertain world
  - acyclic direct graphs representing independence assumptions

The 3 related types of probabilities

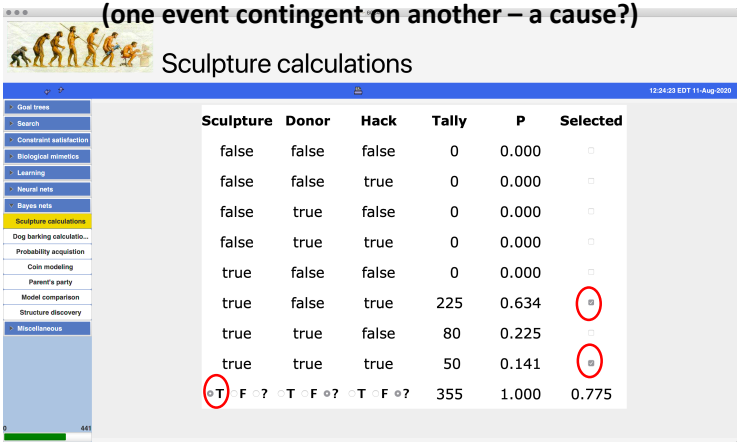
- Joint probability of a completely specified set of events (i.e. involving all of the variables in our universe/problem), e.g., P(Statue, Hack, Donor)
- Marginal probability of an incompletely specified set of events, summed over all the possibilities of others, e.g. P(Statue), with Hack and Donor "added up" over all their possible values or "marginalized out"
- Conditional probability, given that particular information is certain, e.g., P(Statue | Donor)

Joint probability table = a <u>complete</u> probabilistic description of the state of all events, here 3: S, D, H



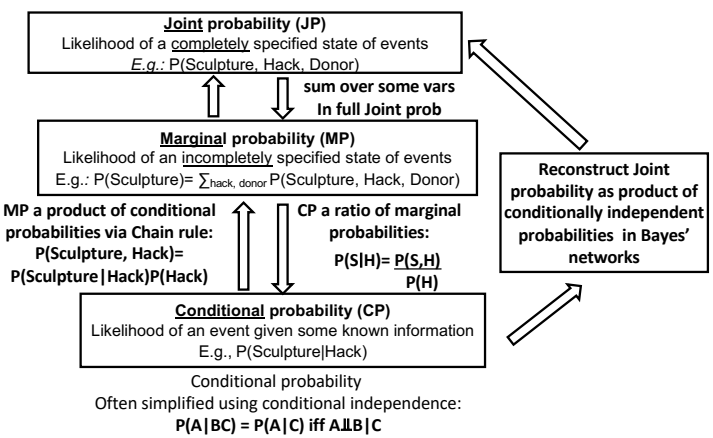
<u>Marginal</u> probability: likelihood of some <u>incompletely</u> specified state of events, e.g., Probability(Sculpture), summed over the Donor and Hack events

<u>Conditional</u> probability: Probability of Hack <u>given</u> Sculpture (one event contingent on another – a cause?)



P(Sculpture | within Hack = True world)

Relating the 3 types of probabilities (see also



Having the right priors is essential for Bayesian inference to work

- Kirk is flipping a fair coin (50-50 heads/tails)
- Spock, a Bayesian AI, has only two prior probabilities (hypotheses) about this coin, each assigned a probability of 50%:
  1. Hypothesis-1: Coin is Head-biased, such that it lands heads 3/4 of the time
  2. Hypothesis-2: Coin is Tail-biased, such that it lands tails 3/4 of the time
- So, Spock's priors are P(Hypothesis-1)=0.5; P(Hypothesis-2)=0.5
- Spock will now start observing the coin flips and being a logical begin, will change its posterior probabilities in accordance with Bayes' rule… will Spock converge in the long run to what we would consider a reasonable result? (viz., after a trillion flips…conclude the coin is unbiased and the prob(Heads)=0.5?)

Spock's posterior probabilities for Heads- vs. Tails-biased coin, depending on # of heads & tails observed

Let *z* be the difference in the observed # Heads–Tails

Then Spock's posterior probabilities are:

$P(\text{Heads} \mid z) = 3^z/(3^z + 1)$

$P(\text{Tails} \mid z) = 1/(3^z + 1)$

Spock's estimate that the next flip will be heads is:

$(3/4)P(H \mid z) + (1/4)P(T \mid z) = (3^{z+1} + 1)/(4(3^z + 1))$

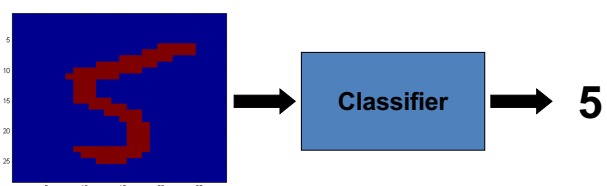
(Exercise for this derivation left for curious students. *Hint*: first consider the probability of exactly *h* Heads turning up after *n* coin flips, for both the Heads-biased and Tails-biased coins as a probability product, and then use this to calculate Spock's posterior probability likelihoods of Heads-biased vs. Tails-biased using Bayes' rule as the ratio of head/tails-biased coins …)

Using conditional probabilities & Bayes take 1: Naïve Bayesian classification

- Problem:
  - Given features $X_1, X_2, \ldots, X_n$
  - Predict a label Y
- Examples:
- Spam Classification
  - Given an email, predict whether it is spam or not
- Medical Diagnosis
  - Given a list of symptoms, predict whether a patient has disease X or not
- Weather
  - Based on temperature, humidity, etc.,… predict if it will rain tomorrow

Specific application

- Digit Recognition – MNIST database from Census



- $X_1,\ldots,X_n \in \{0,1\}$ (Black vs. White pixels)
- $Y \in \{5,6\}$ (predict whether a digit is a 5 or a 6) – a label (in general, $Y \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

A Bayesian classifier

- Use this formula to find the "best" theory Y that accounts for the evidence (the data features) – here, Y can be any of the 10 digits 0,…,9

$$\underset{Y \in \{0,\ldots,9\}}{\text{maximize}} P(Y|X_1,\ldots,X_n)$$

Conventional notation:

$$\arg\max_Y P(Y|X_1,\ldots,X_n)$$

  - For example: what is the probability that the image represents a 5 given its observed pixels?

- Q: So … How do we compute that?

A Bayesian classifier

- Use Bayes Rule!

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

Likelihood — Prior — Normalization Constant

- Why did this help? Well, we think that we might be able to specify how features are "generated" by the class label

A Bayesian classifier

- Let's expand this for our digit recognition task:

$$P(Y=5|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=5)P(Y=5)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

$$P(Y=6|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=6)P(Y=6)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

- To classify, we'll simply compute these two probabilities and predict based on which one is greater– a model selection problem–but what about those joint probabilities :

  $P(X_1, \ldots, X_n \mid Y=\{0,\ldots,9\})$?

Model Parameters

- For the Bayes classifier, we need to "learn" two functions, the likelihood and the prior
- That joint conditional probability poses a problem, however...typical of a joint probability table: if *n* variables, the table is of size 2^n
- How many parameters are required to specify our digit recognition example? (Suppose 50 x 50 pixel array for each image…each pixel can be on or off, if just b/w)
- We will need a lot of training data…

One solution: Naïve Bayes model (a specific instance of a Bayesian, graphical network machine learning model)

- The *Naïve Bayes Assumption*: Assume that all features X_i are conditionally independent given the class label Y
- Recall this lets us multiply the complex conditional out as a product of the variables conditioned on Y separately:

$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

- (We will discuss the validity of this assumption later)

Why is this useful?

- # of parameters for modeling $P(X_1,\ldots,X_n|Y)$:
  - $2(2^n-1)$
- # of parameters for modeling $P(X_1|Y),\ldots,P(X_n|Y)$
  - 2n
- Reduces # of parameters from exponential to linear – a general strategy that makes use of (conditional) independence

Naïve Bayes Training

- **Now that we've decided to use a Naïve Bayes classifier, we need to train it with some data:**

MNIST Training Data

Naïve Bayes Training

- **Training in Naïve Bayes is easy:**
  - **Estimate P(Y=v) as the fraction of records with Y=v**

$$P(Y=v) = \frac{Count(Y=v)}{\#\ records}$$

  - **Estimate P(X_i=u|Y=v) as the fraction of records with Y=v for which X_i=u**

$$P(X_i=u|Y=v) = \frac{Count(X_i=u \wedge Y=v)}{Count(Y=v)}$$

- **(This corresponds to a so-called "Maximum Likelihood estimation" (MLE) of the model parameters)**

Naïve Bayes training

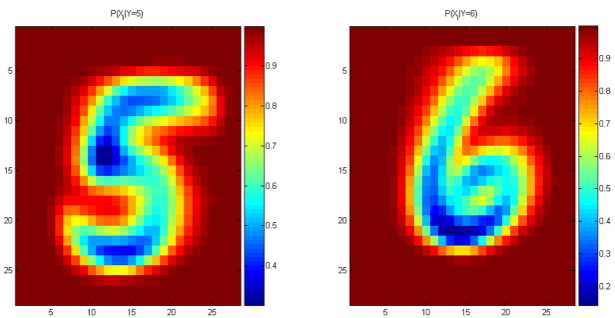
- In practice, some of these counts can be zero
- Fix this by adding "virtual" counts (in the simplest way, i.e., what Laplace did in the 17th century for a gambler):

$$P(X_i=u|Y=v) = \frac{Count(X_i=u \wedge Y=v)+1}{Count(Y=v)+2}$$

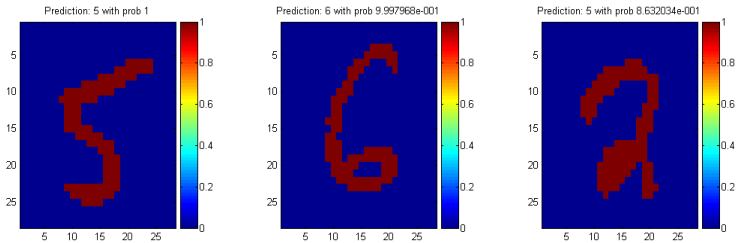
  - This is called *Smoothing*
  - *Smoothing* = estimating what is unobserved from what has already been observed, and is therefore, in general, as hard as the general problem of induction – what this writer considers one of the *Dark Arts*

Naïve Bayes training

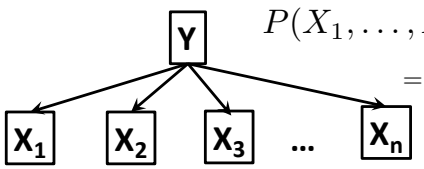
- For binary digits, training amounts to averaging all of the training fives together and all of the training sixes together



Naïve Bayes classification



A graphical model picture of naïve Bayes showing dependencies of X_i's on Y



$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

This is a directed, acyclic graph (DAG) The arrows show a direction of probabilistic dependence-an "inference net" or "graphical model"

Q: is it <u>true</u> that the X_i's are independent from one another, <u>given</u> Y?

Gold star idea: (Conditional) Independence lets us divide and conquer via factorization (containing effects to a local area)

In this case, the exponential size joint probability structure can be boiled down to a much simpler probability structure

Evaluating classification algorithms

I tell you that it achieved 95% accuracy on my data.

Is your technique a success?

Types of errors

- But suppose that
  - The 95% is the correctly classified pixels
  - Only 5% of the pixels are actually edges
  - It misses all the edge pixels

- How do we count the effect of different types of error?

Types of errors

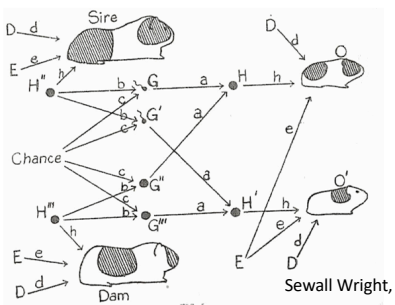
Ground Truth	Prediction: Edge	Prediction: Not edge
Edge	True Positive	False Negative
Not Edge	False Positive	True Negative

Recap so far

- We defined a *Bayes classifier* but saw that it's intractable to compute $P(X_1,\ldots,X_n|Y)$
- We then used the *Naïve Bayes assumption* – that <u>everything</u> is conditionally independent given the class label Y – to *localize* computation and so simplify the computation from exponential to linear

- Q: A natural question– is there some happy compromise where we only assume that *some* features are conditionally independent?
- Yes: that is what Inference nets/Belief nets/Bayes nets/Causal nets/Graphical models are for

Bayes net motivation
What's cause and effect?



Sewall Wright, Harvard Grad student, 1918

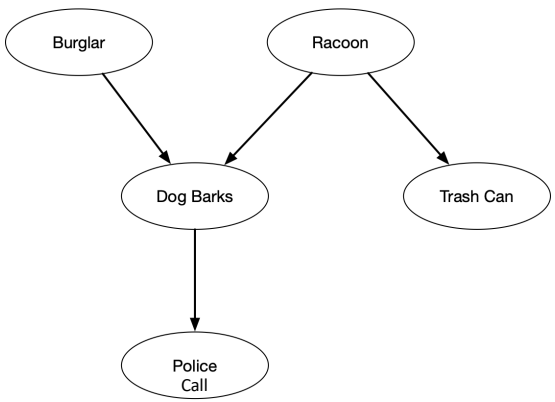
Correlation & Causation, J. Heredity, 1921

Taken to be a formal <u>definition</u> of cause-and-effect

Care to guess who did this and when?

Inference nets/Belief nets/Bayes nets

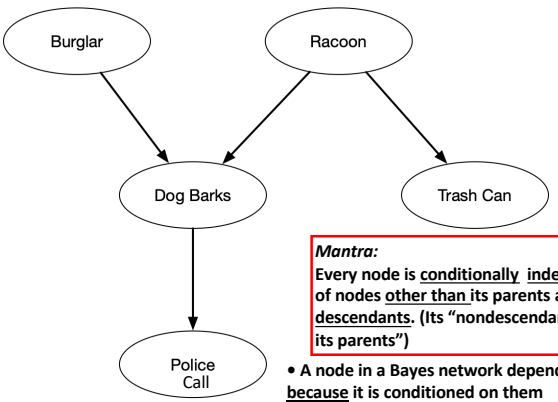
- Story told via a graph – an "influence diagram"



Math behind the story

- A Directed Acyclic Graph (DAG) representing independence conditions – edges denote probabilistic conditioning dependencies among uncertain variables
- Each node is associated with a set of probability distributions
- Lack of edges = assertions of conditional independence
- From this we can reconstruct the full Joint probability table, and then any other probabilities of interest concerning the variables, using the chain rule or Bayes' rule

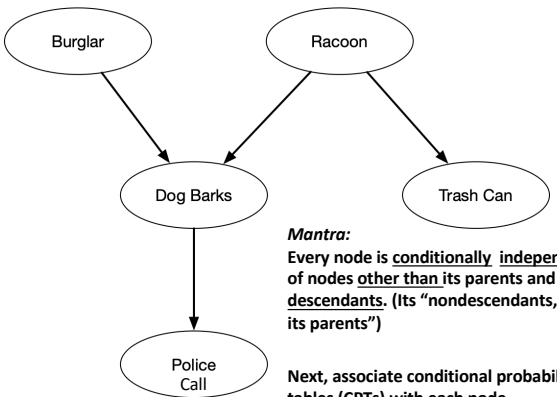
Barking Story expressed as Bayes network



Mantra:
Every node is conditionally independent of nodes other than its parents and its descendants. (Its "nondescendants, given its parents")

- **A node in a Bayes network depends on its parents because it is conditioned on them**
- **A node depends on its descendants because these give (Bayesian) evidence as to the node's state**

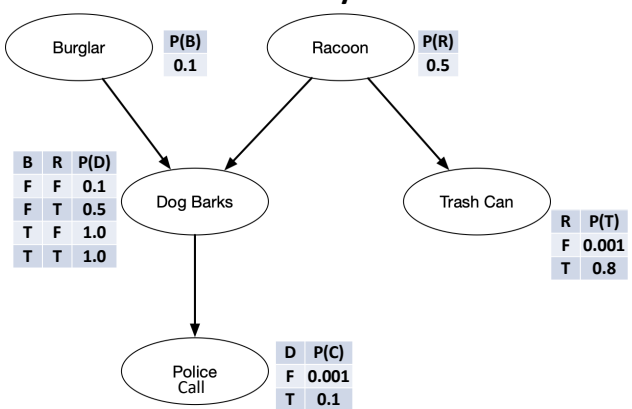
Barking Story expressed as network



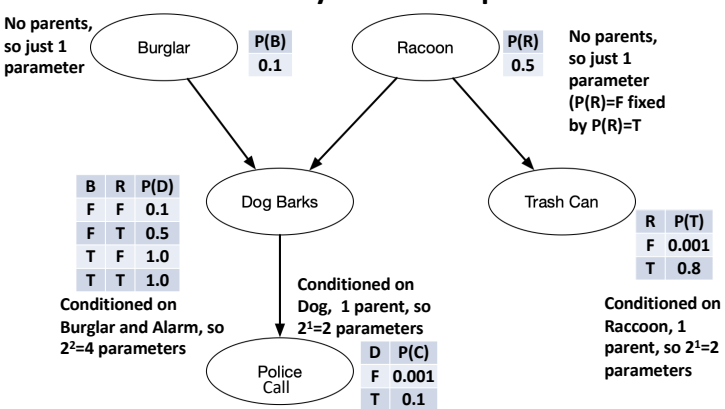
Mantra:
Every node is conditionally independent of nodes other than its parents and its descendants. (Its "nondescendants, given its parents")

Next, associate conditional probability tables (CPTs) with each node

Conditional Probability Tables (CPT) associated with nodes in Bayes' net



Conditional Probability Tables (CPT) associated with nodes in Bayes' net & # parameters

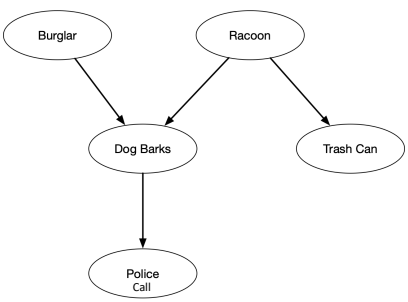


Total: 10 parameters, instead of $2^5-1=31$ for full Joint table

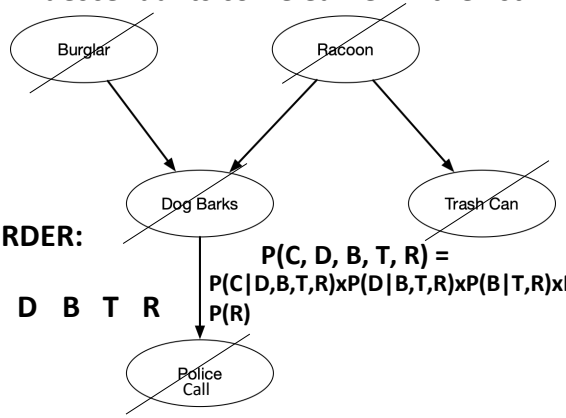
Bayes network allows efficient reconstruction of joint probability tables

Compute P(C,D,B,T,R), by chain rule…

= P(C|D,B,T,R)xP(D|B,T,R)xP(B|T,R)xP(T|R)xP(R)



Chain rule event ordering: ensure no descendants come earlier in the list



ORDER:

C D B T R

P(C, D, B, T, R) =
P(C|D,B,T,R)xP(D|B,T,R)xP(B|T,R)xP(T|R)x P(R)

Bayes network allows efficient reconstruction of joint probability tables

P(C,D,B,T,R), by chain rule…

= P(C|D,B,T,R)xP(D|B,T,R)xP(B|T,R)xP(T|R)xP(R)

But by conditional independence in the graph:

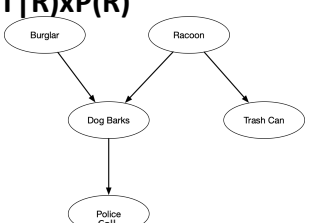
= P(C|D,~~B,T,~~R)xP(D|B,~~T,~~R)xP(B|~~T,R~~)xP(T|R)xP(R)

=P(C|D)xP(D|B,R)xP(B)xP(T|R)xP(R)

=0.1 x 1 x 0.1 x 0.8 x 0.5

=0.004

(what if no Racoon, i.e., R=False?)



In general: Bayes Network Allows Reconstruction of Joint Probability Tables

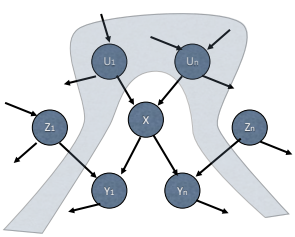
- $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{Parents}(x_i))$

where Parents yields the parents of a node

(compare this to the naïve Bayes formula earlier that made use of conditional independence assumption)

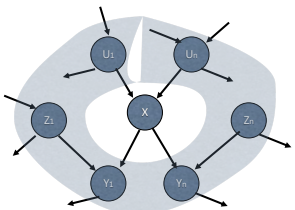
- **Since conditional independence is the key to constructing the networks, we need some tools for determining this**

Visual topological Interpretations re conditional independence in Bayes networks – the general case



A node, *X*, is *conditionally independent* of its non-descendants, Z_i, given its parents, U_i.

Suffices to reconstruct joint probability table



A node, *X*, is *conditionally independent* of all other nodes in the network given its *Markov blanket*: its parents, U_i, children, Y_i, and children's parents, Z_i.

Used for local sampling methods in computation

Most general conditional independence question re a Bayes net: Is a set of nodes X conditionally independent of another set Y, given any third set Z: "dependence separation" or "d-separation"

D(ependence)-separation: implies conditional independence in Bayes nets

- **Are X and Y conditionally independent given Z?**
- **Compute d-separation**
- **How to compute:**

1. Draw: the ancestral subgraph consisting of X, Y, Z and their ancestors
2. Moralize: Add links that between any unlinked pair of nodes that share a common child
3. Disorient: Replace all directed links by undirected links
4. Delete givens: remove Z and its edges
5. Inspect: If remaining variables are disconnected ⇒ X, Y are conditionally independent, given Z

How (not) to do Inference

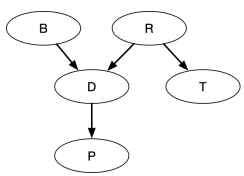
- So, we can reconstruct the probability of any <u>particular</u> scenario
- But, normally we want to know the probabilities of some nodes *given* that we have observed some others
  - E.g., what is the probability of a burglar given that the police were called and the trash can was *not* knocked over?
- By abuse of notation, we write a variable x to represent whatever its value is, and x^+, x^- if its value is known to be T or F (binary case)
  - $P(b^+|p^+,t^-) = \frac{P(b^+,p^+,t^-)}{P(b^+,t^-)}$

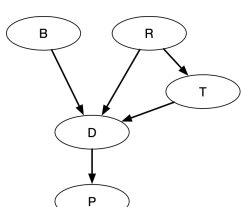
$$\frac{\sum_{d,r} P(p^+,d,b^+,t^-,r)}{\sum_{b,d,r} P(p^+,d,b,t^-,r)}$$

- Downside: exponential number of terms in the "don't care" variables

In simple cases, value propagation suffices (as in our example earlier)

- **Suppose we observe B**
  - **Reduce conditional probability table of its children (D) to the B=T or B=F cases**
  - **Propagate the result**
- **Suppose we observe P**
  - **Use Bayes' Rule to update D**
  - **Propagate the result**
- **Suppose we observe D**
  - **Do both of the above**
- **Because everything is singly connected, one pass updates all probabilities**
- **Much more complex if the network is multiply connected!**



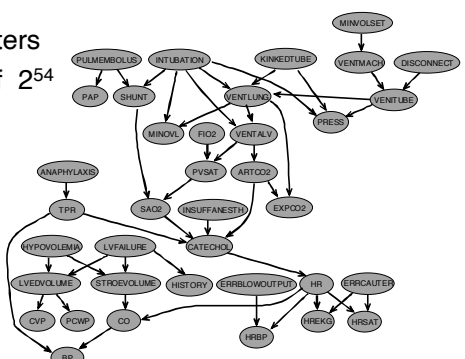


Example: "ICU Alarm" network

Domain: Monitoring Intensive-Care Patients

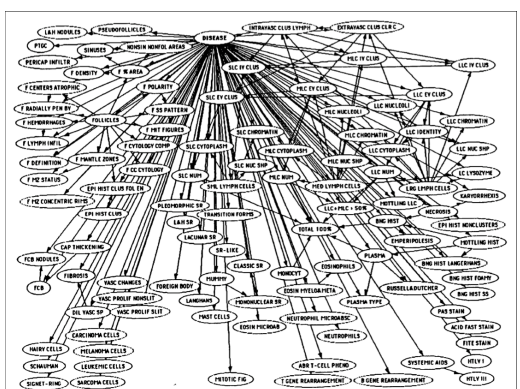
- 37 variables
- 509 parameters

  …instead of 2^{54}



A very large Bayes net

David Heckerman, *Pathfinder/Intellipath*, around 1990



Rules and Probabilities

- Many have wanted to put a probability on assertions and on rules, and compute with likelihoods
- E.g., Mycin's *certainty factor* framework
  - A (p=.3) & B (p=.7) ==(p=.8)==> C (p=?)
- Problems:
  - How to combine uncertainties of preconditions and of rule
  - How to combine evidence from multiple rules
- Theorem: There is NO such algebra that works when rules are considered independently
- Need Bayes Nets for a consistent model of probabilistic inference

Learning Bayes' nets

Three general problems:

1. Learn the numbers in the conditional probability tables – all data observable
2. Compare two network structures & their tables
3. Discover structure of network & associated tables

Problem 1 is much easier than Problem 2, and Problem 2 is *much* easier than Problem 3

Learning Bayes nets?

- Recall that a Bayes Network is fully specified by
  - –a DAG *G* that gives the (in)dependencies among variables
  - –the collection of parameters θ that define the conditional probability tables for each of the $P(x_i|Par(x_i))$
- Then

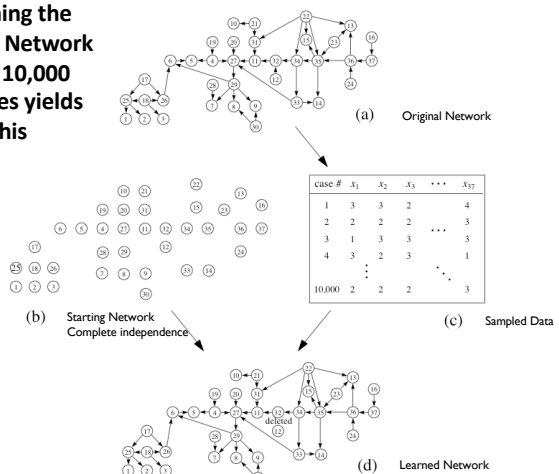
$$P(Graph|Data) \propto P(Data|Graph)P(Graph)$$

- Search over all graph structures

The **"**best" model is the one that predicts the highest probability for the data actually observed

- We get a complex search problem; usually try to penalize more complex graphs (by making their prior probabilities lower)

Learning the ALARM Network from 10,000 Samples yields this



Gold star ideas for probability

How to handle uncertainty– use probability & Bayes nets:

The right thing when you don't know anything – e.g., medicine
You often don't know anything, or, if you do, you don't know much
Locality and modularity imposed by conditional independence
"Prediction is hard, especially about the future" – Yogi Berra