Bayesian Networks

(aka Bayes Nets, Belief Nets, Directed Graphical Models)

Chapter 14.1, 14.2, and 14.4
plus optional paper "Bayesian networks without tears"

[based on slides by Jerry Zhu and Andrew Moore]

Introduction

- Probabilistic models allow us to use probabilistic inference (e.g., Bayes's rule) to compute the probability distribution over a set of unobserved ("hypothesis") given a set of observed variables
- Full joint probability distribution table is great for inference in an uncertain world, but is terrible to obtain and store
- Bayesian Networks allow us to represent joint distributions in manageable chunks using
  - Independence, conditional independence
- Bayesian Network can do any inference

Full Joint Probability Distribution

Making a joint distribution of *N* variables:

1. List all combinations of values (if each variable has *k* values, there are k^N combinations)
2. Assign each combination a probability
3. They should sum to 1

Weather	Temperature	Prob.
Sunny	Hot	150/365
Sunny	Cold	50/365
Cloudy	Hot	40/365
Cloudy	Cold	60/365
Rainy	Hot	5/365
Rainy	Cold	60/365

Using the Full Joint Distribution

- Once you have the joint distribution, you can do anything, e.g. marginalization:

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

- e.g., P(Sunny or Hot) = (150+50+40+5)/365

Convince yourself this is the same as P(sunny) + P(hot) - P(sunny and hot)

Weather	Temperature	Prob.
Sunny	Hot	150/365
Sunny	Cold	50/365
Cloudy	Hot	40/365
Cloudy	Cold	60/365
Rainy	Hot	5/365
Rainy	Cold	60/365

Using the Joint Distribution

- You can also do inference:

$$P(\text{Q} \mid \text{E}) = \frac{\sum_{\text{rows matching Q AND E}} P(\text{row})}{\sum_{\text{rows matching E}} P(\text{row})}$$

$P(\text{Hot} \mid \text{Rainy})$

Weather	Temperature	Prob.
Sunny	Hot	150/365
Sunny	Cold	50/365
Cloudy	Hot	40/365
Cloudy	Cold	60/365
Rainy	Hot	5/365
Rainy	Cold	60/365

The Bad News

- Full Joint distribution requires a lot of storage space
- For N variables, each taking k values, the joint distribution has k^N numbers (and $k^N - 1$ degrees of freedom)
- It would be nice to use fewer numbers …
- Bayesian Networks to the rescue!
  - **Provides a decomposed / factorized representation of the FJPD**
  - **Encodes a collection of conditional independence relations**

Bayesian Networks

- Idea: Represent statistical dependencies graphically
- Directed, acylic graphs (DAGs)
- Nodes = random variables
  - "CPT" stored at each node quantifies conditional probability of node's r.v. given *all* its parents
- Directed arc from A to B means A "**has a direct influence on**" or "**causes**" B
  - Evidence for A increases likelihood of B (***deductive*** influence from causes to effects)
  - Evidence for B increases likelihood of A (***abductive*** influence from effects to causes)
- Encodes conditional independence assumptions

Example

- A: your alarm sounds
- J: your neighbor John calls you
- M: your other neighbor Mary calls you
- John and Mary do not communicate (they promised to call you whenever they hear the alarm)

- What kind of independence do we have?
- What does the Bayes Net look like?

Conditional Independence

- Random variables can be *dependent*, but **conditionally independent**
- Example: Your house has an alarm
  - Neighbor John will call when he hears the alarm
  - Neighbor Mary will call when she hears the alarm
  - Assume John and Mary don't talk to each other
- Is *JohnCall independent* of *MaryCall*?
  - **No** – If John called, it is likely the alarm went off, which increases the probability of Mary calling
  - $P(MaryCall \mid JohnCall) \neq P(MaryCall)$

Conditional Independence

- But, if we *know* the status of the *alarm*, *JohnCall* will ***not*** affect whether or not Mary calls

  $P(MaryCall \mid Alarm, JohnCall) = P(MaryCall \mid Alarm)$
- We say *JohnCall* and *MaryCall* are **conditionally independent** given *Alarm*
- In general, "*A* and *B* are conditionally independent given *C*" means:

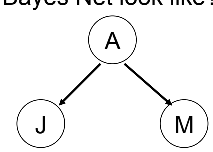
  $P(A \mid B, C) = P(A \mid C)$

  $P(B \mid A, C) = P(B \mid C)$

  $P(A, B \mid C) = P(A \mid C)\, P(B \mid C)$

Example

- A: your alarm sounds
- J: your neighbor John calls you
- M: your other neighbor Mary calls you
- John and Mary do not communicate (they promised to call you whenever they hear the alarm)
- What kind of independence do we have?
  - **Conditional independence**: $P(J,M|A)=P(J|A)P(M|A)$
- What does the Bayes Net look like?

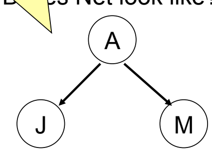


Our BN: $P(A,J,M) = P(A)\, P(J|A)\, P(M|A)$
Chain rule: $P(A,J,M) = P(A)\, P(J|A)\, P(M|A,J)$

Our BN assumes conditional independence, so $P(M|A,J) = P(M|A)$

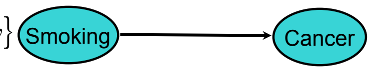
omised

- What ki… endence do we have?
  - Condition… pendence $P(J,M|A)=P(J|A)P(M|A)$
- What does the B… es Net look like?



A Simple Bayesian Network

$S \in \{no, light, heavy\}$



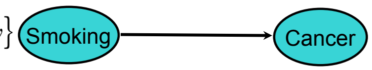
$C \in \{none, benign, malignant\}$

P(S=no)	0.80
P(S=light)	0.15
P(S=heavy)	0.05

	no	light	heavy
P(C=none|S=)	0.96	0.88	0.60
P(C=benign|S=)	0.03	0.08	0.25
P(C=malig|S=)	0.01	0.04	0.15

A Simple Bayesian Network

$S \in \{no, light, heavy\}$



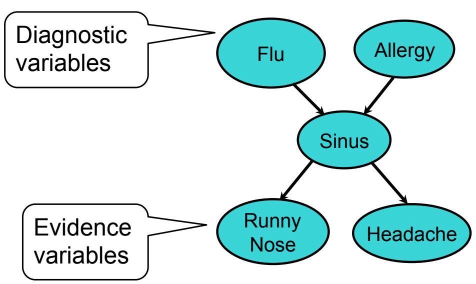
$C \in \{none, benign, malignant\}$

P(S=no)	0.80
P(S=light)	0.15
~~*P*(S=heavy)~~	~~0.05~~

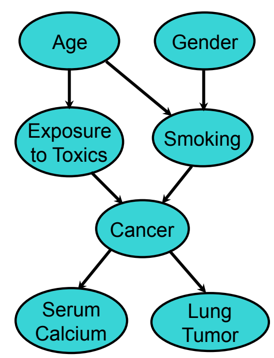
Don't need to store

	no	light	heavy
P(C=none|S=)	0.96	0.88	0.60
P(C=benign|S=)	0.03	0.08	0.25
~~*P*(C=malig|S=)~~	~~0.01~~	~~0.04~~	~~0.15~~

A Bayesian Network



A Bayesian Network

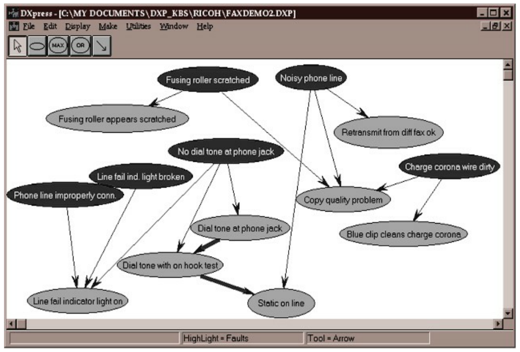


Applications

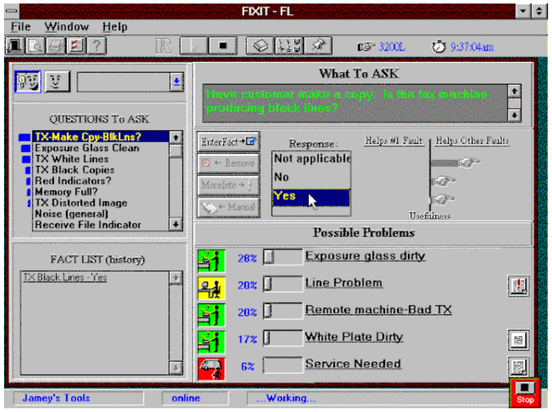
- Medical diagnosis systems
- Manufacturing system diagnosis
- Computer systems diagnosis
- Network systems diagnosis
- Helpdesk troubleshooting
- Information retrieval
- Customer modeling

RICOH Fixit

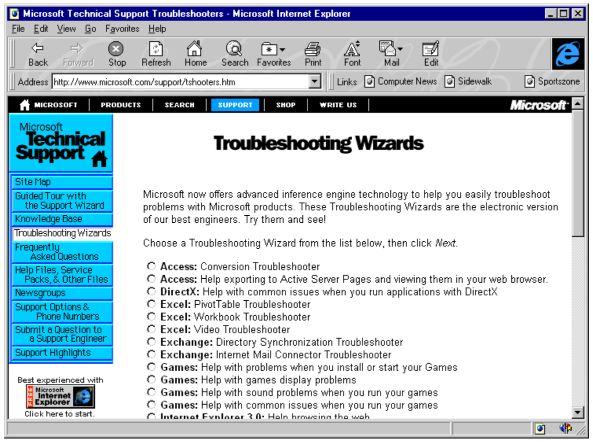
- Diagnostics and information retrieval



FIXIT: Ricoh copy machine



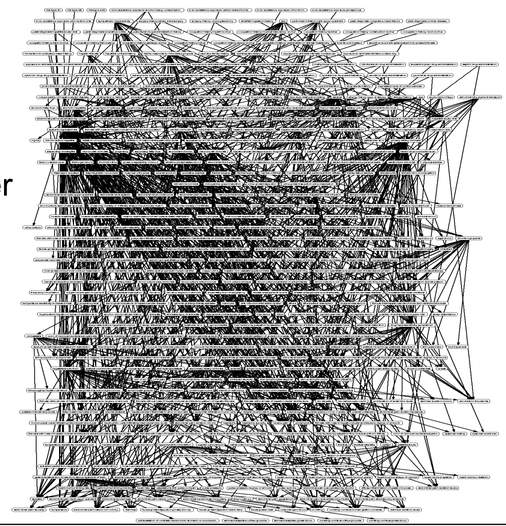
Online Troubleshooters



Pathfinder

- Pathfinder was one of the first BN systems
- It performed diagnosis of lymph-node diseases
- It dealt with over 60 diseases and 100 symptoms and test results
- 14,000 probabilities
- Commercialized and applied to about 20 tissue types

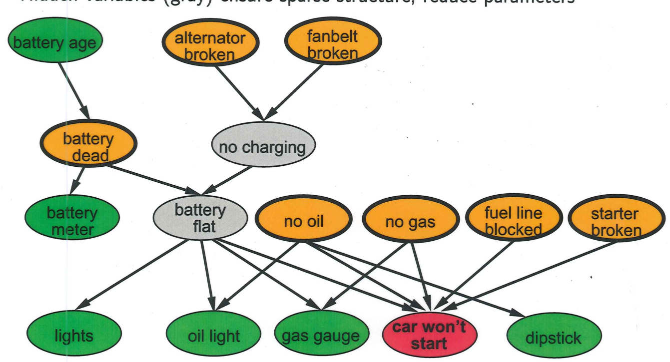
Pathfinder Bayes Net



448 nodes, 906 arcs

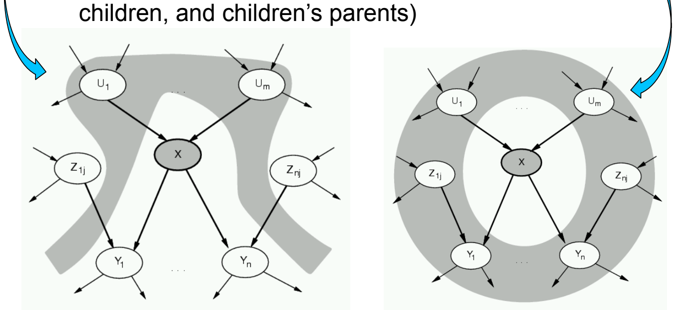
Example: Car diagnosis

Initial evidence: car won't start
Testable variables (green), "broken, so fix it" variables (orange)
Hidden variables (gray) ensure sparse structure, reduce parameters

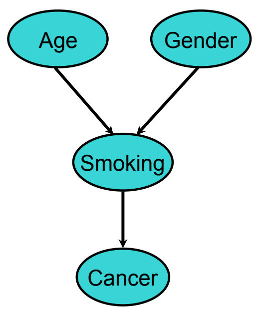


Conditional Independence in Bayes Nets

- **A node is conditionally independent of its non-descendants, given its parents**
- A node is conditionally independent of all other nodes, given its "Markov blanket" (i.e., its parents, children, and children's parents)

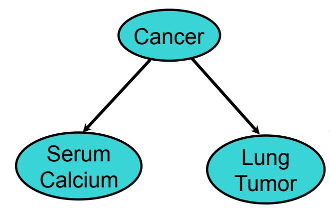


Conditional Independence



Cancer is conditionally independent of *Age* and *Gender* given *Smoking*

More Conditional Independence

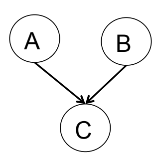


Serum Calcium is conditionally independent of *Lung Tumor*, given *Cancer*

$$P(L \mid SC, C) = P(L \mid C)$$

Interpreting Bayesian Nets

- 2 nodes are (**unconditionally) independent** if there's *no undirected path* between them
- If there's an undirected path between 2 nodes, then whether or not they are independent or dependent depends on what other evidence is known



A and B are independent given nothing else, but are dependent given C

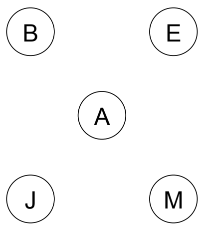
Example with 5 Variables

- B: there's burglary in your house
- E: there's an earthquake
- A: your alarm sounds
- J: your neighbor John calls you
- M: your other neighbor Mary calls you

- B, E are **independent**
- J is directly influenced by only A (i.e., J is **conditionally independent** of B, E, M, given A)
- M is directly influenced by only A (i.e., M is **conditionally independent** of B, E, J, given A)

Creating a Bayes Net

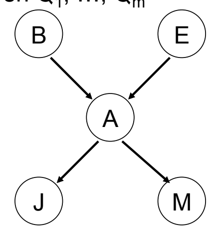
- **Step 1**: Add variables. Choose the variables you want to include in the Bayes Net



B: there's burglary in your house
E: there's an earthquake
A: your alarm sounds
J: your neighbor John calls you
M: your other neighbor Mary calls you

Creating a Bayes Net

- **Step 2**: Add directed edges
  - The graph must be **acyclic**
  - If node X is given parents $Q_1, \ldots, Q_m$, you are promising that any variable that's ***not*** a *descendant* of X is conditionally independent of X given $Q_1, \ldots, Q_m$

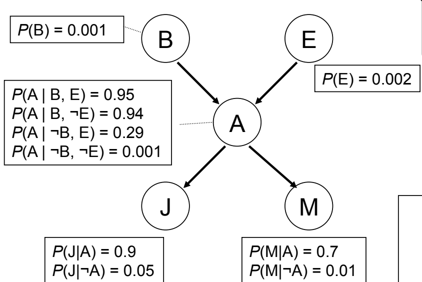


B: there's burglary in your house
E: there's an earthquake
A: your alarm sounds
J: your neighbor John calls you
M: your other neighbor Mary calls you

Creating a Bayes Net

- **Step 3**: Add CPT's
- Each table must list $P(X \mid \text{Parent values})$ for all combinations of parent values

e.g., you must specify $P(J|A)$ AND $P(J|\neg A)$ since they don't have to sum to 1!



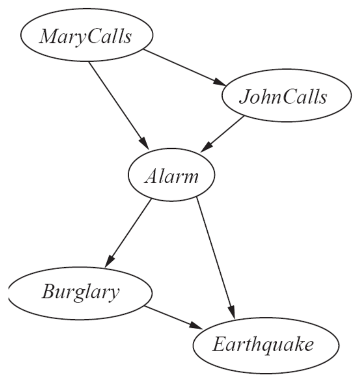
B: there's burglary in your house
E: there's an earthquake
A: your alarm sounds
J: your neighbor John calls you
M: your other neighbor Mary calls you

Creating a Bayes Net

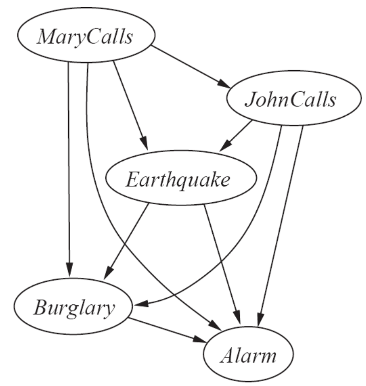
1. Choose a set of relevant variables
2. Choose an ordering of them, call them $x_1, \ldots, x_N$
3. for $i = 1$ to N:
   1. Add node x_i to the graph
   2. Set parents(x_i) to be the minimal subset of $\{x_1 \ldots x_{i-1}\}$, such that x_i is conditionally independent of all other members of $\{x_1 \ldots x_{i-1}\}$ given parents(x_i)
   3. Define the CPT's for P(x_i | assignments of parents(x_i))

- Different ordering leads to different graph, in general
- Best ordering when each variable is considered *after* all variables that directly influence it

The Bayesian Network Created from a Different Variable Ordering



The Bayesian Network Created from a Different Variable Ordering

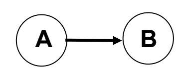


Compactness of Bayes Nets

- **A Bayesian Network is a graph structure for representing conditional independence relations in a compact way**
- A Bayes net encodes the full joint distribution, often with far less parameters (i.e., numbers)
- A full joint table needs k^N parameters (N variables, k values per variable)
  - grows exponentially with N
- If the Bayes net is sparse, e.g., each node has at most M parents ($M << N$), only needs $O(Nk^M)$ parameters
  - grows linearly with N
  - can't have too many parents, though

Variable Dependencies

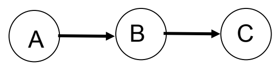
- Directed arc from one variable to another variable



- Is A guaranteed to be *independent* of B?
  - No – Information can be transmitted over 1 arc
    - Example: My knowing the Alarm went off, increases my belief there has been a Burglary, and similarly, my knowing there has been a Burglary increases my belief the Alarm went off

Causal Chain

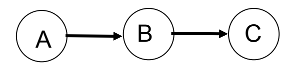
- This local configuration is called a “**causal chain**:”



- Is A guaranteed to be *independent* of C?
  - No – Information can be transmitted between A and C through B if B is *not* observed
    - Example: Not knowing Alarm means that my knowing that a Burglary has occurred increases my belief that Mary calls, and similarly, knowing that Mary Calls increases my belief that there has been a Burglary

Causal Chain

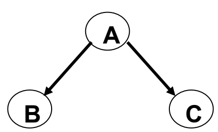
- This local configuration is called a “**causal chain**:”



- Is A *independent* of C *given B*?
  - Yes – Once B is observed, information *cannot* be transmitted between A and C through B; B “blocks” the information path; “C is conditionally independent of A given B”
    - Example: Knowing that the Alarm went off means that also knowing that a Burglary has taken place will **not** increase my belief that Mary Calls

Common Cause

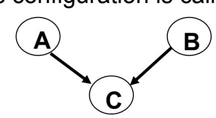
- This configuration is called “**common cause**:”



- Is it guaranteed that B and C are *independent*?
  - No – Information can be transmitted through A to the children of A if A is *not* observed
- Is it guaranteed that B and C are independent *given A*?
  - Yes – Observing the cause, A, blocks the influence between effects B and C; “B is conditionally independent of C given A”

Common Effect

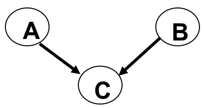
- This configuration is called “**common effect**:”



- Are A and B *independent*?
  - Yes
    - Example: Burglary and Earthquake cause the Alarm to go off, but they are not correlated
  - Proof: $P(a,b) = \Sigma_c P(a,b,c)$ by marginalization
    $= \Sigma_c P(a) P(b|a) P(c|a,b)$ by chain rule
    $= \Sigma_c P(a) P(b) P(c|a,b)$ by cond. indep.
    $= P(a) P(b) \Sigma_c P(c|a,b)$
    $= P(a) P(b)$ since last term = 1

Common Effect

- This configuration is called "**common effect**:"



- Are A and B independent *given C*?
  - No – Information can be transmitted through C among the parents of C if C *is* observed
    - Example: If I already know that the Alarm went off, my further knowing that there has been an Earthquake, *decreases* my belief that there has been a Burglary. Called "explaining away."
  - Similarly, if C has descendant D and D is given, then A and B are *not* independent

D-Separation

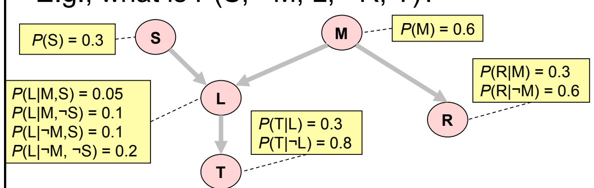
Determining if two variables in a Bayesian Network are independent or conditionally independent given a set of observed evidence variables, is determined using "**d-separation**"

D-separation is covered in CS 760

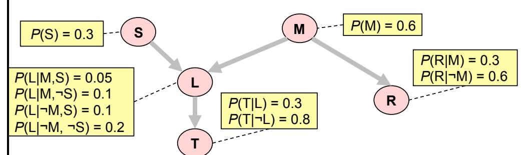
Computing a Joint Entry from a Bayes Net

How to compute an entry in the joint distribution?

E.g., what is P(S, ¬M, L, ¬R, T)?



Computing with Bayes Net



Apply the Chain Rule + conditional independence!

P(T, ¬R, L, ¬M, S)
= *P*(T | ¬R, L, ¬M, S) * *P*(¬R, L, ¬M, S)
= *P*(T | L) * *P*(¬R, L, ¬M, S)
= *P*(T | L) * *P*(¬R | L, ¬M, S) * *P*(L, ¬M, S)
= *P*(T | L) * *P*(¬R | ¬M) * *P*(L, ¬M, S)
= *P*(T | L) * *P*(¬R | ¬M) * *P*(L | ¬M, S) * *P*(¬M, S)
= *P*(T | L) * *P*(¬R | ¬M) * *P*(L | ¬M, S) * *P*(¬M | S) * *P*(S)
P(T | L) * *P*(¬R | ¬M) * *P*(L | ¬M, S) * *P*(¬M) * *P*(S)

Variable Ordering

Before applying chain rule, best to reorder all of the variables, listing first the leaf nodes, then all the parents of the leaves, etc. Last variables listed are those that have no parents, i.e., the root nodes.

So, for previous example,
P(S,L,M,T,R) = P(T,R,L,S,M)

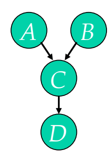
The General Case

$P(X_1=x_1, X_2=x_2, \ldots, X_{n-1}=x_{n-1}, X_n=x_n)$
$= P(X_n=x_n, X_{n-1}=x_{n-1}, \ldots, X_2=x_2, X_1=x_1)$
$= P(X_n=x_n \mid X_{n-1}=x_{n-1}, \ldots, X_2=x_2, X_1=x_1) * P(X_{n-1}=x_{n-1}, \ldots, X_2=x_2, X_1=x_1)$
$= P(X_n=x_n \mid X_{n-1}=x_{n-1}, \ldots, X_2=x_2, X_1=x_1) * P(X_{n-1}=x_{n-1} \mid \ldots X_2=x_2, X_1=x_1) *$
$P(X_{n-2}=x_{n-2}, \ldots, X_2=x_2, X_1=x_1)$
:

$$= \prod_{i=1}^{n} P\Big(\big(X_i = x_i\big) \;\Big|\; \big(\big(X_{i-1} = x_{i-1}\big), \;\ldots,\; \big(X_1 = x_1\big)\big)\Big)$$

$$= \prod_{i=1}^{n} P\Big(\big(X_i = x_i\big) \;\Big|\; \text{Assignments of Parents}\big(X_i\big)\Big)$$

Computing Joint Probabilities using a Bayesian Network



How is ***any*** joint probability computed?

Sum the relevant joint probabilities:

Compute: $P(a,b)$
$= P(a,b,c,d) + P(a,b,c,\neg d) + P(a,b,\neg c,d) + P(a,b,\neg c,\neg d)$

Compute: $P(c)$
$= P(a,b,c,d) + P(a,\neg b,c,d) + P(\neg a,b,c,d) + P(\neg a,\neg b,c,d) +$
$P(a,b,c,\neg d) + P(a,\neg b,c,\neg d) + P(\neg a,b,c,\neg d) + P(\neg a,\neg b,c,\neg d)$

- A BN can answer *any* query (i.e., probability) about the domain by marginalization ("summing out") over the relevant joint probabilities

Where Are We Now?

- We defined a Bayes net, using small number of parameters, to describe the joint probability
- Any joint probability can be computed as

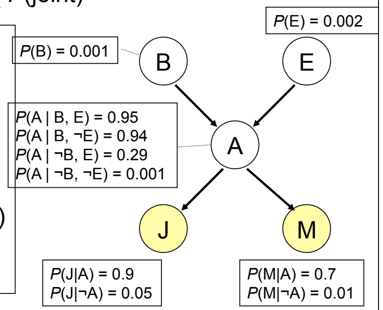
  $P(x_1,\ldots, x_N) = \prod_i P(x_i \mid \text{parents}(x_i))$
- The above joint probability can be computed in time linear in the number of nodes, N
- With this joint distribution, we can compute *any* conditional probability, $P(Q \mid E)$; thus we can perform any inference
- How?

Inference by Enumeration

$$P(\text{Q} \mid \text{E}) = \frac{\sum_{\text{joint matching Q AND E}} P(\text{joint})}{\sum_{\text{joint matching E}} P(\text{joint})}$$ by def. of cond. prob.

For example: P(B | J, ¬M)

1. Compute P(B,J, ¬M)
2. Compute P(J, ¬M)
3. Return P(B,J, ¬M)/P(J, ¬M)



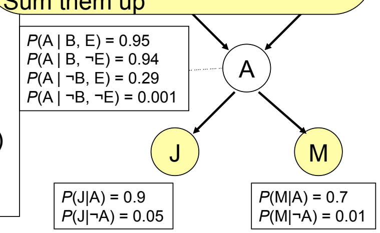
Inference by Enumeration

$$P(\text{Q} \mid \text{E}) = \frac{\sum_{\text{joint matching Q}} \ldots}{\sum_{\text{joint matchi}} \ldots}$$

Compute the joint (4 of them)
P(B, J, ¬M, A, E)
P(B, J, ¬M, A, ¬E)
P(B, J, ¬M, ¬A, E)
P(B, J, ¬M, ¬A, ¬E)
Each is O(N) for sparse graph
$P(x_1,\ldots x_N) = \prod_i P(x_i \mid \text{parents}(x_i))$
Sum them up

For example: P(B | J,¬M)

1. Compute P(B,J,¬M)
2. Compute P(J,¬M)
3. Return P(B,J,¬M)/P(J,¬M)



Inference by Enumeration

$$P(\text{Q} \mid \text{E}) = \frac{\sum_{\text{joint matching Q}} \ldots}{\sum_{\text{joint matchin}} \ldots}$$

Compute the joint (8 of them)
P(J,¬M,B,A,E)
P(J,¬M,B,A,¬E)
P(J,¬M,B,¬A,E)
P(J,¬M,B,¬A,¬E)
P(J,¬M,¬B,A,E)
P(J,¬M,¬B,A,¬E)
P(J,¬M,¬B,¬A,E)
P(J,¬M,¬B,¬A,¬E)
Each is O(N) for sparse graph
$P(x_1,\ldots x_N) = \prod_i P(x_i \mid \text{parents}(x_i))$
Sum them up

For example: P(B | J,¬M)

1. Compute P(B,J,¬M)
2. Compute P(J,¬M)
3. Return P(B,J,¬M)/P(J,

Inference by Enumeration

$$P(\text{Q} \mid \text{E}) = \frac{\sum_{\text{joint matching Q AND E}} P(\text{joint})}{\sum_{\text{joint matching E}} P(\text{joint})}$$

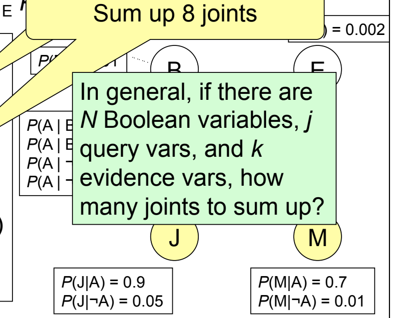
Sum up 4 joints

Sum up 8 joints

For example: P(B | J,¬M)

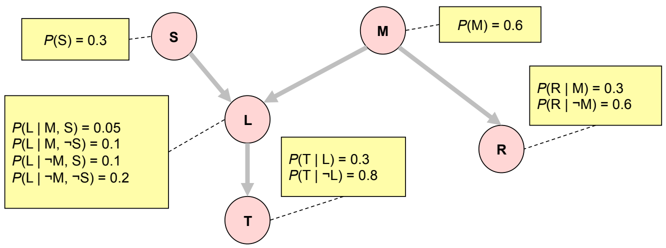
1. Compute P(B,J,¬M)
2. Compute P(J,¬M)
3. Return P(B,J,¬M)/P(J,¬M)

In general, if there are N Boolean variables, j query vars, and k evidence vars, how many joints to sum up?



Another Example

Compute $P(R \mid T, \neg S)$ from the following Bayes Net



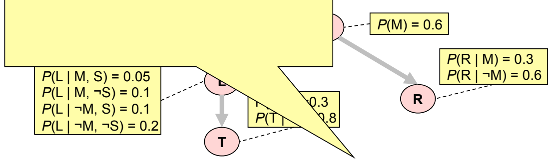
Another Example

Step 1: Compute $P(R, T, \neg S)$

Step 2: Compute $P(T, \neg S)$

Step 3: Return

$$\frac{P(R, T, \neg S)}{P(T, \neg S)}$$



Compute $P(R \mid T, \neg S)$

Another Example

Step 1: Compute $P(R, T, \neg S)$

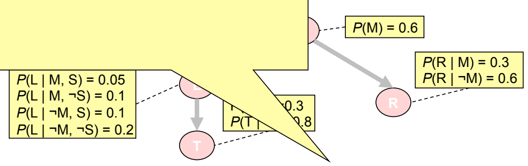
Sum of all the rows in the Joint that match $R \wedge T \wedge \neg S$

Step 2: Compute $P(T, \neg S)$

Sum of all the rows in the Joint that match $T \wedge \neg S$

Step 3: Return

$$\frac{P(R, T, \neg S)}{P(T, \neg S)}$$



Compute $P(R \mid T, \neg S)$

Another Example

4 joint computes

Step 1: Compute $P(R, T, \neg S)$

Sum of all the rows in the Joint that match $R \wedge T \wedge \neg S$

Step 2: Compute $P(T, \neg S)$

Sum of all the rows in the Joint that match $T \wedge \neg S$

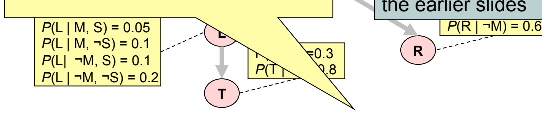
8 joint computes

Step 3: Return

$$\frac{P(R, T, \neg S)}{P(R, T, \neg S) + P(\neg R, T, \neg S)}$$

$= P(T, \neg S)$

Each of these obtained by the "computing a joint probability entry" method in the earlier slides



Compute $P(R \mid T, \neg S)$

- Inference through a Bayes Net can go both "forward" and "backward" through arcs
- **Causal** (top-down) inference
  - Given a cause, infer its effects
  - E.g., *P*(T | S)
- **Diagnostic** (bottom-up) inference
  - Given effects/symptoms, infer a cause
  - E.g., *P*(S | T)

The Good News

We can do inference. That is, we can compute ***any*** conditional probability:

P(Some variable | Some other variable values)

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{joint entries matching } E_1 \text{ and } E_2} P(\text{joint entry})}{\sum_{\text{joint entries matching } E_2} P(\text{joint entry})}$$

"**Inference by Enumeration**" Algorithm

The Bad News

- In general if there are *N* variables, while evidence contains *j* variables, and each variable has *k* values, how many joints to sum up? $k^{(N-j)}$
- It is this summation that makes **inference by enumeration** inefficient
  - Computing conditional probabilities by enumerating all matching entries in the joint is expensive: **Exponential in the number of variables**
- Some computation can be saved by carefully ordering the terms and re-using intermediate results (**variable elimination algorithm**)
- A more complex algorithm called a **join tree** (junction tree) can save even more computation
- But, even so, **exact inference with an arbitrary Bayes Net is NP-Complete**

Variable Elimination Algorithm

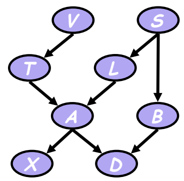
General idea:

- Write query in the form

$$P(x_n, \boldsymbol{e}) = \sum_{x_k} \cdots \sum_{x_3} \sum_{x_2} \prod_i P(x_i \mid pa_i)$$

- Iteratively
  - Move all irrelevant terms outside of innermost sum
  - Perform innermost sum, getting a new term
  - Insert the new term into the product

Compute $P(d)$



Need to eliminate: *v, s, x, t, l, a, b*

Initial factors:

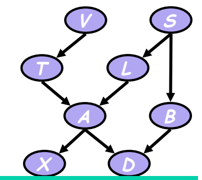
$P(v, s, t, l, a, b, x, d) =$

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

Inference by Enumeration (i.e., brute force) approach:

$$P(d) = \sum_x \sum_b \sum_a \sum_l \sum_t \sum_s \sum_v P(v,s,t,l,a,b,x,d)$$

- We want to compute $P(d)$
- Need to eliminate: *v, s, x, t, l, a, b*



Initial factors

$$P(v)P(s)P(t|v)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

Eliminate: *v*

Compute: $f_v(t) = \sum_v P(v)P(t|v)$

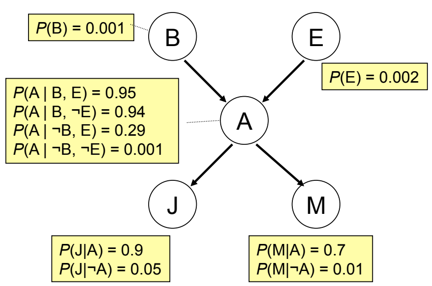
$$\Rightarrow f_v(t)P(s)P(l|s)P(b|s)P(a|t,l)P(x|a)P(d|a,b)$$

Note: $f_v(t) = P(t)$

Idea behind Variable Elimination Algorithm

Parameter (CPT) Learning for BN

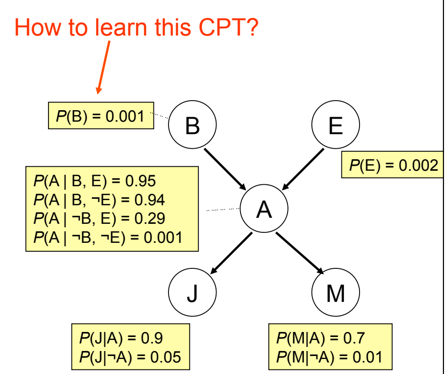
- Where do you get these CPT numbers?
  - Ask domain experts, or
  - Learn from data



Parameter (CPT) Learning for BN

- Learn from a data set like this:

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

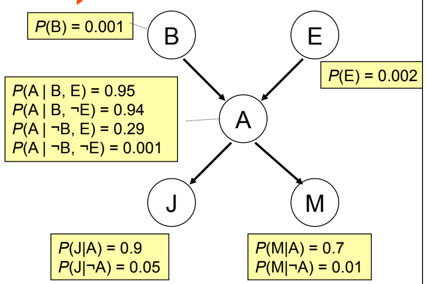


Parameter (CPT) Learning for BN

- Learn from a data set like this:

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

Count #(B) and #(¬B) in dataset.
$P(\mathrm{B}) = \#(\mathrm{B}) / [\#(\mathrm{B}) + \#(\neg \mathrm{B})]$

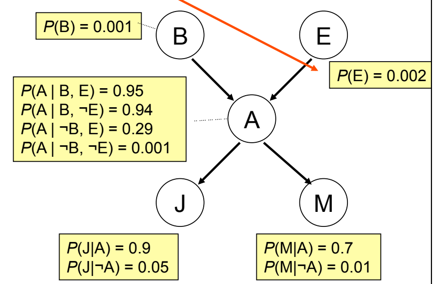


Parameter (CPT) Learning for BN

- Learn from a data set like this:

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

Count #(E) and #(¬E) in dataset.
$P(\mathrm{E}) = \#(\mathrm{E}) / [\#(\mathrm{E}) + \#(\neg \mathrm{E})]$

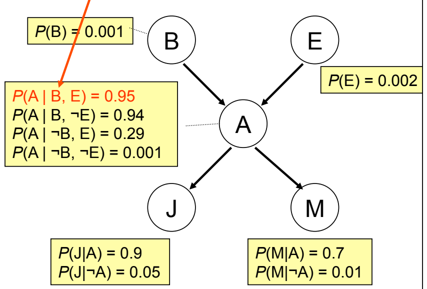


Parameter (CPT) Learning for BN

- Learn from a data set like this:

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

Count #(A) and #(¬A) in dataset
where B=true and E=true.
$P(\mathrm{A}|\mathrm{B},\mathrm{E}) = \#(\mathrm{A}) / [\#(\mathrm{A}) + \#(\neg \mathrm{A})]$

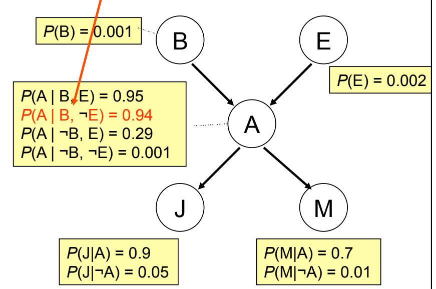


Parameter (CPT) Learning for BN

- Learn from a data set like this:

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

Count #(A) and #(¬A) in dataset
where B=true and E=false.
$P(\mathrm{A}|\mathrm{B},\neg \mathrm{E}) = \#(\mathrm{A}) / [\#(\mathrm{A}) + \#(\neg \mathrm{A})]$

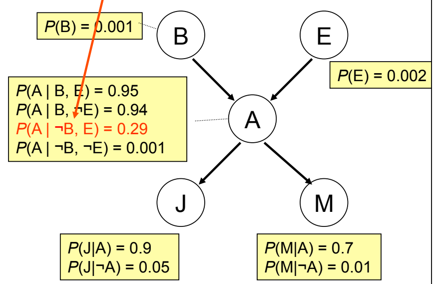


Parameter (CPT) Learning for BN

- Learn from a data set like this:

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

Count #(A) and #(¬A) in dataset
where B=false and E=true.
P(A|¬B,E) = #(A) / [#(A) + #(¬A)]

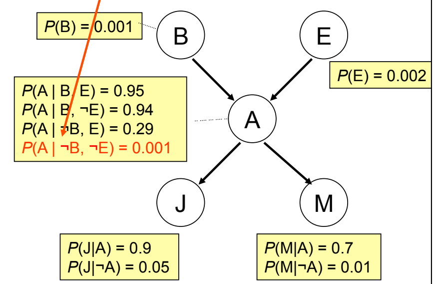


Parameter (CPT) Learning for BN

- Learn from a data set like this:

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

Count #(A) and #(¬A) in dataset
where B=false and E=false.
P(A|¬B, ¬E) = #(A) / [#(A) + #(¬A)]



Parameter (CPT) Learning for BN

- **'Unseen event' problem**

(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(B, ¬E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, J, ¬M)
(¬B, E, A, J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(¬B, ¬E, ¬A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
(~B, ~E, ¬A, ¬J, ¬M)
(B, E, A, ¬J, M)
(¬B, ¬E, ¬A, ¬J, ¬M)
…

Count #(A) and #(¬A) in dataset
where B=true and E=true.
P(A|B,E) = #(A) / [#(A) + #(¬A)]

What if there's ***no*** row with
(B, E, ¬A, *, *) in the dataset?

Do you want to set
P(A|B,E) = 1
P(¬A|B,E) = 0?

Why or why not?

Parameter (CPT) Learning for BN

- *P*(*X*=*x* | parents(*X*)) = (frequency of *x* given parents) is called the Maximum Likelihood (ML) estimate
- ML estimate is vulnerable to the 'unseen event' problem when the dataset is small
  - flip a coin 3 times, all heads → one-sided coin?
- Simplest solution: **'Add one' smoothing**

Smoothing CPTs

- 'Add one' smoothing: **add 1 to all counts**
- In the previous example, count #(A) and #(¬A) in dataset where B=true and E=true
  - *P*(A|B,E) = [#(A)**+1**] / [#(A)**+1** + #(¬A)**+1**]
  - If #(A)=1, #(¬A)=0:
    - without smoothing *P*(A|B,E) = 1, *P*(¬A|B,E) = 0
    - with smoothing *P*(A|B,E) = 0.67, *P*(¬A|B,E) = 0.33
  - If #(A)=100, #(¬A)=0:
    - without smoothing *P*(A|B,E) = 1, *P*(¬A|B,E) = 0
    - with smoothing *P*(A|B,E) = 0.99, *P*(¬A|B,E) = 0.01
- Smoothing saves you when you don't have enough data, and hides away when you do
- It's a form of Maximum *a posteriori* (MAP) estimation

Naïve Bayes Classifier

- Find *v* = $\text{argmax}_v P(Y = v) \prod_{i=1}^{n} P(X_i = u_i | Y = v)$

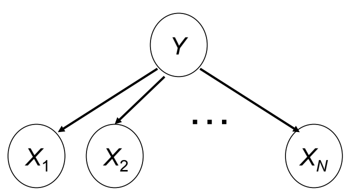
Class variable

Evidence variable

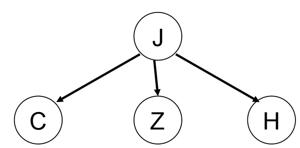
- Assumes all evidence variables are conditionally independent of each other given the class variable
- Robust since it gives the right answer as long as the correct class is more likely than all others

BN Special Case: Naïve Bayes

- A special Bayes Net structure:
  - a 'class' variable *Y* at root, compute $P(Y \mid X_1, \ldots, X_N)$
  - evidence nodes X_i (observed features) are all leaves
  - conditional independence between all evidence assumed. Usually not valid, but often empirically OK



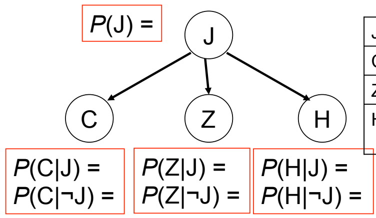
A Special BN: Naïve Bayes Classifiers



J	Person is Junior
C	Brought coat to class
Z	Lives in zipcode 53706
H	Saw "Hunger Games 1" more than once

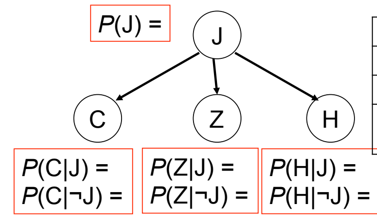
- What's stored in the CPTs?

A Special BN: Naïve Bayes Classifiers



J	Person is Junior
C	Brought coat to class
Z	Lives in zipcode 53706
H	Saw "Hunger Games 1" more than once

A Special BN: Naïve Bayes Classifiers



J	Person is Junior
C	Brought coat to class
Z	Lives in zipcode 53706
H	Saw "Hunger Games 1" more than once

- A new person shows up in class wearing an "I live in Union South where I saw the 'Hunger Games 1' every night" coat.
- What's the probability that the person is a Junior?

Is the Person a Junior?

- Input (evidence): C, Z, H
- Output (query): J

$P(J|C,Z,H)$

$= P(J,C,Z,H) / P(C,Z,H)$ by def. of cond. prob.

$= P(J,C,Z,H) / [P(J,C,Z,H) + P(\neg J,C,Z,H)]$ by marginalization

where

$P(J,C,Z,H) = P(J)P(C|J)P(Z|J)P(H|J)$ by chain rule and conditional independence associated with Bayes Net

$P(\neg J,C,Z,H) = P(\neg J)P(C|\neg J)P(Z|\neg J)P(H|\neg J)$

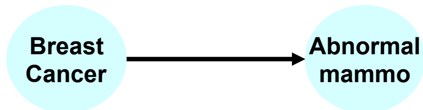
Smoothing CPTs for Naïve Bayes

- "Add 1 Smoothing" ensures that each conditional probability > 0
- Assume *c* possible classes (i.e., class variable has *c* possible values) and a "**bag of words**" model for describing each example ("document"): If there are *k* distinct token *types* in the vocabulary, v_1, …, v_k, each example is represented by a vector of length *k* where the i^{th} entry is the number of times word *i* occurs in the example
- Let n_{ci} = number of times token type v_i occurs in *all* training examples in class *c*, including multiple occurrences in the *same* training example
- Let n_c = total number of tokens in all examples in class *c*
- Compute conditional probabilities as:

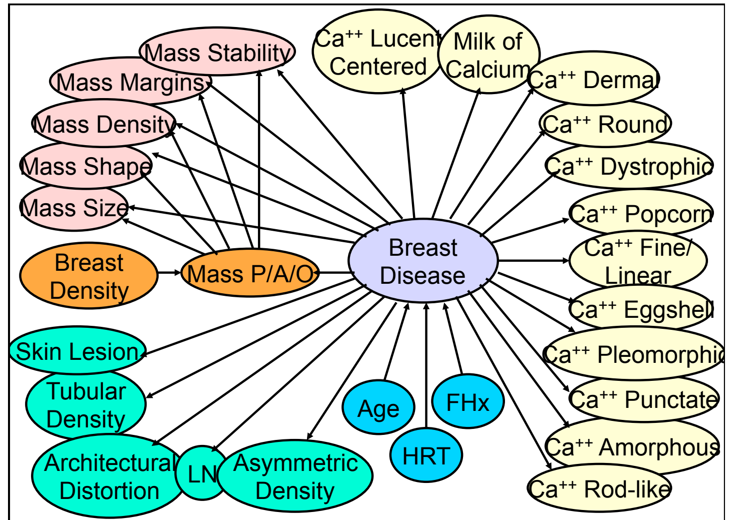
$$P(v_i \mid c) = \frac{n_{ci} + 1}{n_c + k}$$

- Note: $\sum_{i=1}^{k} P(v_i \mid c) = 1$

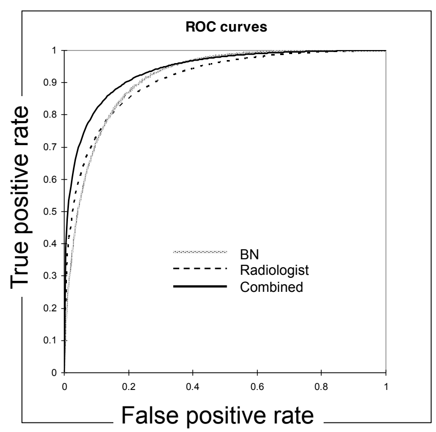
Application: Bayesian Networks for Breast Cancer Diagnosis



Elizabeth S. Burnside
Department of Radiology
University of Wisconsin Hospitals



Results: ROC Curves



Radiologist
AUC = .916

Bayes Net
AUC = .919

Bayesian Network Properties

- Bayesian Networks compactly encode joint distributions
- Topology of a Bayesian Network is only guaranteed to encode conditional independencies
  - Arcs do *not* necessarily represent causal relations

What You Should Know

- Inference with joint distribution
- Problems of joint distribution
- Bayesian Networks: representation (nodes, arcs, CPT) and meaning
- Compute joint probabilities from Bayes net
- Inference by enumeration
- Naïve Bayes