Bayesian Networks

(aka Bayes Nets, Belief Nets)

(one type of Graphical Model)

[based on slides by Jerry Zhu and Andrew Moore]

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:

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

- 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:

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

P(Hot | Rainy)

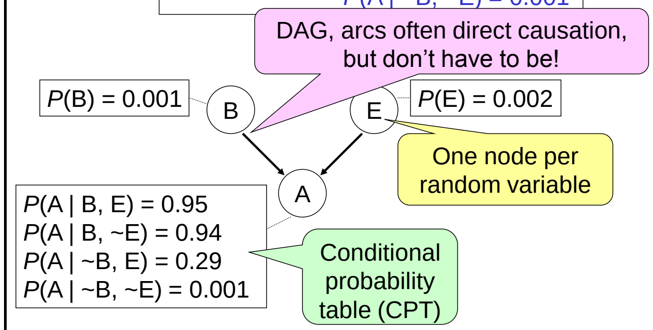
Weather	Temperature	Prob.
Sunny	Hot	150/365
Sunny	Cold	50/365
Cloudy	Hot	40/365
Cloudy	Cold	60/365
Rainy	Hot	5/365
Rainy	Cold	60/365

The Bad News

- 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 representation of the FJPD
  - Encodes a collection of conditional independence relations

Introducing Bayesian Networks

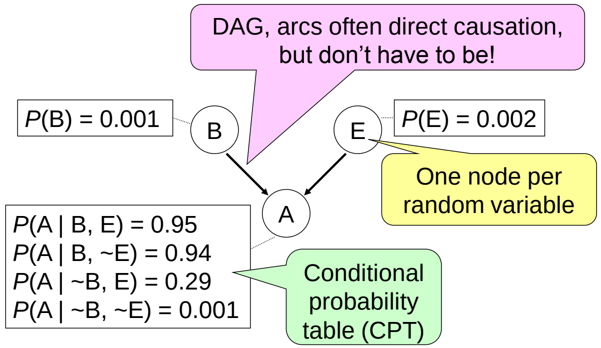
P(B) = 0.001	*P*(A | B, E) = 0.95
P(E) = 0.002	*P*(A | B, ~E) = 0.94
P(B | E) = *P*(B)	*P*(A | ~B, E) = 0.29
	P(A | ~B, ~E) = 0.001



Joint Probability from Bayes Net

$$P(x_1, \ldots x_N) = \prod_i P(x_i \mid \text{parents}(x_i))$$

- Example: *P*(~B, E, ~A) = *P*(~B) *P*(E) *P*(~A | ~B, E)



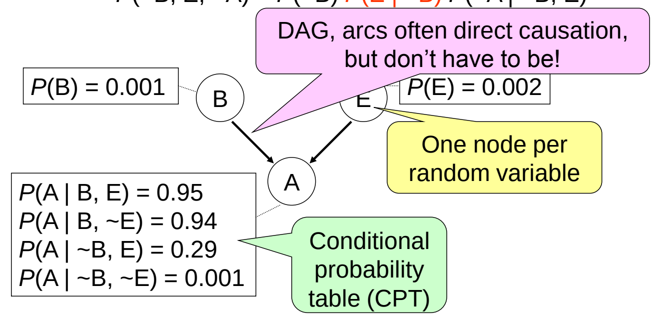
… with Bayes Net

Our B.N. has this independence assumption

…$(x_i \mid \text{parents}(x_i))$)

- Exa… *P*(~B) *P*(E) *P*(~A | ~B, E)
- Recall the chain rule:

  P(~B, E, ~A) = *P*(~B) *P*(E | ~B) *P*(~A | ~B, E)



Bayesian Networks

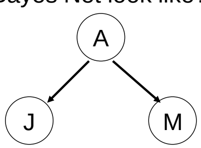
- Directed, acylic graphs (DAGs)
- Nodes = random variables
  - CPT stored at each node quantifies conditional probability of node's r.v. given all its parents
- Arc from A to B means A "has a **direct influence** on" or "causes" B
  - Evidence for A increases likelihood of B (***deductive*** influence 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?

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?

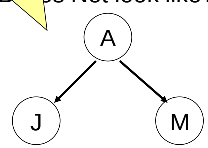


Our BN: $P(A,J,M) = P(A)\ P(J|A)\ P(M|A)$
Chain rule: $P(A,J,M) = P(A)\ P(J|A)\ P(M|A,J)$

Our B.N. 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

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

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

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

	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

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

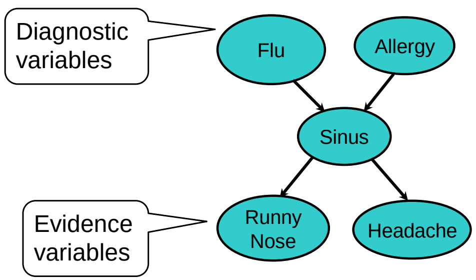
P(S=no)	0.80
P(S=light)	0.15
~~*P*(S=heavy)~~	~~0.05~~

Not needed

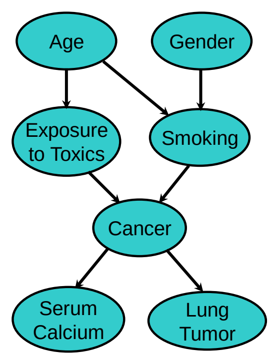
$C \in \{none, benign, malignant\}$

	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

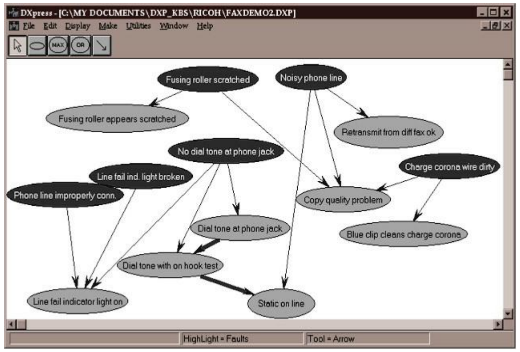


Applications

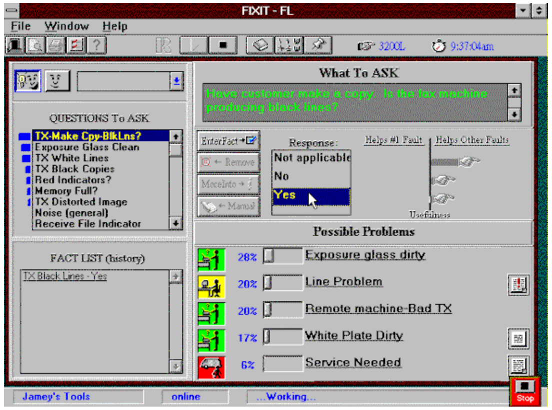
- Medical diagnosis systems
- Manufacturing system diagnosis
- Computer systems diagnosis
- Network systems diagnosis
- Helpdesk troubleshooting
- Information retrieval
- Customer modeling

RICOH Fixit

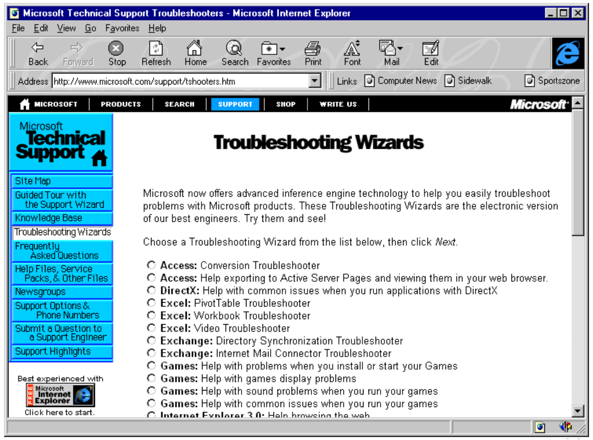
- Diagnostics and information retrieval



FIXIT: Ricoh copy machine



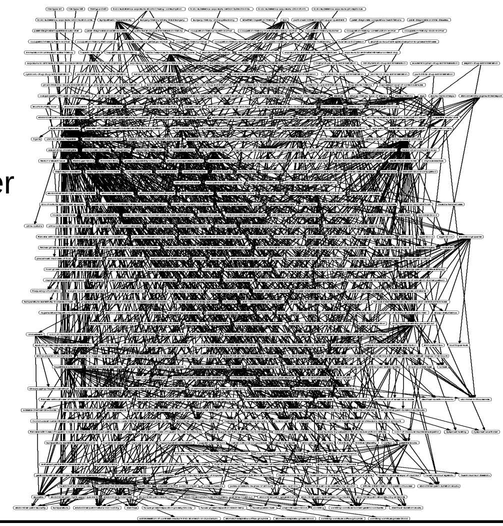
Online Troubleshooters



Pathfinder

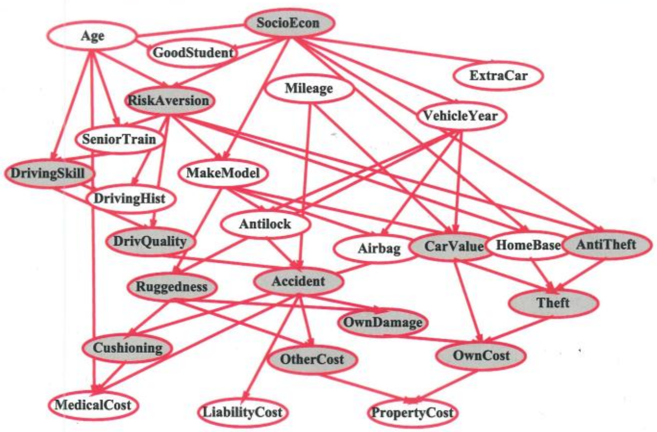
- Pathfinder is one of the first BN systems
- It performs diagnosis of lymph-node diseases
- It deals with over 60 diseases and 100 symptoms and test results
- 14,000 probabilities
- Commercialized by Intellipath and Chapman Hall and applied to about 20 tissue types

Pathfinder Bayes Net



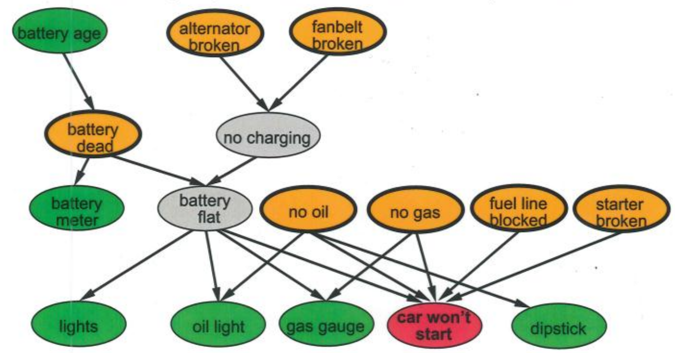
448 nodes, 906 arcs

Example: Car insurance



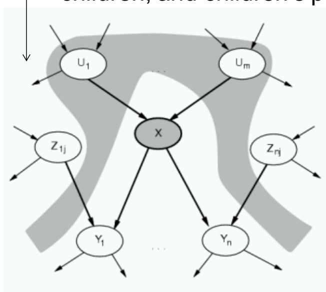
Example: Car diagnosis

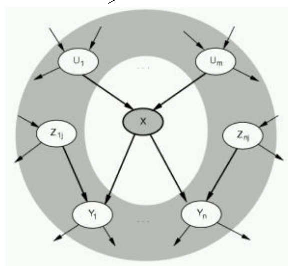
Initial evidence: car won't start
Testable variables (green), "broken, so fix it" variables (orange)
Hidden variables (gray) ensure sparse structure, reduce parameters



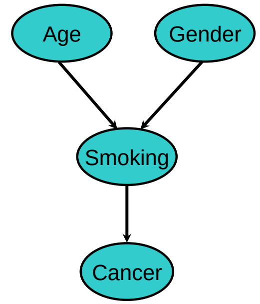
Conditional Independence in Bayes Nets

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



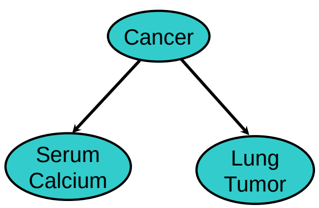


Conditional Independence



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

More Conditional Independence

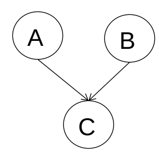


Serum Calcium is conditionally independent of *Lung Tumor*, given *Cancer*

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

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

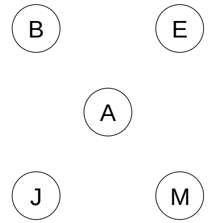
Example with 5 Variables

- 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

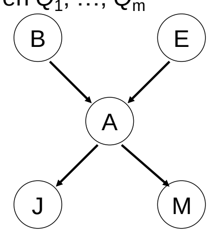
- **Step 1**: add variables. Choose the variables you want to include in the Bayes Net



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 *descendent* of X is conditionally independent of X given $Q_1, \ldots, Q_m$

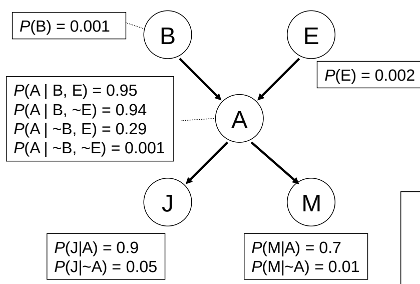


B: there's burglary in your house
E: there's an earthquake
A: your alarm sounds
J: your neighbor John calls you
M: your other neighbor Mary calls you

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|~A). 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 var is considered after all vars that directly influence it

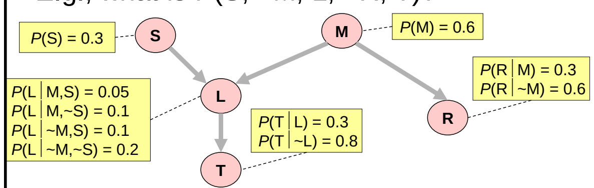
Compactness of Bayes Nets

- A Bayesian Network is a graph structure for representing conditional independence relations in a compact way
- A Bayes net encodes a 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 $O(Nk^M)$
    - grows linearly with N
    - can't have too many parents, though

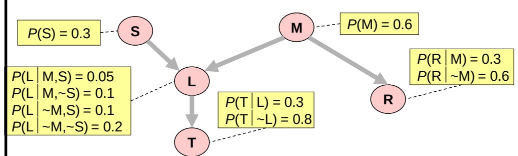
Computing a Joint Entry from a Bayes Net

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!

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)

The General Case

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

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

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

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

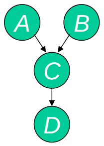
$P(X_{n-2}{=}x_{n-2}, \ldots, X_2{=}x_2, X_1{=}x_1) =$

:

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

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

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 summing 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 with the number of nodes, *N*
- With this joint distribution, we can compute *any* conditional probability, *P*(Q | E), thus we can perform any inference
- How?

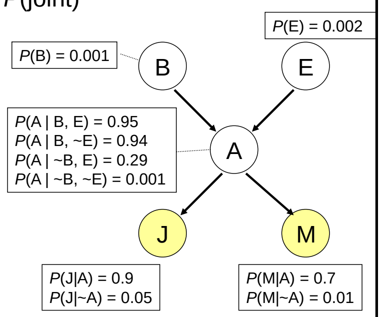
Inference by Enumeration

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

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)



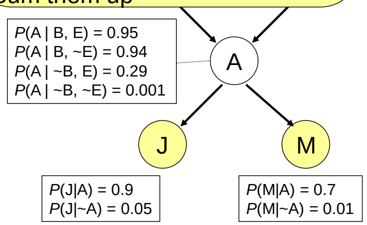
Inference b…

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

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)

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



Inference by…

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

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,

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

Inference by Enumeration

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

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)

Sum up 4 joints

Sum up 8 joints

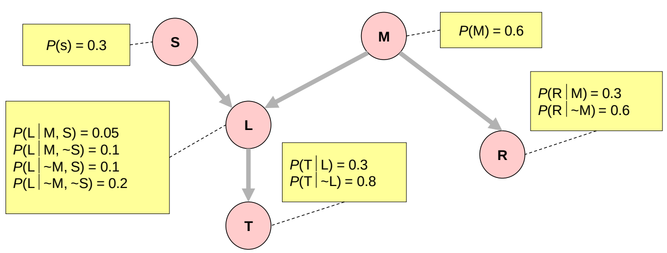
In general, if there are N variables, while evidence contains j variables, how many joints to sum up?

P(J|A) = 0.9
P(J|~A) = 0.05

P(M|A) = 0.7
P(M|~A) = 0.01

Another Example

Compute P(R | T, ~S) from the following Bayes Net



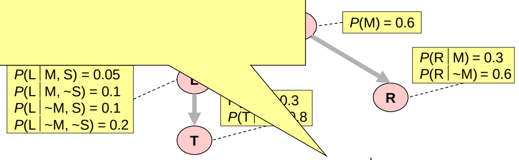
Another Example

Step 1: Compute P(R, T, ~S)

Step 2: Compute P(T, ~S)

Step 3: Return

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



Compute P(R | T, ~S)?

Another Example

Step 1: Compute P(R, T, ~S)

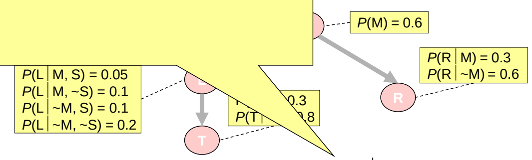
Sum of all the rows in the Joint that match R ^ T ^ ~S

Step 2: Compute P(T, ~S)

Sum of all the rows in the Joint that match T ^ ~S

Step 3: Return

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



Compute P(R | T, ~S)?

Another Example

Step 1: Compute P(R, T, ~S)

Sum of all the rows in the Joint that match R ^ T ^ ~S

4 joint computes

Step 2: Compute P(~R, T, ~S)

Sum of all the rows in the Joint that match ~R ^ T ^ ~S

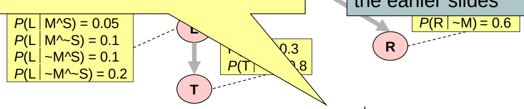
4 joint computes

Step 3: Return

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

= P(T, ~S)

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



Compute P(R | T, ~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)

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

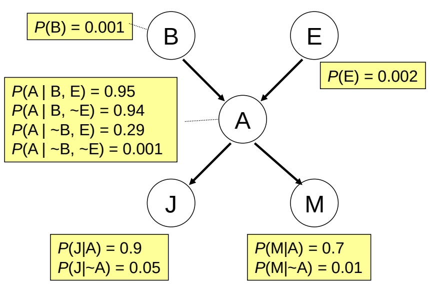
"**Inference by Enumeration**" Algorithm

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**)
- A more complex algorithm called join tree (junction tree) can save even more computation
- But, even so, **exact inference with an arbitrary Bayes Net is NP-Complete**

Parameter (CPT) Learning for BN

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

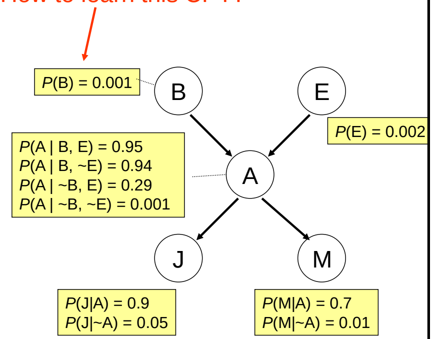


Parameter (CPT) Learning for BN

- 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)
…

How to learn this CPT?

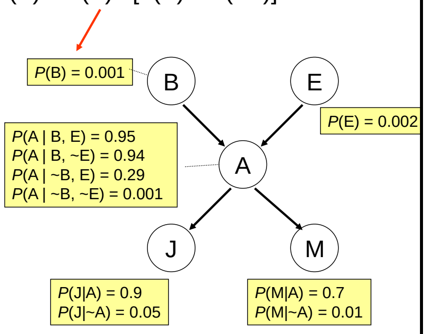


Parameter (CPT) Learning for BN

- 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(B) = #(B) / [#(B) + #(~B)]

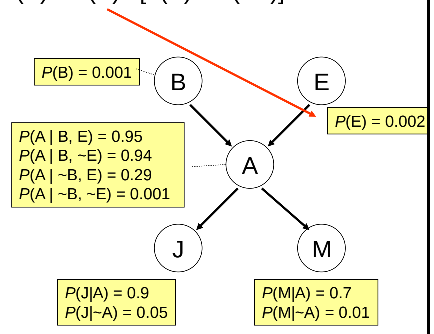


Parameter (CPT) Learning for BN

- 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(E) = #(E) / [#(E) + #(~E)]

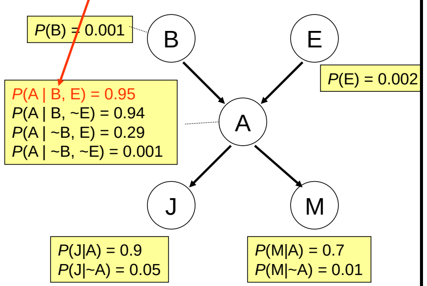


Parameter (CPT) Learning for BN

- 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(A|B,E) = #(A) / [#(A) + #(~A)]

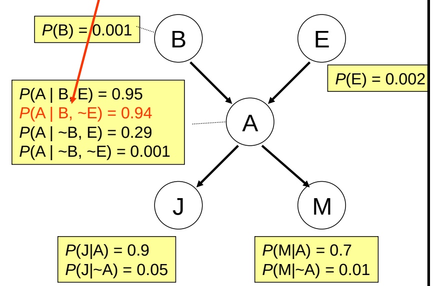


Parameter (CPT) Learning for BN

- 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=true and E=false.
P(A|B,~E) = #(A) / [#(A) + #(~A)]

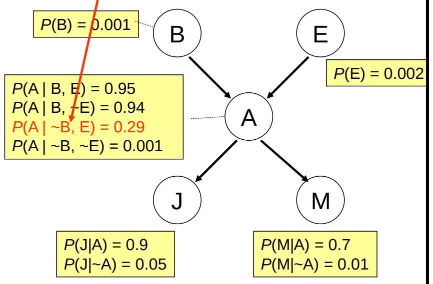


Parameter (CPT) Learning for BN

- 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=true.
P(A|~B,E) = #(A) / [#(A) + #(~A)]

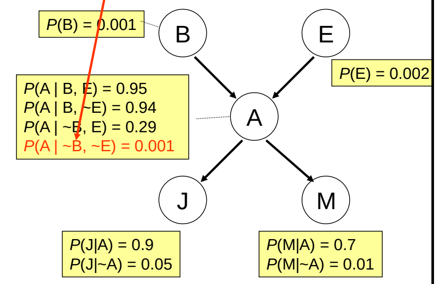


Parameter (CPT) Learning for BN

- 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 'unseen event' problem when dataset is small
  - flip a coin 3 times, all heads → one-sided coin?
- Simplest solution: **'Add one' smoothing**

Smoothing CPT

- '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 bravely saves you when you don't have enough data, and humbly hides away when you do
- It's a form of Maximum *a posteriori* (MAP) estimation

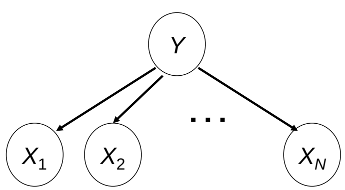
Naïve Bayes Classifier

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

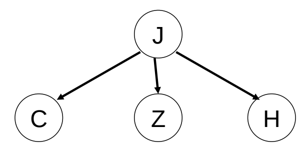
  Class variable — Evidence variable
- Assumes all evidence variables are conditionally independent of each other given the class variable
- Robust since it gives the right answer as long as the correct class is more likely than all others

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



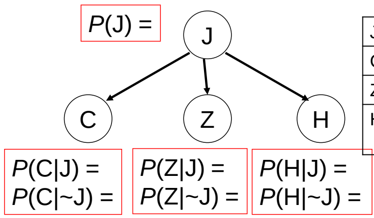
A Special BN: Naïve Bayes Classifiers



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

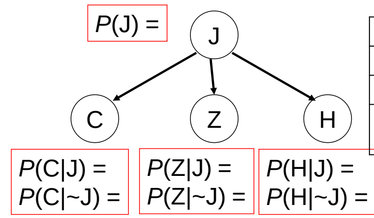
- What's stored in the CPTs?

A Special BN: Naïve Bayes Classifiers



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

- A new person shows up in class wearing an “I live in Union South where I saw the Hunger Games every night” overcoat.
- 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.

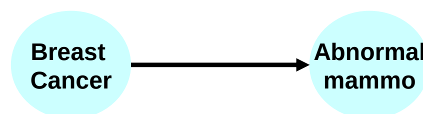
$= P(J,C,Z,H) / [P(J,C,Z,H) + P(\sim J,C,Z,H)]$ by marginalization

where

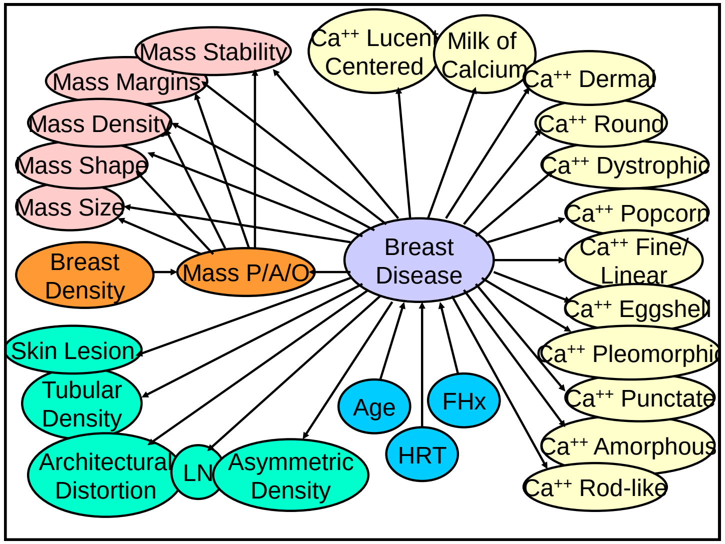
$P(J,C,Z,H) = P(J)P(C|J)P(Z|J)P(H|J)$ by chain rule and conditional independence associated with B.N.

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

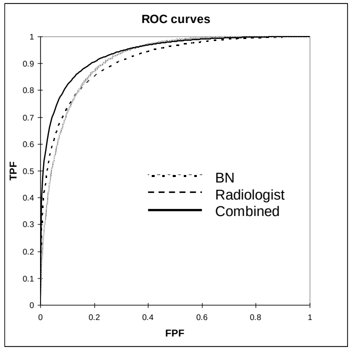
Application: Bayesian Networks for Breast Cancer Diagnosis



Elizabeth S. Burnside
Department of Radiology
University of Wisconsin Hospitals



Results



Radiologist
.916

Bayes Net
.919

Combined
.948

What You Should Know

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