

Bayes Networks

Xiaojin Zhu

`jerryzhu@cs.wisc.edu`

**Computer Sciences Department
University of Wisconsin, Madison**

Outline

- Joint probability is great for inference in a uncertain world, but is terrible to obtain and store
- Bayes net allows us to build joint distributions in manageable chunks
 - Independence, conditional independence
- Bayes net can do any inference
 - but naïve algorithms can be terribly inefficient
 - Some inference algorithms can be more efficient
- Parameter learning in Bayes nets

Joint distribution

- Making a joint distribution of N variables:
 1. List all combinations of values (if each var has k values, k^N combinations)
 2. Assign each combination a probability number
 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 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. $P(\text{Sunny or Hot}) = (150+50+40+5)/365$

Convince yourself this is the same as $P(\text{sunny}) + P(\text{hot}) - P(\text{sunny and hot})$

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

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

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

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

- Joint distribution can take up huge space
- For N variables, each taking k values, the joint distribution has k^N numbers
- It would be good to use fewer numbers...

Using fewer numbers

- Suppose there are two events:
 - B: there's burglary in your house
 - E: there's an earthquake
- The joint distribution of them has 4 entries
- Do we have to invent these 4 numbers, for each combination $P(B, E)$, $P(B, \sim E)$, $P(\sim B, E)$, $P(\sim B, \sim E)$?
 - Can we 'derive' them using $P(B)$ and $P(E)$?
 - What assumption do we need?

Independence

- “Whether there’s a burglary doesn’t depend on whether there’s an earthquake.”

- This is specified as

$$P(B \mid E) = P(B)$$

- Very strong statement! Equivalently

$$P(E \mid B) = P(E)$$

$$P(B, E) = P(B) P(E)$$

- It required domain knowledge other than probability. It needed an understanding of **causation**

Independence

- With independence, we have (can you prove them?)

$$P(B, \sim E) = P(B)P(\sim E),$$

$$P(\sim B, E) = P(\sim B)P(E),$$

$$P(\sim B, \sim E) = P(\sim B)P(\sim E)$$

- Say $P(B)=0.001$, $P(E)=0.002$, $P(B|E)=P(B)$, the joint probability table is:

Burglary	Earthquake	Prob
B	E	
B	$\sim E$	
$\sim B$	E	
$\sim B$	$\sim E$	

- Now we can do anything, since we have the joint.

A more interesting example

- B: there's burglary in your house
- E: there's an earthquake
- A: your alarm sounds
- Your alarm is supposed to sound when there's a burglary. But it sometimes doesn't. And it can occasionally be triggered by an earthquake

A more interesting example

- B: there's burglary in your house
- E: there's an earthquake
- A: your alarm sounds
- Your alarm is supposed to sound when there's a burglary. But it sometimes doesn't. And it can occasionally be triggered by an earthquake
- The knowledge we have so far:
 - $P(B)=0.001$, $P(E)=0.002$, $P(B|E)=P(B)$
 - Alarm is NOT independent of whether there's burglary, nor is it independent of earthquake
- We already know the joint of B, E. All we need is
$$P(A \mid \text{burglary} = b, \text{earthquake} = e)$$
for the 4 cases of $b=B, \sim B, e=E, \sim E$, to get the full joint

A more interesting example

- B: there's burglary in your house
- E: there's an earthquake
- A: your alarm sounds
- Your alarm is supposed to sound when there's a burglary. But it sometimes doesn't. And it can occasionally be triggered by an earthquake

$P(B)=0.001$	$P(A \mid B, E)=0.95$
$P(E)=0.002$	$P(A \mid B, \sim E)=0.94$
$P(B E)=P(B)$	$P(A \mid \sim B, E)=0.29$
	$P(A \mid \sim B, \sim E)=0.001$

- These 6 numbers specify the joint, instead of 7
- Savings are larger with more variables!

A more interesting example

- B: there's burglary in your house
- E: there's an earthquake
- A: your alarm sounds

- Your alarm is supposed to be triggered by either a burglary or an earthquake

Quiz: can you express $P(\sim B, E, \sim A)$ in terms of these probabilities?

$P(B)=0.001$	$P(A B, E)=0.95$
$P(E)=0.002$	$P(A B, \sim E)=0.94$
$P(B E)=P(B)$	$P(A \sim B, E)=0.29$
	$P(A \sim B, \sim E)=0.001$

- These 6 numbers specify the joint, instead of 7
- Savings are larger with more variables!

Introducing Bayes Net

$P(B)=0.001$	$P(A B, E)=0.95$
$P(E)=0.002$	$P(A B, \sim E)=0.94$
$P(B E)=P(B)$	$P(A \sim B, E)=0.29$
	$P(A \sim B, \sim E)=0.001$

DAG, often direct causation,
but don't have to be!

$$P(B)=0.001$$

B

$$P(E)=0.002$$

E

One node per
random variable

$P(A B, E)=0.95$
$P(A B, \sim E)=0.94$
$P(A \sim B, E)=0.29$
$P(A \sim B, \sim E)=0.001$

A

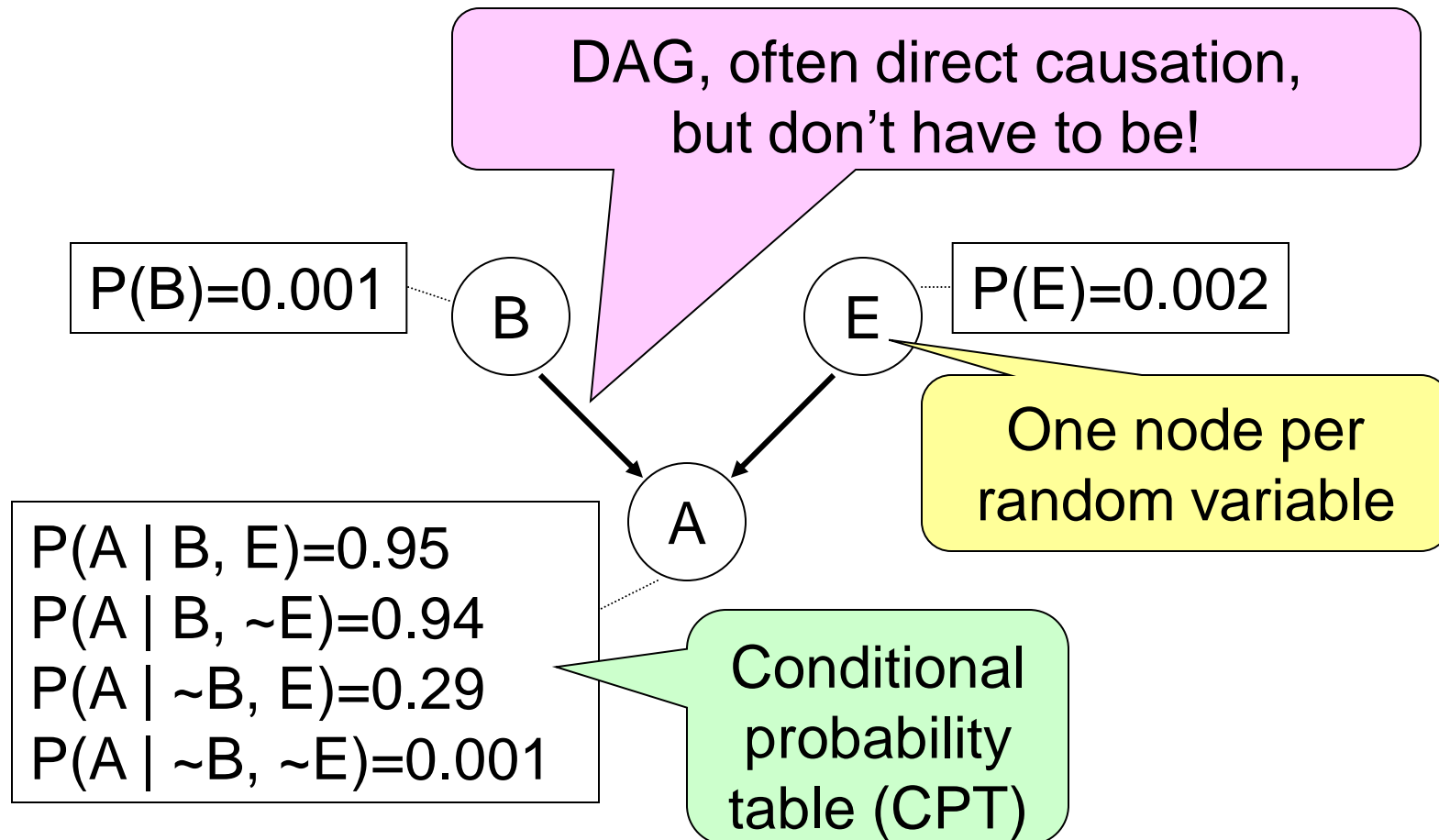
Conditional
probability
table (CPT)

Bayes Net
= Bayesian Network
= Belief Network

Join probability with Bayes Net

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

- Example: $P(\sim B, E, \sim A) = P(\sim B) P(E) P(\sim A \mid \sim B, E)$



Our B.N. has this independence assumption

with Bayes Net

- Example. $P(\sim B, E, \sim A) = P(\sim B) P(E) P(\sim A | \sim B, E)$
- Recall the chain rule:

$$P(\sim B, E, \sim A) = P(\sim B) P(E | \sim B) P(\sim A | \sim B, E)$$

DAG, often direct causation, but don't have to be!

$$P(B)=0.001$$

B

$$P(E)=0.002$$

E

One node per random variable

A

$$\begin{aligned} P(A | B, E) &= 0.95 \\ P(A | B, \sim E) &= 0.94 \\ P(A | \sim B, E) &= 0.29 \\ P(A | \sim B, \sim E) &= 0.001 \end{aligned}$$

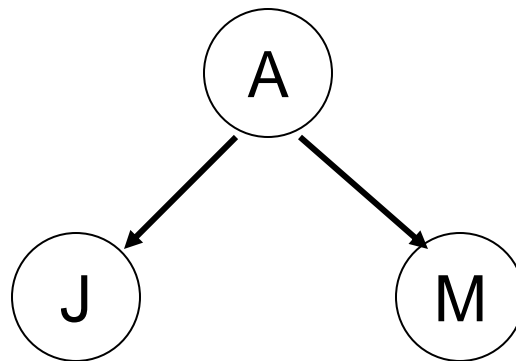
Conditional probability table (CPT)

More to the story...

- 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?

More to the story...

- 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?

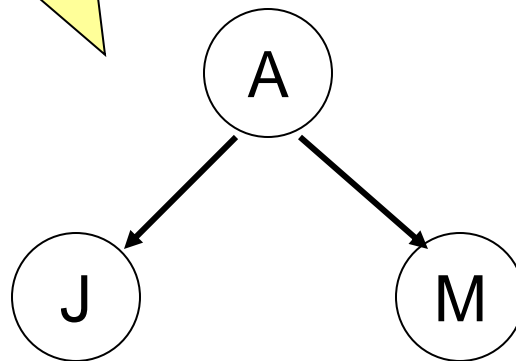


Our BN: $P(A, J, M) = P(A) P(J|A) P(M|A)$
Chain rule: $P(A, J, M) = P(A) P(J|A) P(M|A, J)$

Our B.N. assumes conditional independence
 $P(M|A, J) = P(M|A)$

omised

- 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?

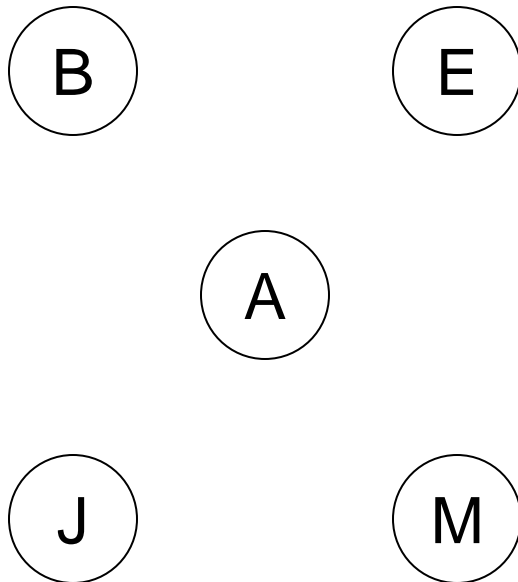


Now with 5 variables

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

Creating a Bayes Net

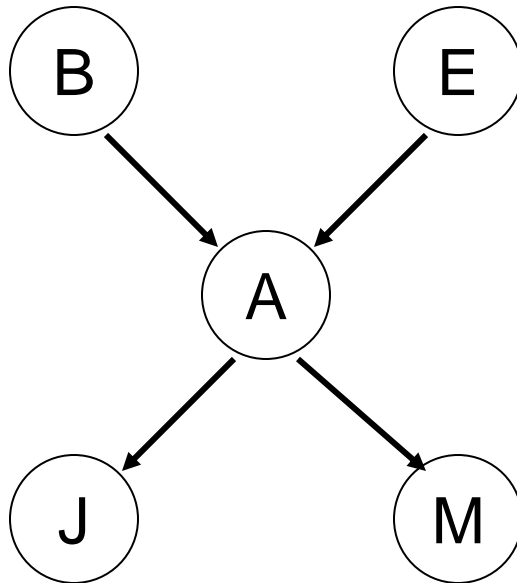
- Step 1: add variables. Choose the variables you want to include in the Bayes Net



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, \dots, Q_m , you are promising that any variable that's not a descendent of X is conditionally independent of X given Q_1, \dots, Q_m

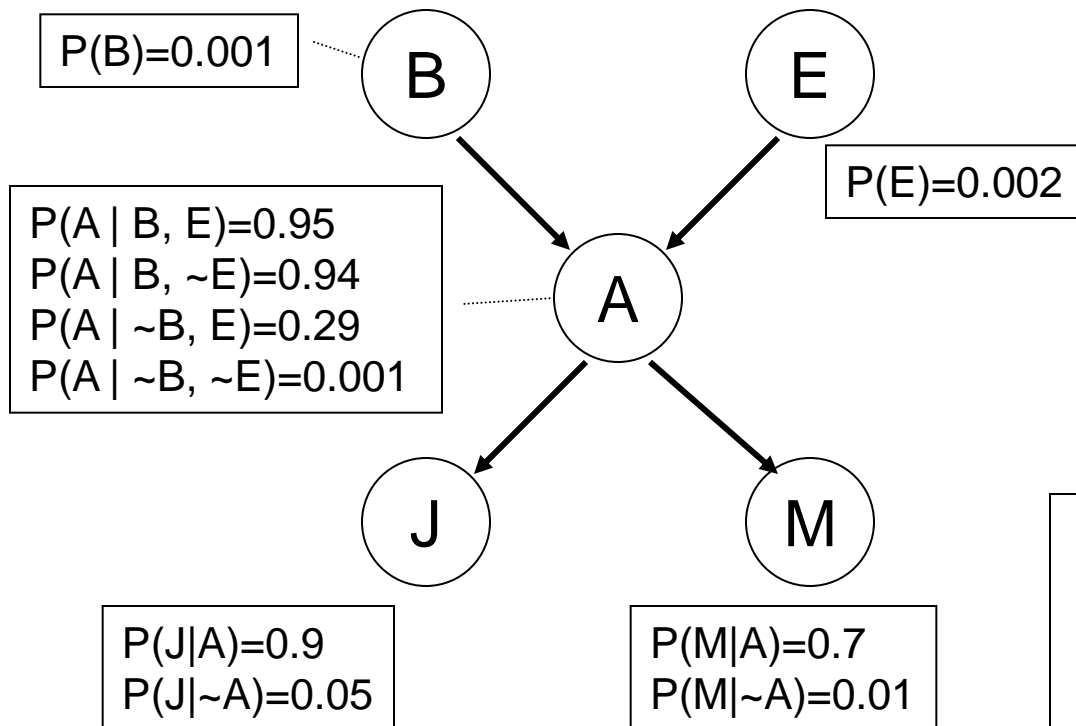


B: there's burglary in your house
E: there's an earthquake
A: your alarm sounds
J: your neighbor John calls you
M: your other neighbor Mary calls you

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** $P(J|\sim A)$. They don't have to sum to 1!



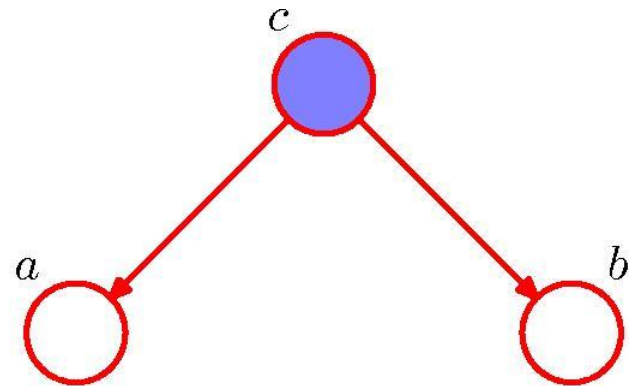
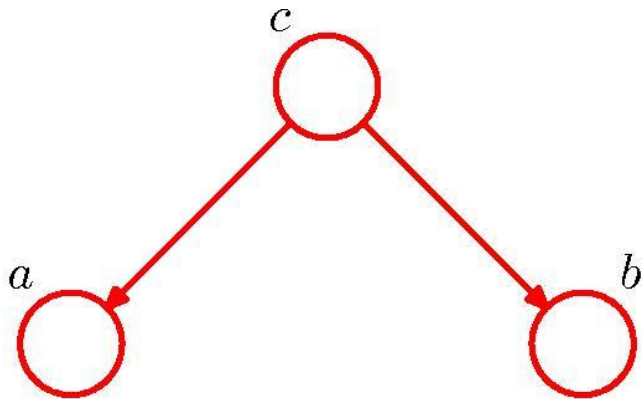
B: there's burglary in your house
E: there's an earthquake
A: your alarm sounds
J: your neighbor John calls you
M: your other neighbor Mary calls you

Creating a Bayes Net

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

Conditional independence

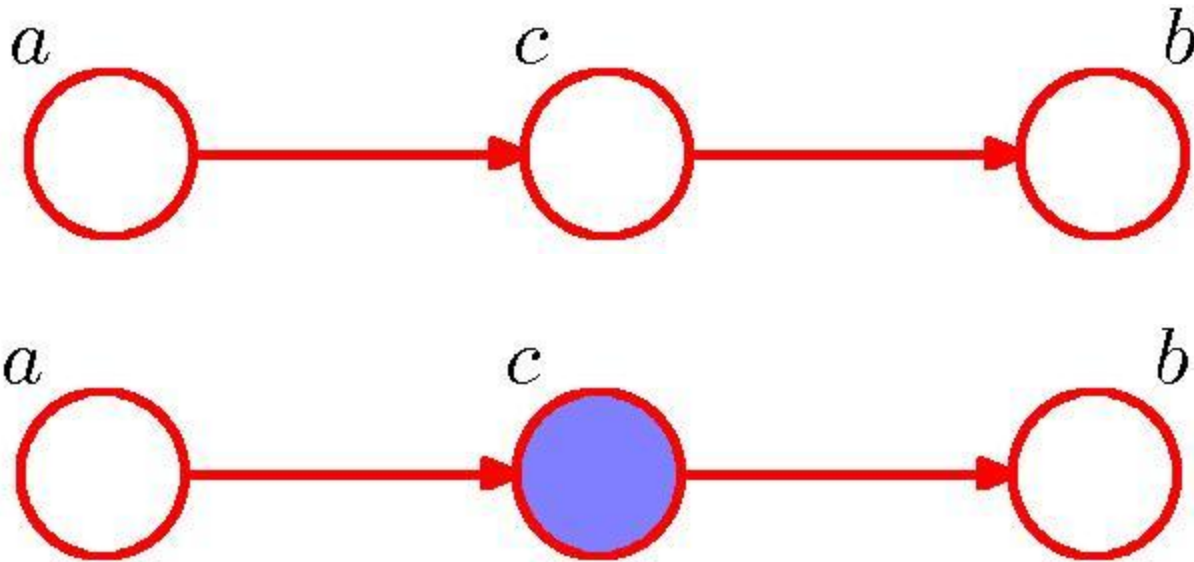
- Case 1: tail-to-tail
- a, b in general not independent
- But a, b conditionally independent given c
- c is a 'tail-to-tail' node, if c observed, it blocks the path



[Examples from Bishop PRML]

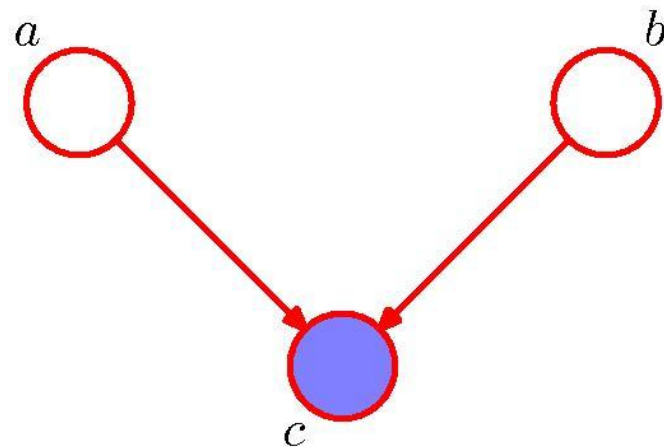
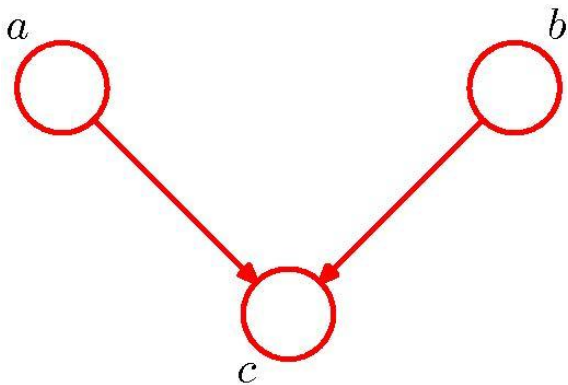
Conditional independence

- Case 2: head-to-tail
- a , b in general not independent
- But a , b conditionally independent given c
- c is a 'head-to-tail' node, if c observed, it blocks the path



Conditional independence

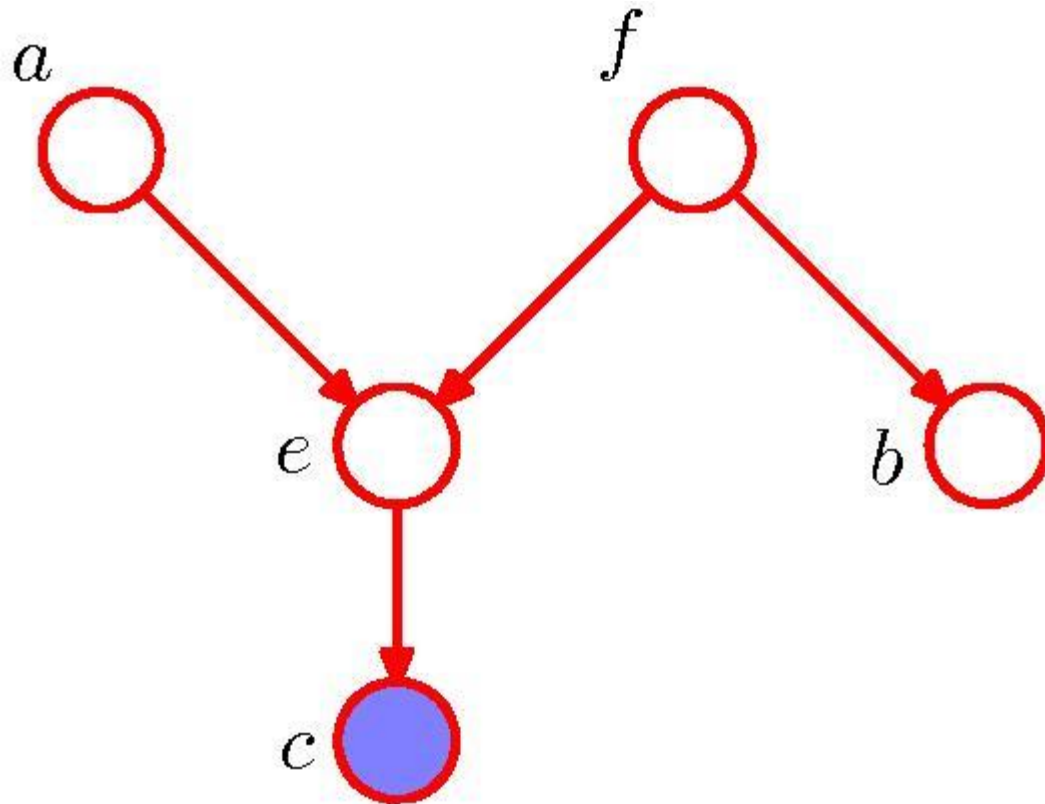
- Case 3: head-to-head
- a, b in general independent
- But a, b NOT conditionally independent given c
- c is a 'head-to-head' node, if c observed, it unblocks the path
 - Important: or if any of c 's descendant is observed



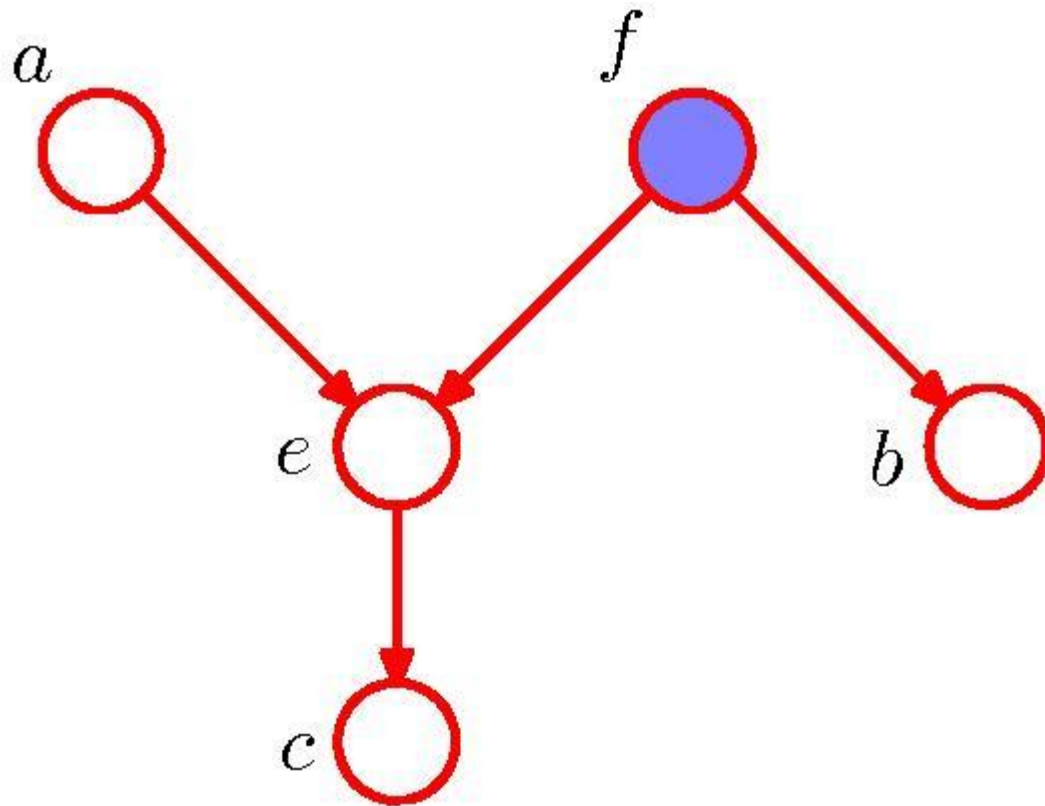
Summary: D-separation

- For any groups of nodes A , B , C : A and B are conditionally independent given C , if
 - all (undirected) paths from any node in A to any node in B are blocked
- A path is blocked if it includes a node such that either
 - The arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in C , or
 - The arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in C .

- The path from a to b not blocked by either e or f
- a, b conditionally dependent given c

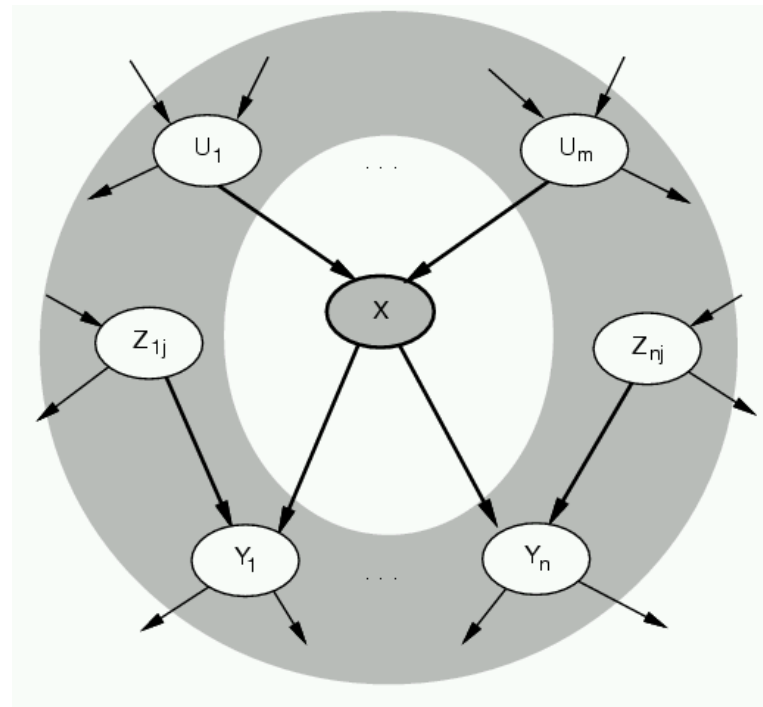
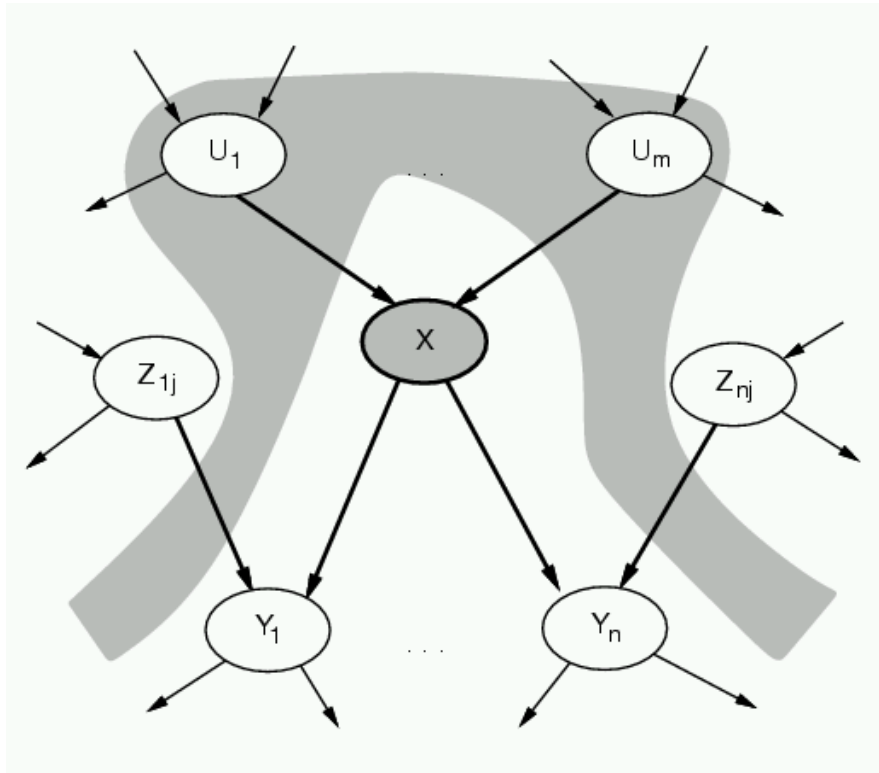


- The path a to b is blocked both at e and at f
- a, b conditionally independent given f



Conditional independence in Bayes Net

- A node is cond. indep. of its non-descendants, given its parents
- A node is cond. indep. of all other nodes, given its Markov blanket (parents, children, spouses)



Compactness of Bayes net

- A Bayes net encodes a joint distribution, often with **far less** parameters
- 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 ($M \ll N$), only needs $O(Nk^M)$
 - grows linearly with N
 - can't have too many parents, though

Where are we now?

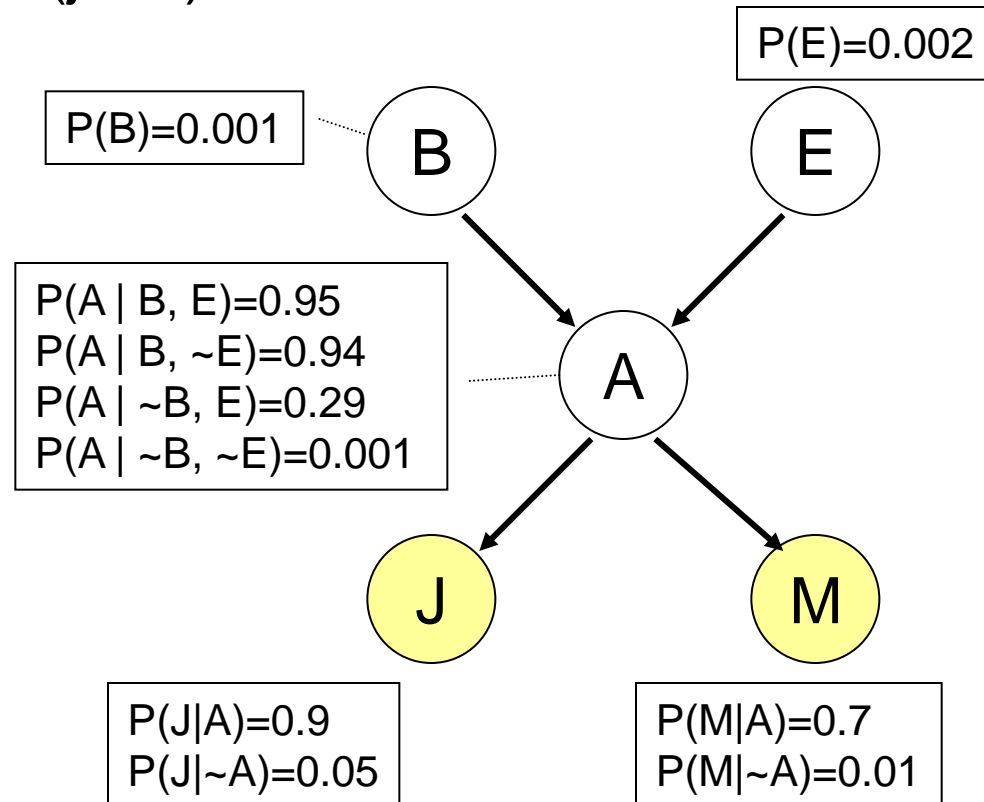
- We defined a Bayes net, using small number of parameters, to describe the joint probability
- Any joint probability can be computed as
$$P(x_1, \dots, x_N) = \prod_i P(x_i \mid \text{parents}(x_i))$$
- The above joint probability can be computed in time linear with the number of nodes N
- With this joint distribution, we can compute any conditional probability $P(Q \mid E)$, thus we can perform any inference
- How?

Inference by enumeration

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

For example $P(B \mid J, \sim M)$

1. Compute $P(B, J, \sim M)$
2. Compute $P(J, \sim M)$
3. Return $P(B, J, \sim M) / P(J, \sim M)$



Inference

$$P(Q|E) = \frac{\sum_{\text{joint matching Q}} P(Q, E)}{\sum_{\text{joint matching E}} P(Q, E)}$$

For example $P(B | J, \sim M)$

1. Compute $P(B, J, \sim M)$
2. Compute $P(J, \sim M)$
3. Return $P(B, J, \sim M) / P(J, \sim M)$

Compute the joint (4 of them)

$$P(B, J, \sim M, A, E)$$

$$P(B, J, \sim M, A, \sim E)$$

$$P(B, J, \sim M, \sim A, E)$$

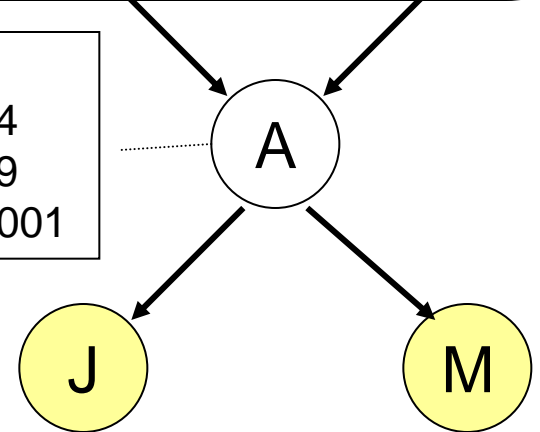
$$P(B, J, \sim M, \sim A, \sim E)$$

Each is $O(N)$ for sparse graph

$$P(x_1, \dots, x_N) = \prod_i P(x_i | \text{parents}(x_i))$$

Sum them up

$$\begin{aligned} P(A | B, E) &= 0.95 \\ P(A | B, \sim E) &= 0.94 \\ P(A | \sim B, E) &= 0.29 \\ P(A | \sim B, \sim E) &= 0.001 \end{aligned}$$



$$\begin{aligned} P(J|A) &= 0.9 \\ P(J|\sim A) &= 0.05 \end{aligned}$$

$$\begin{aligned} P(M|A) &= 0.7 \\ P(M|\sim A) &= 0.01 \end{aligned}$$

Inference by Enumeration

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

For example $P(B \mid J, \sim M)$

1. Compute $P(B, J, \sim M)$
2. Compute $P(J, \sim M)$
3. Return $P(B, J, \sim M) / P(J, \sim M)$

Compute the joint (8 of them)

$P(J, \sim M, B, A, E)$

$P(J, \sim M, B, A, \sim E)$

$P(J, \sim M, B, \sim A, E)$

$P(J, \sim M, B, \sim A, \sim E)$

$P(J, \sim M, \sim B, A, E)$

$P(J, \sim M, \sim B, A, \sim E)$

$P(J, \sim M, \sim B, \sim A, E)$

$P(J, \sim M, \sim B, \sim A, \sim E)$

Each is $O(N)$ for sparse graph

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

Sum them up

Inference by enumeration

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

Sum up 4 joints

Sum up 8 joints

=0.002

For example $P(B | J, \sim M)$

1. Compute $P(B, J, \sim M)$
2. Compute $P(J, \sim M)$
3. Return $P(B, J, \sim M) / P(J, \sim M)$

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

$P(J|A)=0.9$
 $P(J|\sim A)=0.05$

$P(M|A)=0.7$
 $P(M|\sim A)=0.01$

Inference by enumeration

- In general if there are N variables, while evidence contains j variables, and each variable has k values, how many joints to sum up?

Inference by enumeration

- 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
- Some computation can be saved by carefully order the terms and re-use intermediate results (variable elimination)
- A more complex algorithm called join tree (junction tree) can save even more computation

Inference by enumeration

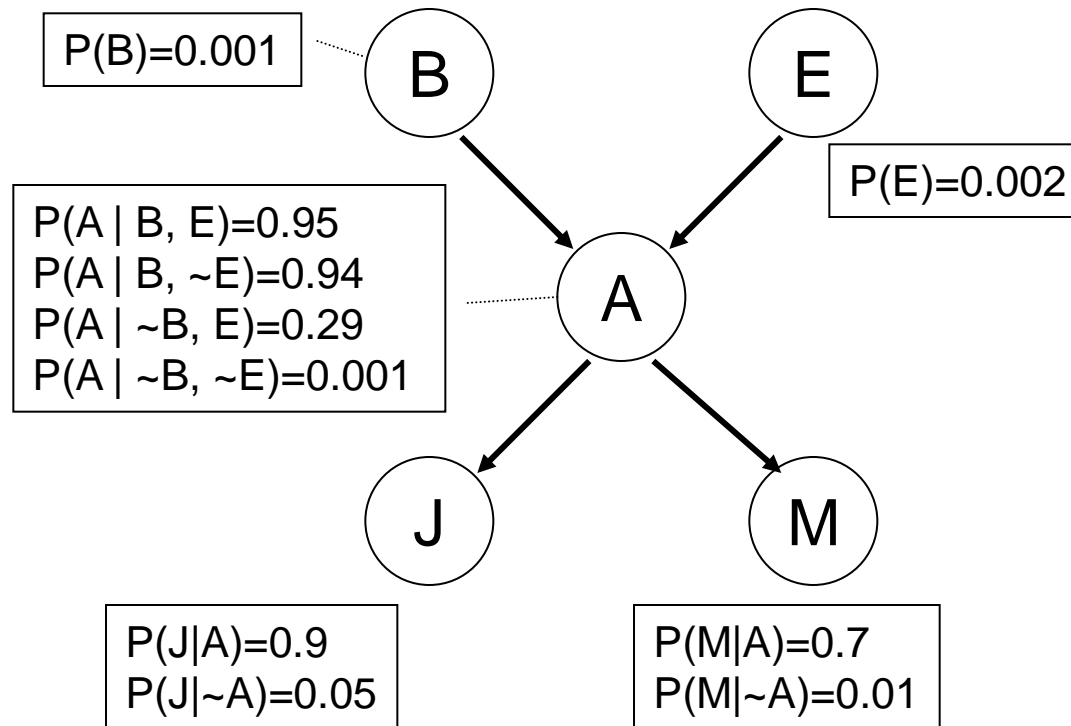
- In general if there are N variables, while evidence contains j variables, and each variable has k values, how many joints to sum up? $k^{(N-j)}$
- It is this summation that makes **enumeration** inefficient
- **The bad news: exact inference with an arbitrary Bayes Net is intractable**
- Some computation can be saved by carefully order the terms and re-use intermediate results (**variable elimination**)
- A more complex algorithm called **join tree** (**junction tree**) can save even more computation

Approximate inference by sampling

- Inference can be done **approximately** by sampling
- General sampling approach:
 - Generate many, many samples (each sample is a complete assignment of all variables)
 - Count the fraction of samples matching query and evidence
 - As the number of samples approaches ∞ , the fraction converges to the posterior
$$P(\text{query} \mid \text{evidence})$$
- We'll see 3 sampling algorithms (there are more...)
 1. Simple sampling
 2. Likelihood weighting
 3. Gibbs sampler

1. Simple sampling

- This BN defines a joint distribution
- Can you generate a set of samples that have the same underlying joint distribution?

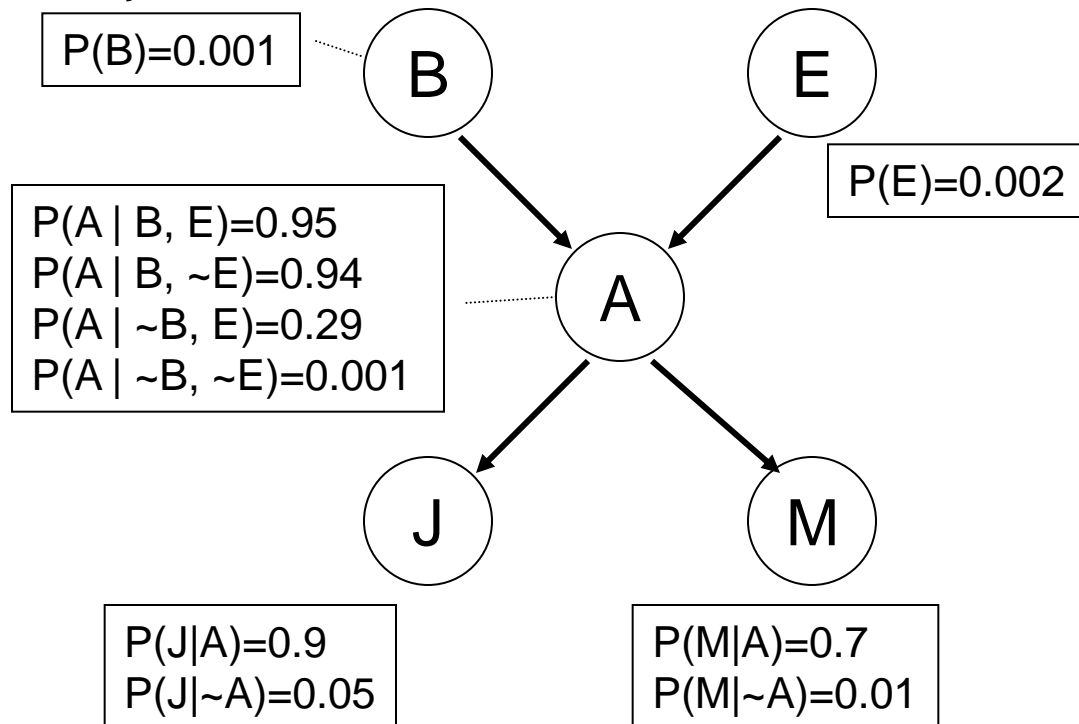


1. Simple sampling

1. Sample B: $x = \text{rand}(0,1)$. If $(x < 0.001)$ $B = \text{true}$ else $B = \text{false}$
2. Sample E: $x = \text{rand}(0,1)$. If $(x < 0.002)$ $E = \text{true}$ else $E = \text{false}$
3. If $(B == \text{true} \text{ and } E == \text{true})$ sample $A \sim \{0.95, 0.05\}$
elseif $(B == \text{true} \text{ and } E == \text{false})$ sample $A \sim \{0.94, 0.06\}$
elseif $(B == \text{false} \text{ and } E == \text{false})$ sample $A \sim \{0.29, 0.71\}$
else sample $A \sim \{0.001, 0.999\}$
4. Similarly sample J
5. Similarly sample M

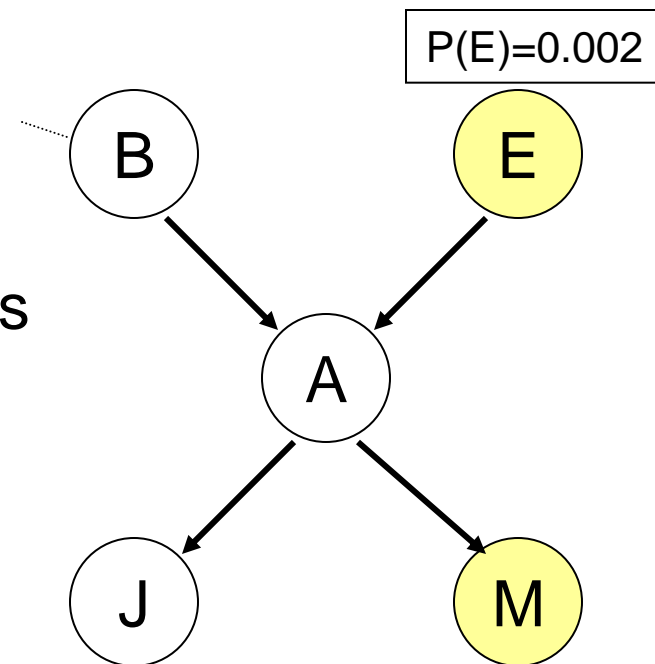
This generates
one sample.

Repeat to generate
more samples



1. Inference with simple sampling

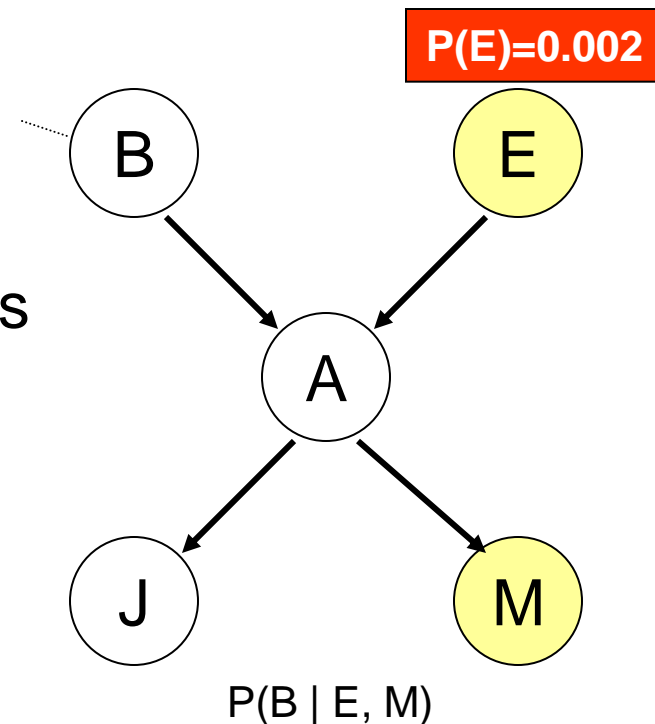
- Say we want to infer B, given E, M, i.e. $P(B \mid E, M)$
- We generate tons of samples
- Keep those samples with $E=\text{true}$ and $M=\text{true}$, **throw away others**
- In the ones we keep (N of them), count the ones with $B=\text{true}$, i.e. those fit our query (N1)
- We return an estimate of
$$P(B \mid E, M) \approx N1 / N$$
- The quality of this estimate improves as we sample more
- You should be able to generalize the method to arbitrary BN



1. Inference with simple sampling

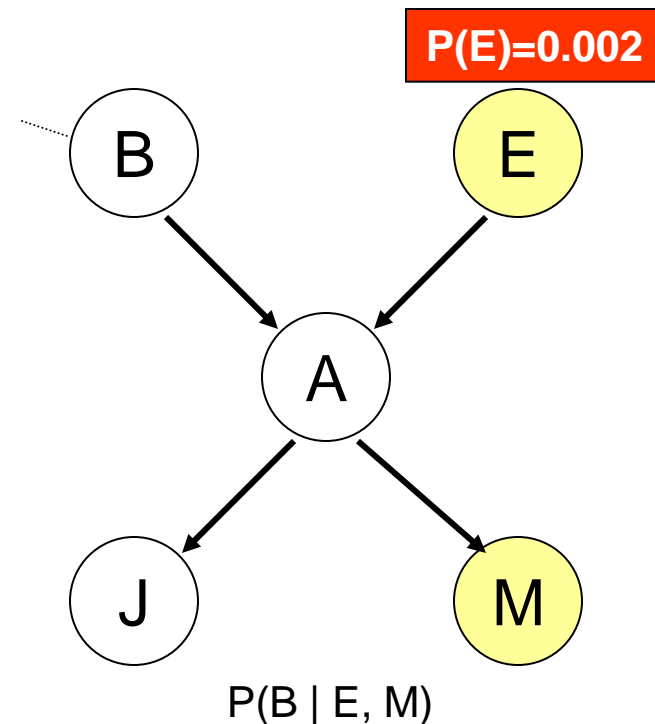
- Say we want to infer B, given E, M, i.e. $P(B \mid E, M)$
- We generate tons of samples
- Keep those samples with $E=\text{true}$ and $M=\text{true}$, **throw away others**
- In the ones we keep (N of them), count the ones with $B=\text{true}$, i.e. those fit our query (N1)
- We return an estimate of $P(B \mid E, M) \approx N1 / N$
- The quality of this estimate improves as we sample more

Can you see a problem with simple sampling?



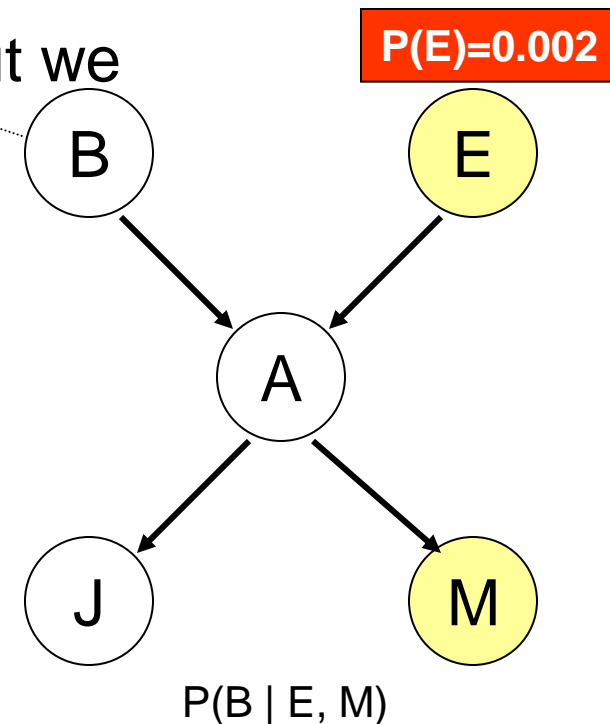
1. Inference with simple sampling

- Since $P(E)=0.002$, we expect only 1 sample with $E=\text{true}$ in every 500 samples
- We'll throw away the 499 samples. Huge waste
- This observation leads to...



2. Likelihood weighting

- Say we've generated B, and we're about to generate E
- E is an evidence node, known to be true
- In simple sampling, we will generate
 - E=true $P(E)=0.2\%$ of the time
 - E=false 99.8% of the time
- Instead we always generate E=true but we weight the sample down by $P(E)=0.002$
- Initially the sample has weight $w=1$, now $w=w*0.002$
- This is 'virtually throwing away'

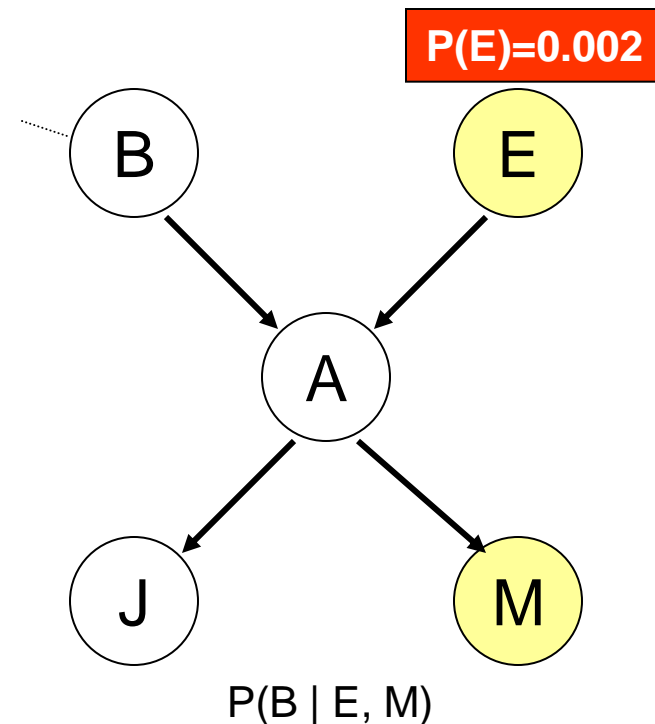


2. Likelihood weighting

- Generate A, J as before
- When it's time to generate evidence M from $P(M|A)$, again always generate $M=\text{true}$, but weight the sample by $w=w \cdot P(M|A)$ [note it depends on A's value]
- If $A=\text{true}$ and $P(M|A)=0.7$, the final weight for this sample is $w=0.002 \cdot 0.7$

- Keep all samples, each with weight w_1, \dots, w_N
- Return estimate

$$P(B|E,M) = \frac{\sum_{B=\text{true}} w_i}{\sum_{\text{all}} w_i}$$



2. Likelihood weighting

- Generate A, J as before
- When it's time to **generate evidence** M from $P(M|A)$, again always generate $M=true$, but weight the sample by $w = P(M=false|A=true)$

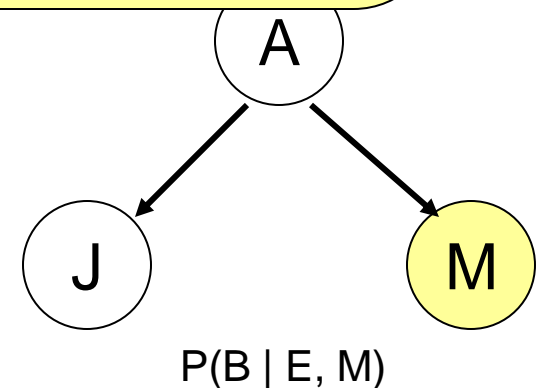
- If A=same, we apply this trick whenever we generate the value for an evidence node.

You should be able to generalize likelihood weighting to general BN.

- Keep weights w_1, \dots, w_n

- Return estimate

$$P(B|E,M) = \frac{\sum_{B=true} w_i}{\sum_{all} w_i}$$



3. Gibbs sampler

Gibbs sampler is the simplest method in the family of Markov Chain Monte Carlo (MCMC) methods

1. Start from an arbitrary sample, but fix evidence nodes to their observed values, e.g.

(B=true, E=true, A=false, J=false, M=true)

2. For each hidden node X, fixing all other nodes, resample its value from

$$P(X=x \mid \text{Markov-blanket}(X))$$

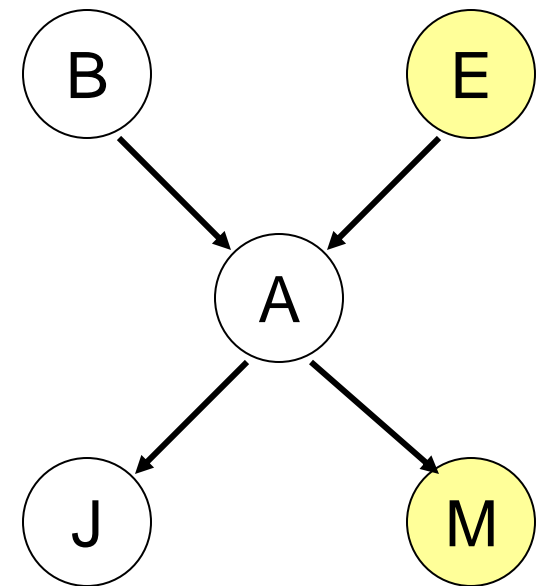
For example, we sample B from

$$P(B \mid E=\text{true}, A=\text{false})$$

Update with its new sampled value, and move on to A, J.

3. We now have a new sample.

Repeat from step 2.



$$P(B \mid E, M)$$

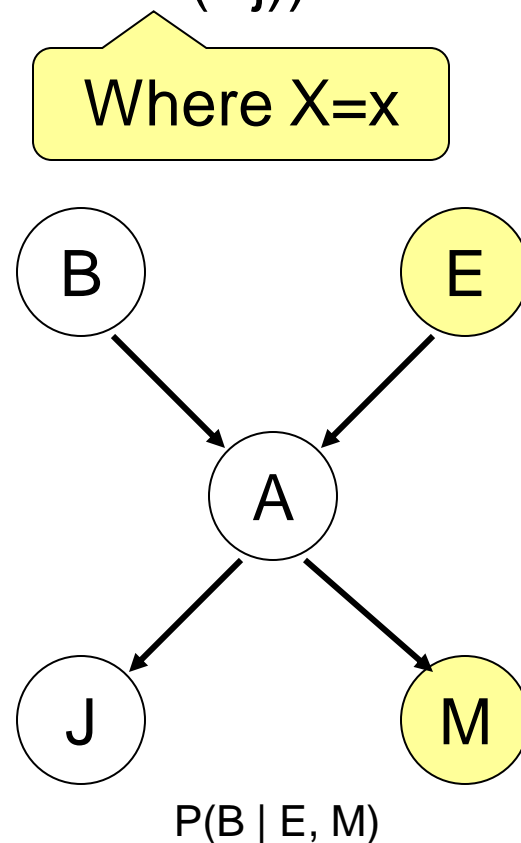
3. Gibbs sampler

- Keep all samples. $P(B \mid E, M)$ is the fraction with $B=\text{true}$

- In general, $P(X=x \mid \text{Markov-blanket}(X)) \propto P(X=x \mid \text{parents}(X)) * \prod_{Y_j \in \text{children}(X)} P(y_j \mid \text{parents}(Y_j))$

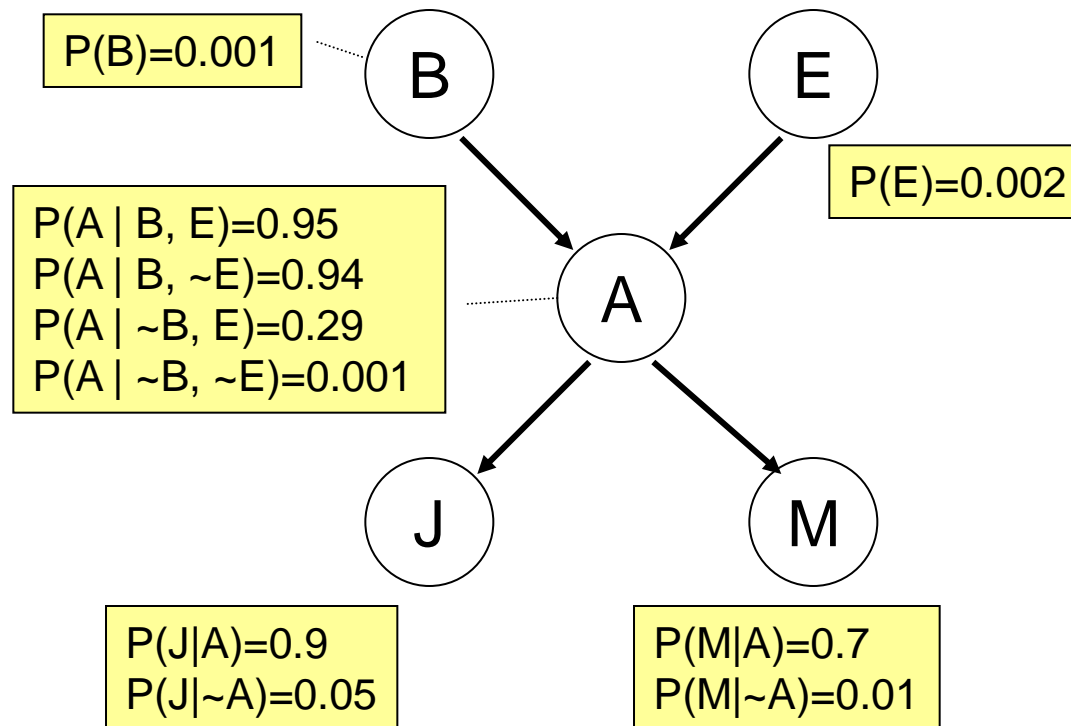
Compute the above for $X=x_1, \dots, x_k$, then normalize

- 'burn-in': do not use the first N_b samples (e.g. $N_b=1000$)



Parameter (CPT) learning for BN

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

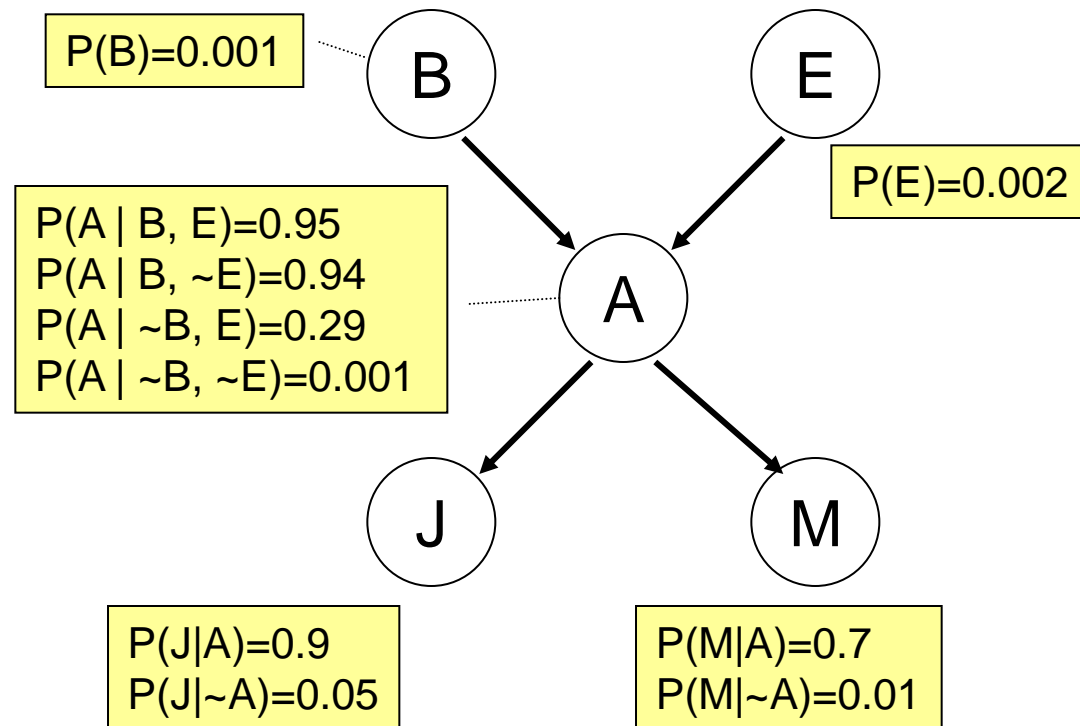


Parameter (CPT) learning for BN

- Learn from a data set like this:

($\sim B$, $\sim E$, $\sim A$, J , $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, J , $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
(B , $\sim E$, A , J , M)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, M)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
(B , E , A , $\sim J$, M)
($\sim B$, $\sim E$, $\sim A$, $\sim J$, $\sim M$)
...

How to learn this CPT?



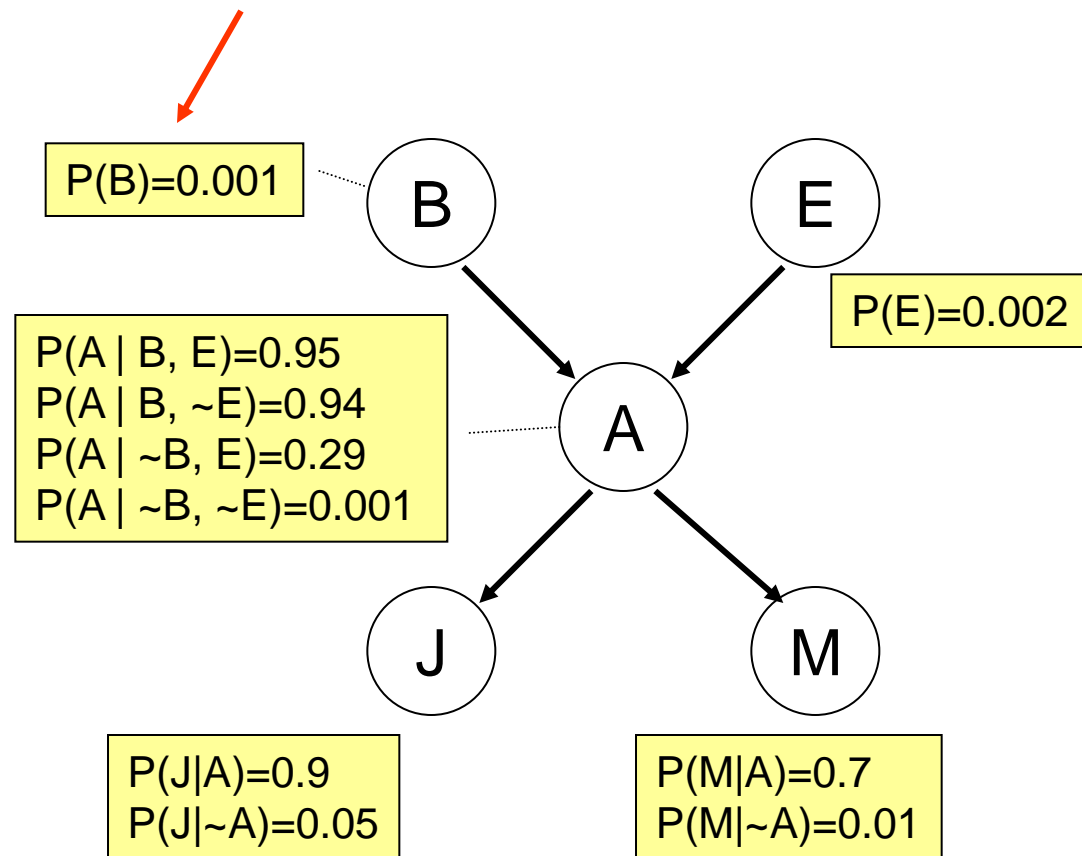
Parameter (CPT) learning for BN

- 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)
(B, E, A, ~J, M)
(~B, ~E, ~A, ~J, ~M)
...

Count $\#(B)$ and $\#(\sim B)$ in dataset.

$$P(B) = \#(B) / [\#(B) + \#(\sim B)]$$



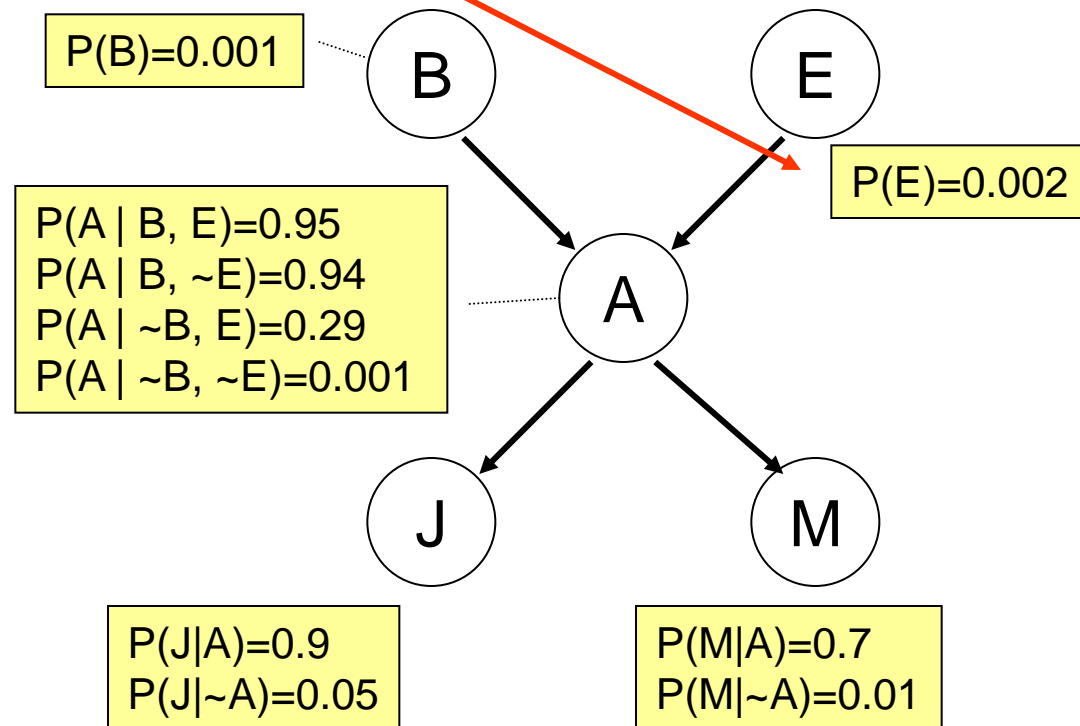
Parameter (CPT) learning for BN

- Learn from a data set like this:

(~E, ~E, ~A, J, ~M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, J, ~M)
(~E, ~E, ~A, ~J, ~M)
(B, ~E, A, J, M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, ~J, M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, J, ~M)
(~E, E, A, J, M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, ~J, M)
(~E, ~E, ~A, ~J, ~M)
(~E, ~E, ~A, ~J, ~M)
(B, E, A, ~J, M)
(~E, ~E, ~A, ~J, ~M)
...

Count $\#(E)$ and $\#(\sim E)$ in dataset.

$$P(E) = \#(E) / [\#(E) + \#(\sim E)]$$



Parameter (CPT) learning for BN

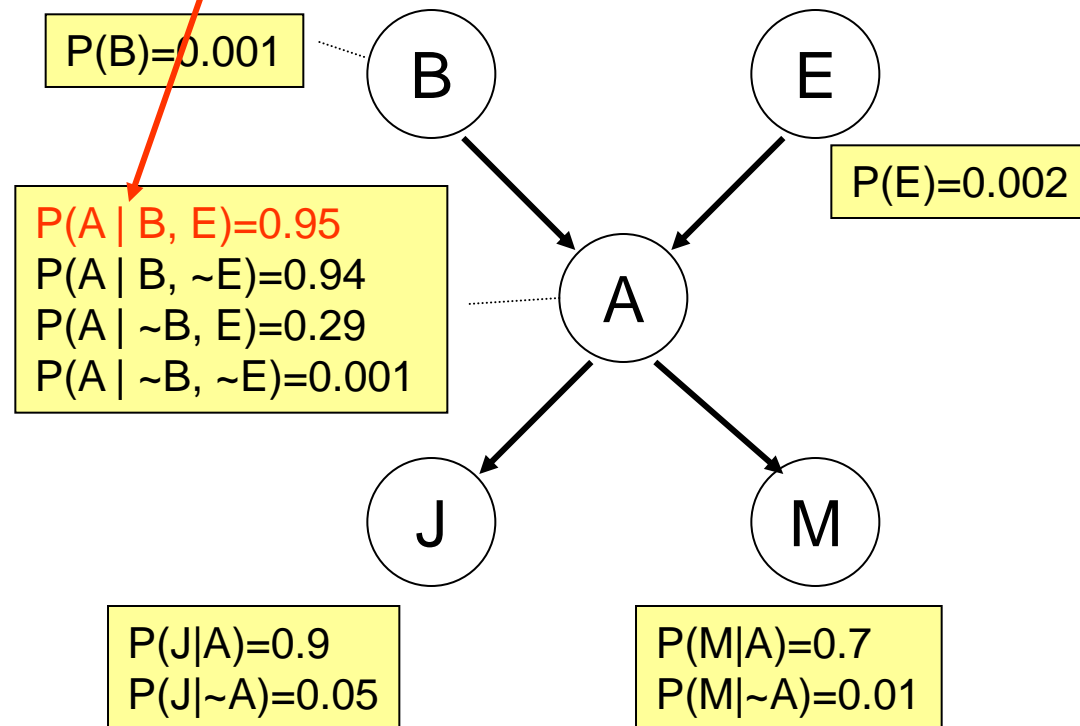
- Learn from a data set like this:

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

...

Count $\#(A)$ and $\#(\sim A)$ in dataset
where **B=true** and **E=true**.

$$P(A|B,E) = \#(A) / [\#(A) + \#(\sim A)]$$



Parameter (CPT) learning for BN

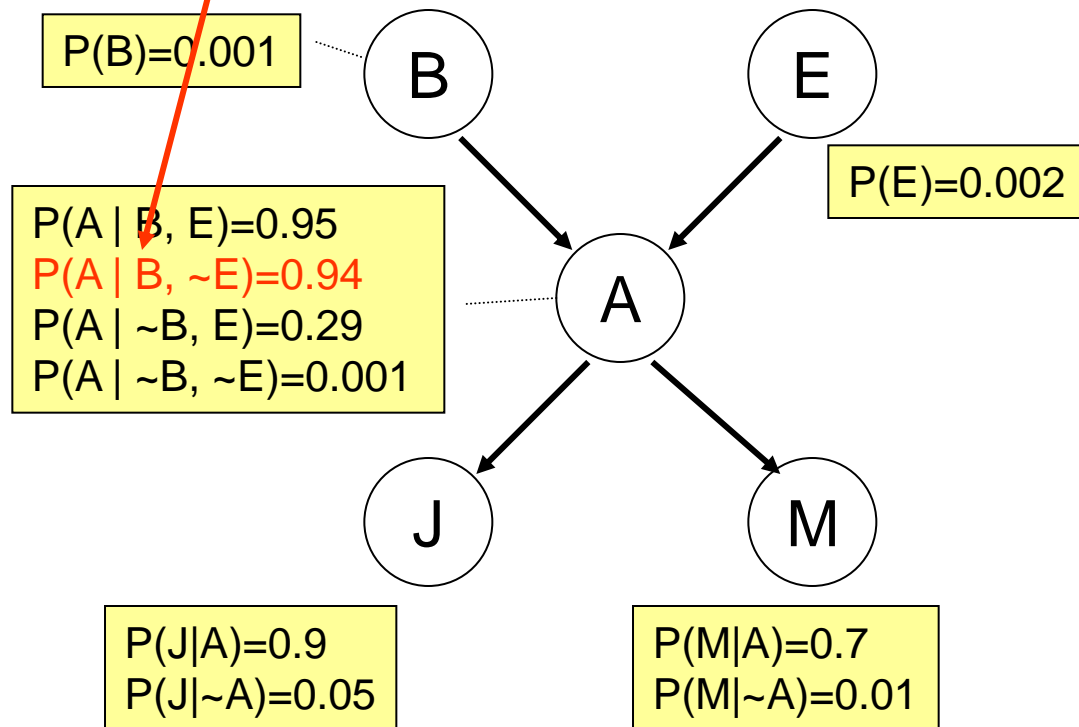
- Learn from a data set like this:

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

...

Count $\#(A)$ and $\#(\sim A)$ in dataset where **B=true** and **E=false**.

$$P(A|B, \sim E) = \#(A) / [\#(A) + \#(\sim A)]$$



Parameter (CPT) learning for BN

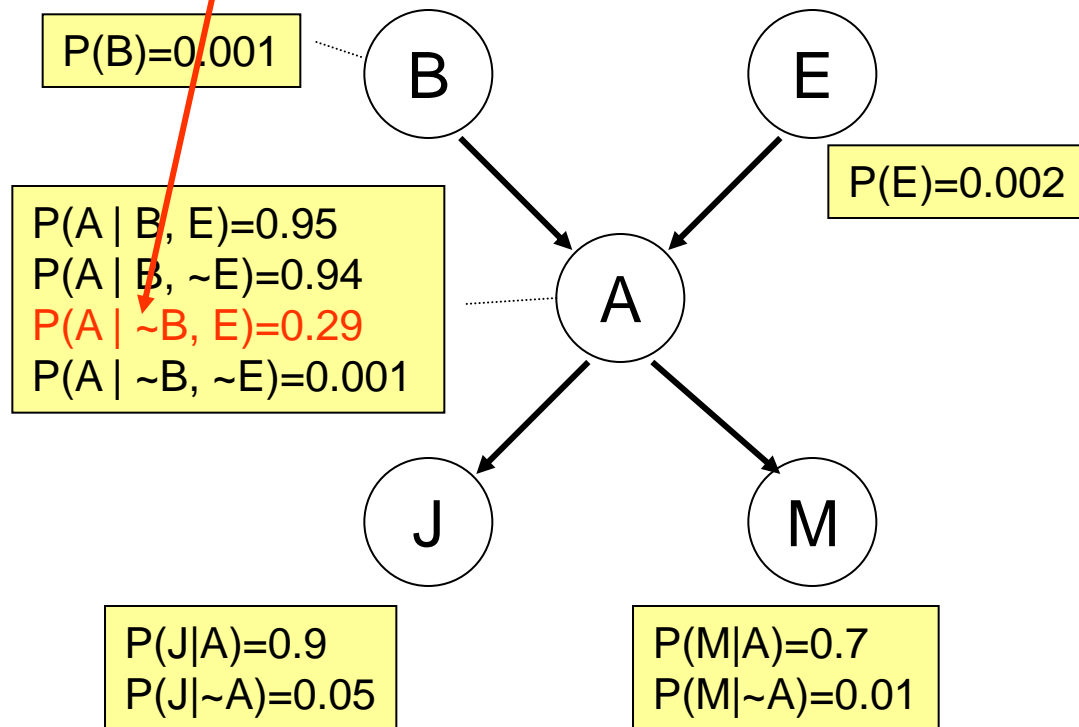
- Learn from a data set like this:

(~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 $\#(\sim A)$ in dataset
where **B=false** and **E=true**.

$$P(A|\sim B, E) = \#(A) / [\#(A) + \#(\sim A)]$$



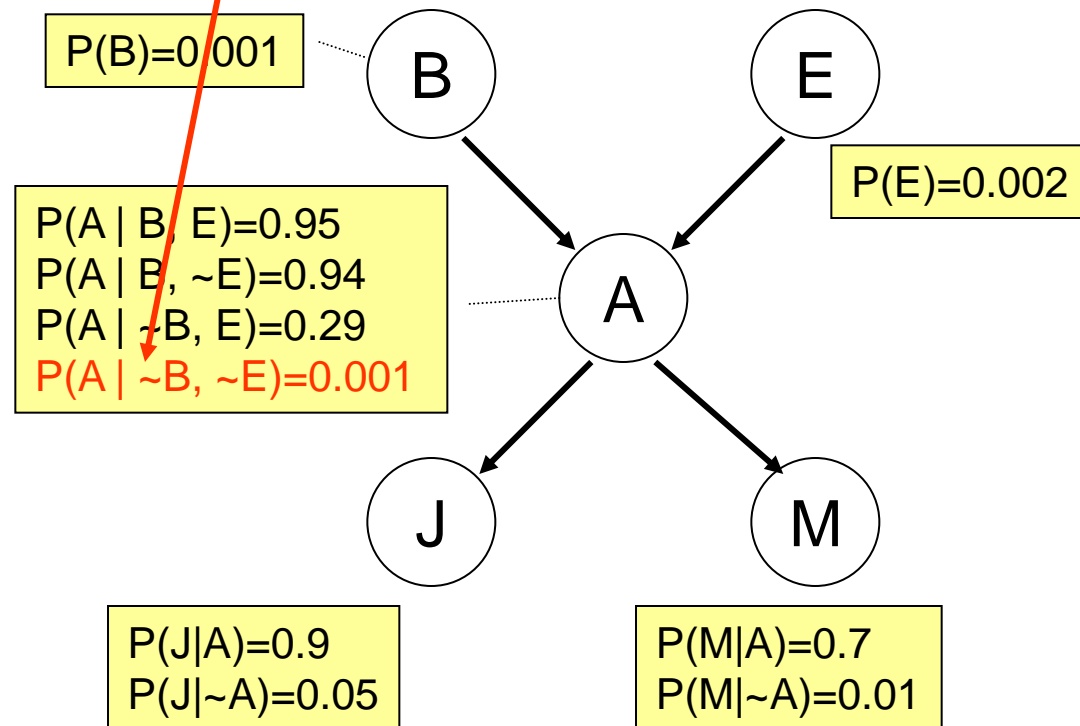
Parameter (CPT) learning for BN

- 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 $\#(\sim A)$ in dataset where **B=false** and **E=false**.

$$P(A|\sim B, \sim E) = \#(A) / [\#(A) + \#(\sim A)]$$



...

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 $\#(\sim A)$ in dataset
where **B=true and E=true**.

$$P(A|B,E) = \#(A) / [\#(A) + \#(\sim A)]$$

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

Do you want to set

$$P(A|B,E)=1$$

$$P(\sim A|B,E)=0?$$

Why or why not?

...

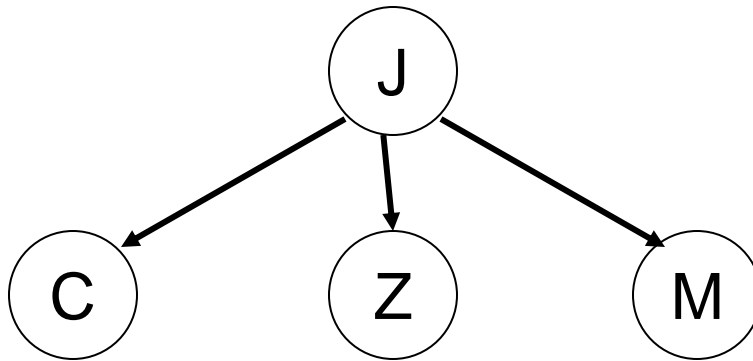
Parameter (CPT) learning for BN

- $P(X=x \mid \text{parents}(X))$ = (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 \rightarrow one-sided coin?
- 'Add one' smoothing: the simplest solution.

Smoothing CPT

- ‘Add one’ smoothing: **add 1 to all counts**
- In the previous example, count $\#(A)$ and $\#(\sim A)$ in dataset where $B=\text{true}$ and $E=\text{true}$
 - $P(A|B,E) = [\#(A)+1] / [\#(A)+1 + \#(\sim A)+1]$
 - If $\#(A)=1$, $\#(\sim A)=0$:
 - without smoothing $P(A|B,E)=1$, $P(\sim A|B,E)=0$
 - with smoothing $P(A|B,E)=0.67$, $P(\sim A|B,E)=0.33$
 - If $\#(A)=100$, $\#(\sim A)=0$:
 - without smoothing $P(A|B,E)=1$, $P(\sim A|B,E)=0$
 - with smoothing $P(A|B,E)=0.99$, $P(\sim A|B,E)=0.01$
- 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) estimate

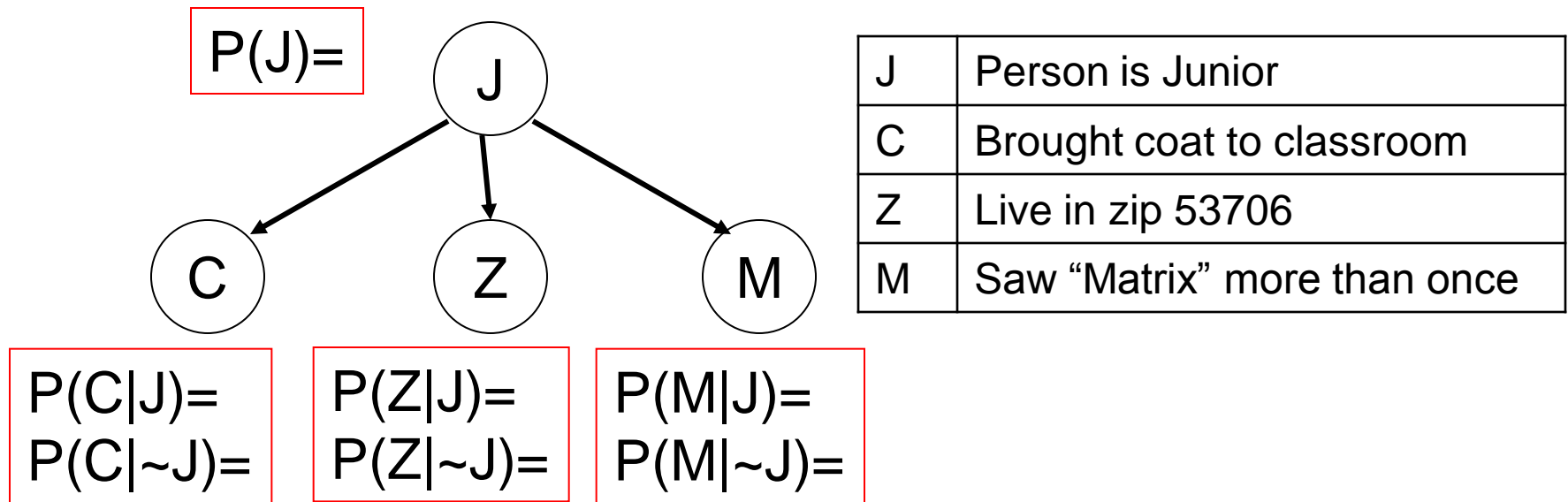
A special BN: Naïve Bayes Classifiers



J	Person is Junior
C	Brought coat to classroom
Z	Live in zip 53706
M	Saw “Matrix” more than once

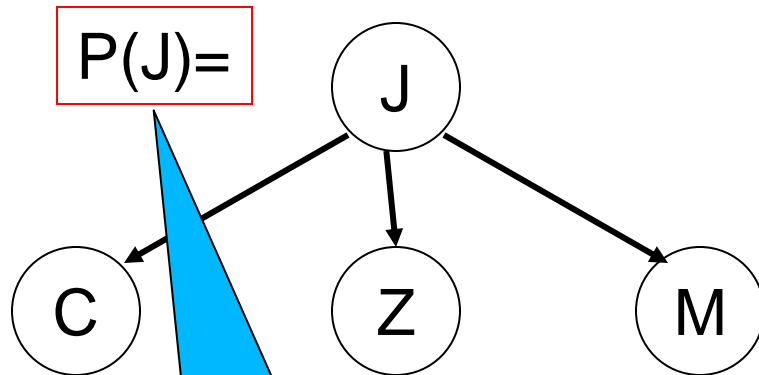
- What's stored in the CPTs?

A special BN: Naïve Bayes Classifiers



- Suppose we have a database of 30 people who attend a lecture. How could we use it to estimate the values in the CPTs?

A special BN: Naïve Bayes Classifiers



J	Person is Junior
C	Brought coat to classroom
Z	Live in zip 53706
M	Saw "Matrix" more than once

$$P(J)=$$

$$P(C|J)=$$

$$P(C|\sim J)=$$

$$P(Z|J)=$$

$$P(Z|\sim J)=$$

$$P(M|J)=$$

$$P(M|\sim J)=$$

Juniors

people in database

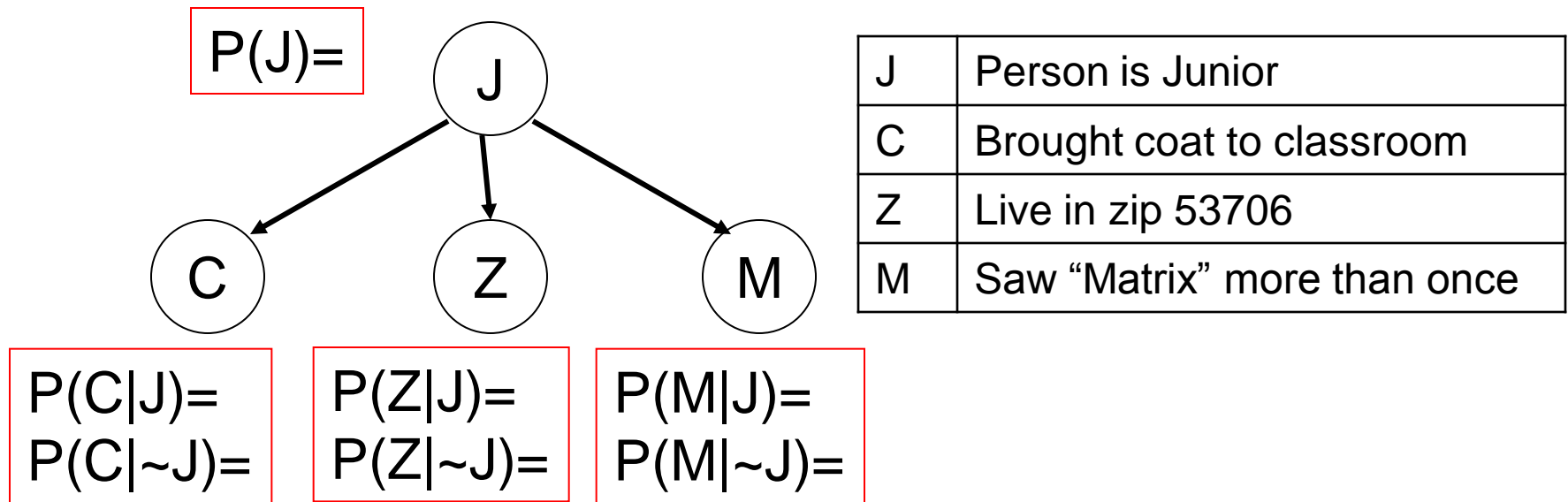
Juniors who saw M>1

Juniors

non-juniors who saw M>1

non-juniors

A special BN: Naïve Bayes Classifiers



- A new person showed up at class wearing an “I live right above the Union Theater where I saw Matrix every night” overcoat.
- What’s the probability that the person is a Junior?

Is the person a junior?

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

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

$$= P(J,C,Z,M) / P(C,Z,M)$$

$$= P(J,C,Z,M) / [P(J,C,Z,M)+P(\sim J,C,Z,M)]$$

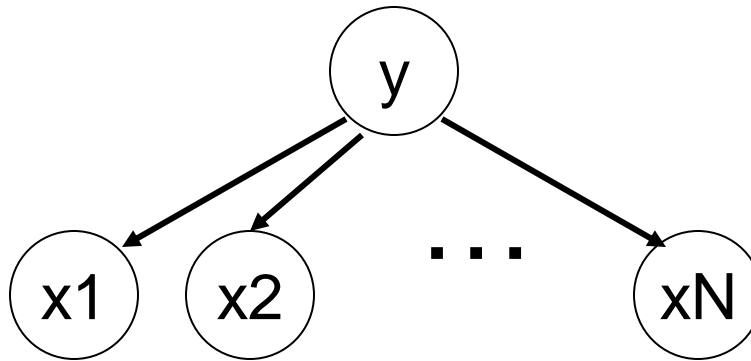
where

$$P(J,C,Z,M)=P(J)P(C|J)P(Z|J)P(M|J)$$

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

BN example: Naïve Bayes

- A special structure:
 - a 'class' node y at root, want $P(y|x_1 \dots x_N)$
 - evidence nodes x (observed features) as leaves
 - **conditional independence between all evidence** (Assumed. Usually wrong. Empirically OK)



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
- Inference by sampling
 - Simple sampling, likelihood weighting, Gibbs
- CPT parameter learning from data
- Naïve Bayes