Bayesian Networks

Machine Learning 10-601B
Seyoung Kim

Many of these slides are derived from William Cohen. Thanks!

Bayesian Networks

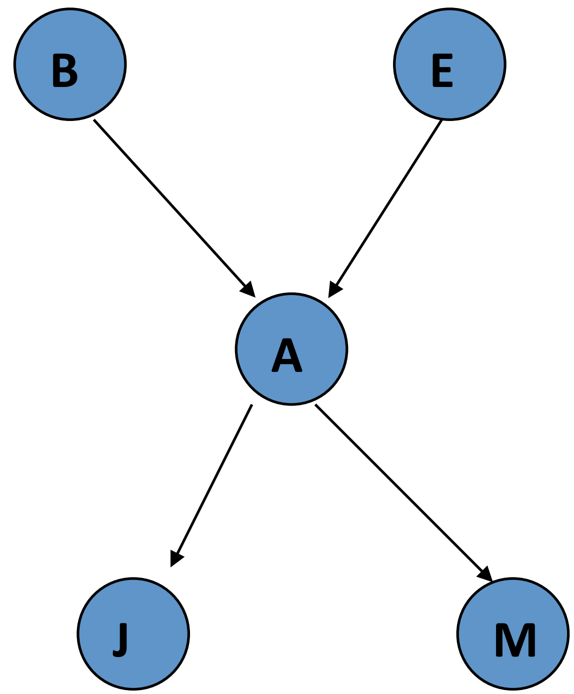
B – Did a burglary occur?

E – Did an earthquake occur?

A – Did the alarm sound off?

M – Mary calls

J – John calls



Bayesian network: Inference

- Once the network is constructed, we can use algorithms for inferring the values of unobserved variables.
- For example, in our previous network the only observed variables are the phone calls. However, what we are really interested in is whether there was a burglary or not.
- How can we determine that?

Inference

- Let's start with a simpler question

  - How can we compute a joint distribution from the network?

  - For example, $P(B, \neg E, A, J, \neg M)$?

- Answer:

  - That's easy, let's use the network

Computing: P(B,¬E,A,J, ¬M)

P(B,¬E,A,J, ¬M) =

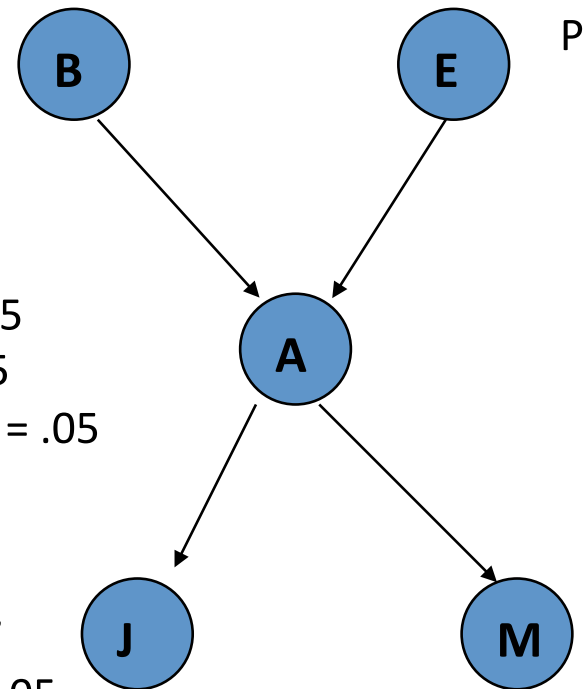
P(B)P(¬E)P(A | B, ¬E) P(J | A)P(¬M | A)

= 0.05*0.9*.85*.7*.2

= 0.005355

P(B)=.05

P(E)=.1



P(A|B,E) =.95
P(A|B,¬E) = .85
P(A| ¬ B,E) =.5
P(A| ¬ B, ¬ E) = .05

P(J|A))=.7
P(J|¬A) = .05

P(M|A) =.8
P(M|¬A) = .15

Computing: $P(B, \neg E, A, J, \neg M)$

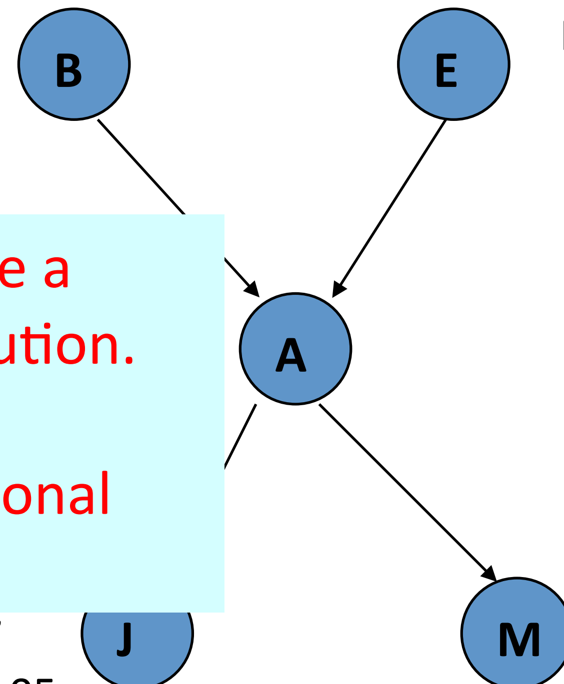
$P(B, \neg E, A, J, \neg M) =$

$P(B)P(\neg E)P(A \mid B, \neg E)\, P(J \mid A)P(\neg M \mid A)$

$= 0.05 * 0.9 * .85 * .7 * .2$

$= 0.005355$

We can easily compute a complete joint distribution. What about partial distributions? Conditional distributions?



Inference

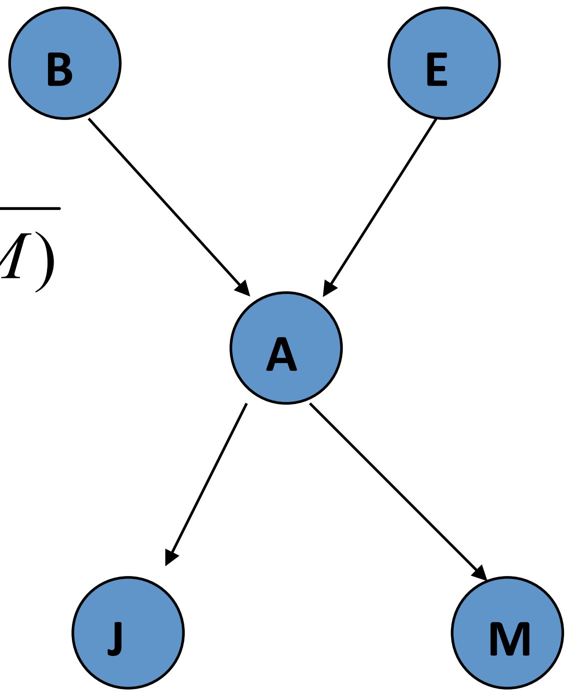
- We are interested in queries of the form:

  P(B | J,¬M)

- This can also be written as:

$$P(B \mid J, \neg M) = \frac{P(B, J, \neg M)}{P(B, J, \neg M) + P(\neg B, J, \neg M)}$$

- How do we compute the new joint?



Inference in Bayesian networks

- We will discuss three methods:

1. Enumeration
2. Variable elimination
3. Stochastic inference

Computing partial joints

$$P(B \mid J, \neg M) = \frac{P(B, J, \neg M)}{P(B, J, \neg M) + P(\neg B, J, \neg M)}$$

Sum all instances with these settings (the sum is over the possible assignments to the other two variables, E and A)

Computing: P(B,J, ¬M)

P(B,J, ¬M) =

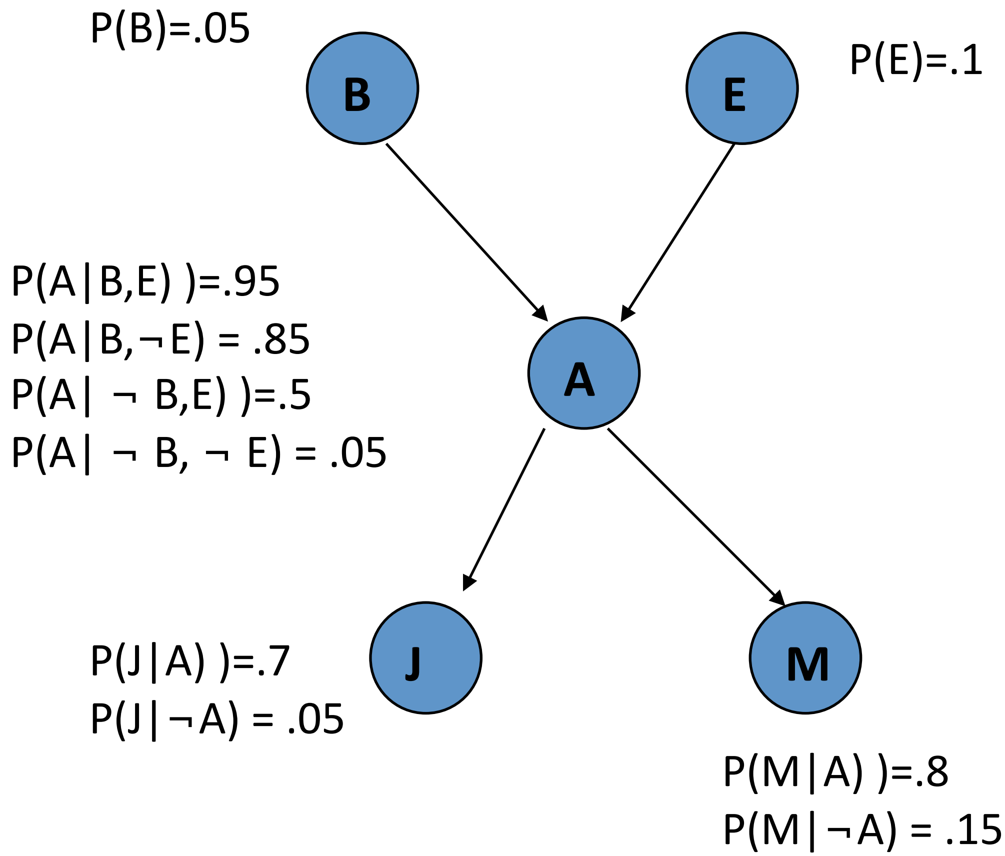
P(B,J, ¬M,A,E) +

P(B,J, ¬M, ¬ A,E) +

P(B,J, ¬M,A, ¬ E) +

P(B,J, ¬M, ¬ A, ¬ E)

= 0.0007+0.00001+0.005+0.0003
= 0.00601



Computing partial joints

$$P(B \mid J, \neg M) = \frac{P(B, J, \neg M)}{P(B, J, \neg M) + P(\neg B, J, \neg M)}$$

Sum all instances with these settings (the sum is over the possible assignments to the other two variables, E and A)

- This method can be improved by re-using calculations (similar to dynamic programming)
- Still, the number of possible assignments is exponential in the number of unobserved variables?
- That is, unfortunately, the best we can do. General querying of Bayesian networks is NP-complete

Inference in Bayesian networks is NP complete (sketch)

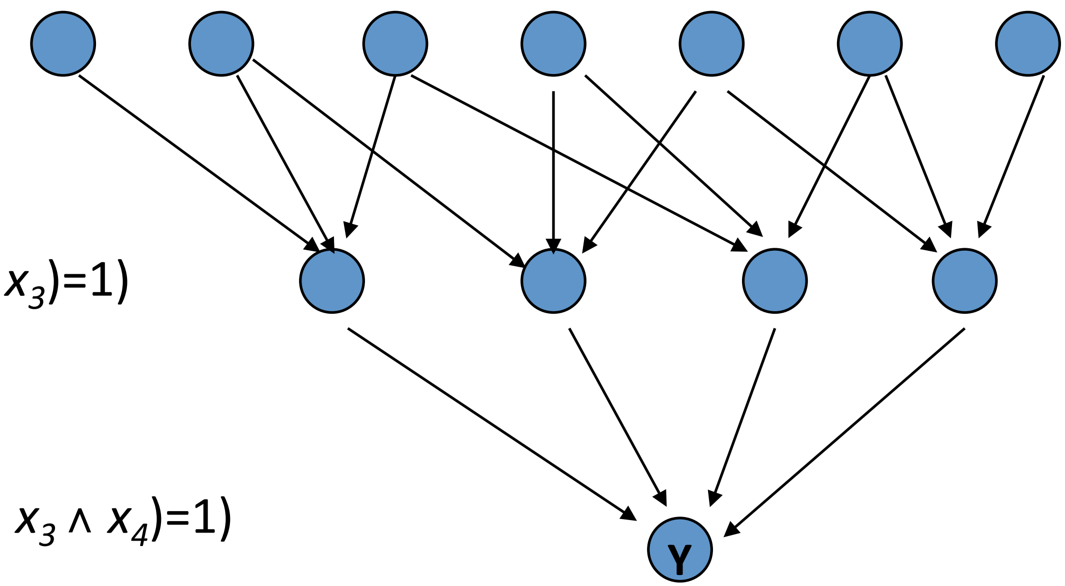
- Reduction from 3SAT
- Recall: 3SAT, find satisfying assignments to the following problem: $(a \vee b \vee c) \wedge (d \vee \neg b \vee \neg c) \ldots$

What is P(Y=1)?

$P(x_i=1) = 0.5$

$P(x_i=1) = P((x_1 \vee x_2 \vee x_3)=1)$

$P(Y=1) = P((x_1 \wedge x_2 \wedge x_3 \wedge x_4)=1)$



Inference in Bayesian networks

- We will discuss three methods:

1. Enumeration
2. Variable elimination
3. Stochastic inference

Variable elimination

P(B,J, ¬M) =

P(B,J, ¬M,A,E)+

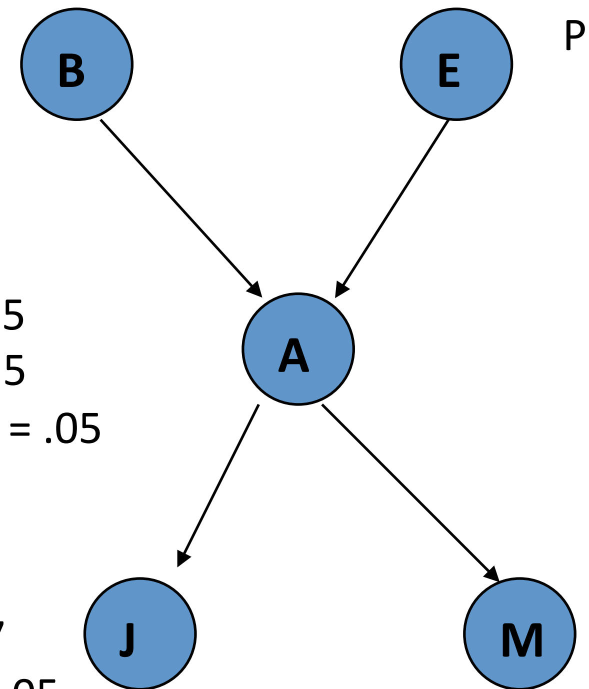
P(B,J, ¬M, ¬ A,E) +

P(B,J,¬M,A, ¬ E) +

P(B,J, ¬M, ¬ A, ¬ E)

= 0.0007+0.00001+0.005+0.0003

= 0.00601



Reuse computations rather than recompute probabilities

Computing: P(B,J, ¬M)

P(B,J, ¬M) =

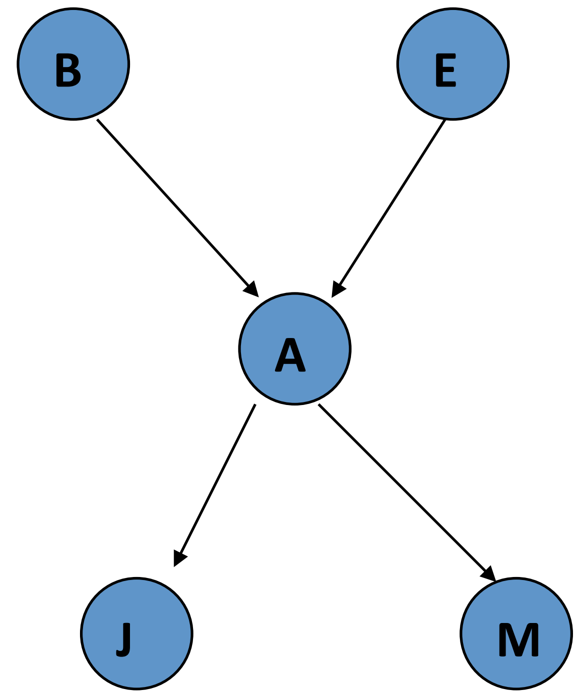
P(B,J, ¬M,A,E)+

P(B,J, ¬M, ¬ A,E) +

P(B,J, ¬M,A, ¬ E) +

P(B,J, ¬M, ¬ A, ¬ E) =

$$\sum_a \sum_e P(B)P(e)P(a \mid B,e)P(M \mid a)P(J \mid a)$$



Store as a function of a and use whenever necessary (no need to recompute each time)

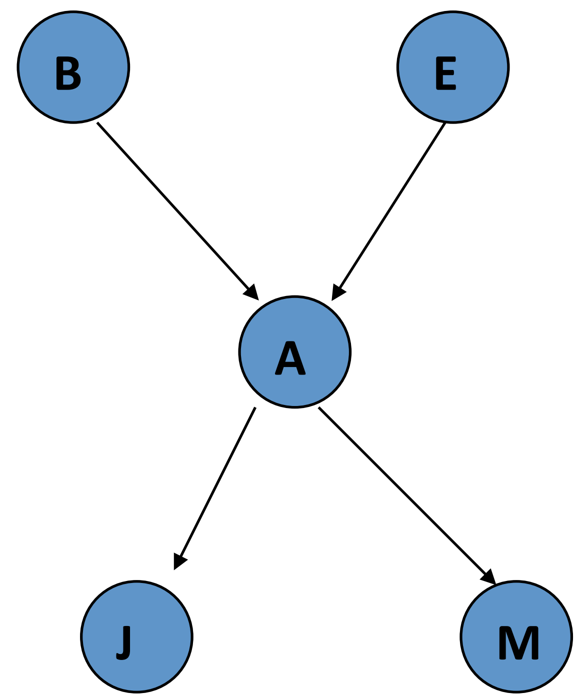
Variable elimination

$$P(B,J,M) = \sum_a \sum_e P(B)P(e)P(a \mid B,e)P(M \mid a)P(J \mid a)$$

$$= P(B)\sum_e P(e)\sum_a P(a \mid B,e)P(M \mid a)P(J \mid a)$$

Set: $$f_M(A) = \begin{pmatrix} P(M \mid A) \\ P(M \mid \neg A) \end{pmatrix}$$

$$f_J(A) = \begin{pmatrix} P(J \mid A) \\ P(J \mid \neg A) \end{pmatrix}$$



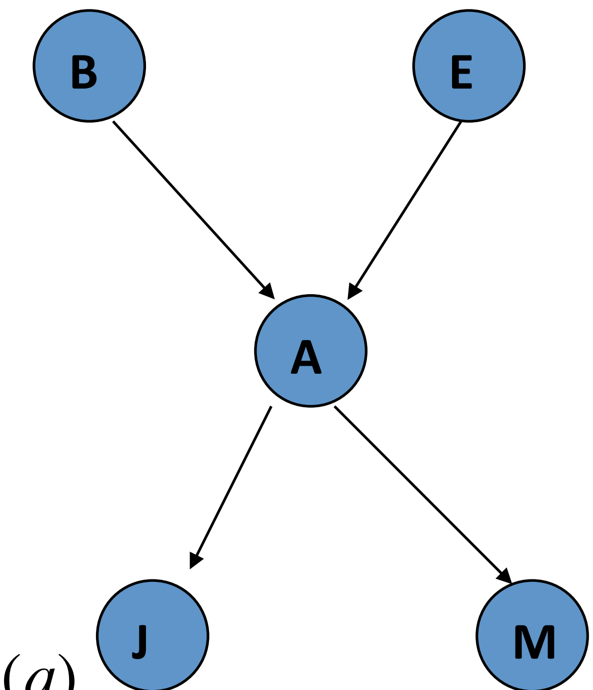
Variable elimination

$$P(B,J,M) = \sum_a \sum_e P(B)P(e)P(a \mid B,e)P(M \mid a)P(J \mid a)$$

$$= P(B)\sum_e P(e)\sum_a \; P(a \mid B,e)P(M \mid a)P(J \mid a)$$

Set: $$f_M(A) = \begin{pmatrix} P(M \mid A) \\ P(M \mid \neg A) \end{pmatrix}$$

$$f_J(A) = \begin{pmatrix} P(J \mid A) \\ P(J \mid \neg A) \end{pmatrix}$$



$$P(B,J,M) = P(B)\sum_e P(e)\sum_a \; P(a \mid B,e)\, f_M(a)\, f_J(a)$$

Variable elimination

$$= P(B)\sum_e P(e)\sum_a P(a \mid B,e) f_M(a) f_J(a)$$

Lets continue with these functions:

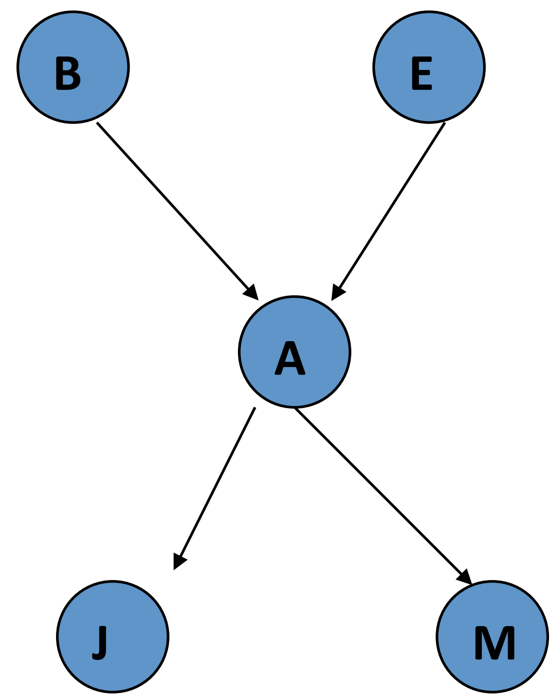
$$f_A(a,B,e) = P(a \mid B,e)$$

We can now define the following function:

$$f_{A,J,M}(B,e) = \sum_a f_A(a,B,e) f_J(a) f_M(a)$$

And so we can write:

$$P(B,J,M) = P(B)\sum_e P(e) f_{A,J,M}(B,e)$$



Variable elimination

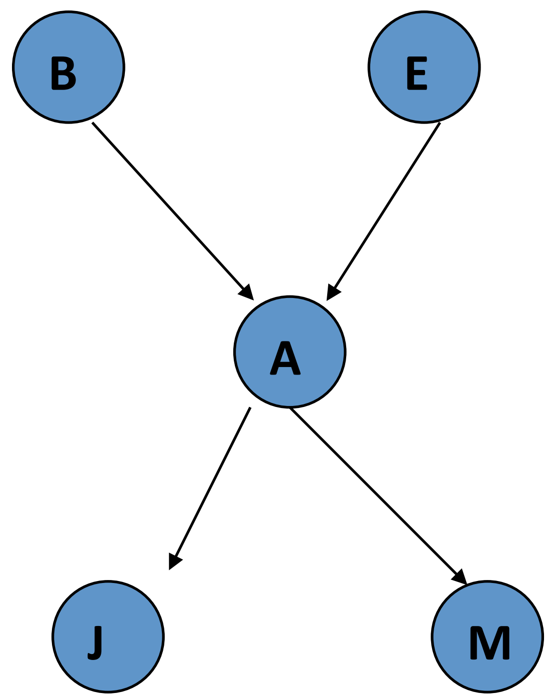
$$P(B,J,M) = P(B)\sum_{e} P(e) f_{A,J,M}(B,e)$$

Lets continue with another function:

$$f_{E,A,J,M}(B) = \sum_{e} P(e) f_{A,J,M}(B,e)$$

And finally we can write:

$$P(B,J,M) = P(B) f_{E,A,J,M}(B)$$

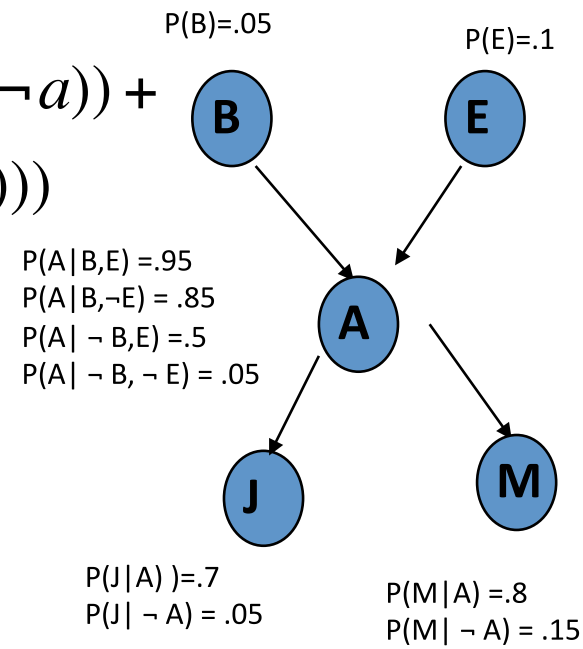


Example

$$P(B,J,M) = P(B) f_{E,A,J,M}(B)$$

$$= 0.05 \sum_{e} P(e) f_{A,J,M}(B,e) = 0.05(0.1 f_{A,J,M}(B,e) + 0.9 f_{A,J,M}(B,\neg e))$$

$$0.05(0.1(0.95 f_J(a) f_M(a) + 0.05 f_J(\neg a) f_M(\neg a)) +$$
$$0.9(.85 f_J(a) f_M(a) + .15 f_J(\neg a) f_M(\neg a)))$$



Calling the same function multiple times

Final computation (normalization)

$$P(B \mid J, \neg M) = \frac{P(B, J, \neg M)}{P(B, J, \neg M) + P(\neg B, J, \neg M)}$$

Algorithm

- *e* - evidence (the variables that are known)
- *vars* - the conditional probabilities derived from the network in reverse order (bottom up)
- For each *var* in *vars*

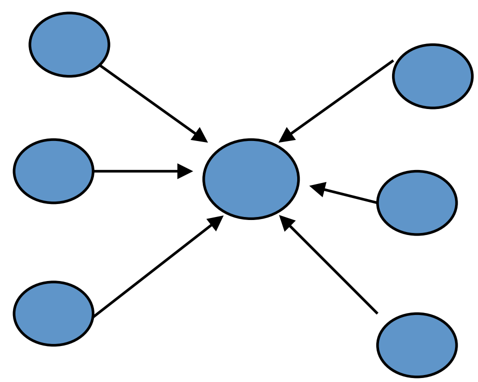
  - *factors* <- make_factor (*var,e*)

  - if *var* is a hidden variable then create a new factor by summing out *var*
- Compute the product of all factors
- Normalize

Computational complexity

- We are reusing computations so we are reducing the running time.
- However, there are still cases in which this algorithm will lead to exponential running time.
- Consider the case of $f_x(y_1 \ldots y_n)$. When factoring x out we would need to account for all possible values of the y's.

Variable elimination can lead to significant cost saving but its efficiency depends on the network structure



Inference in Bayesian networks

- We will discuss three methods:

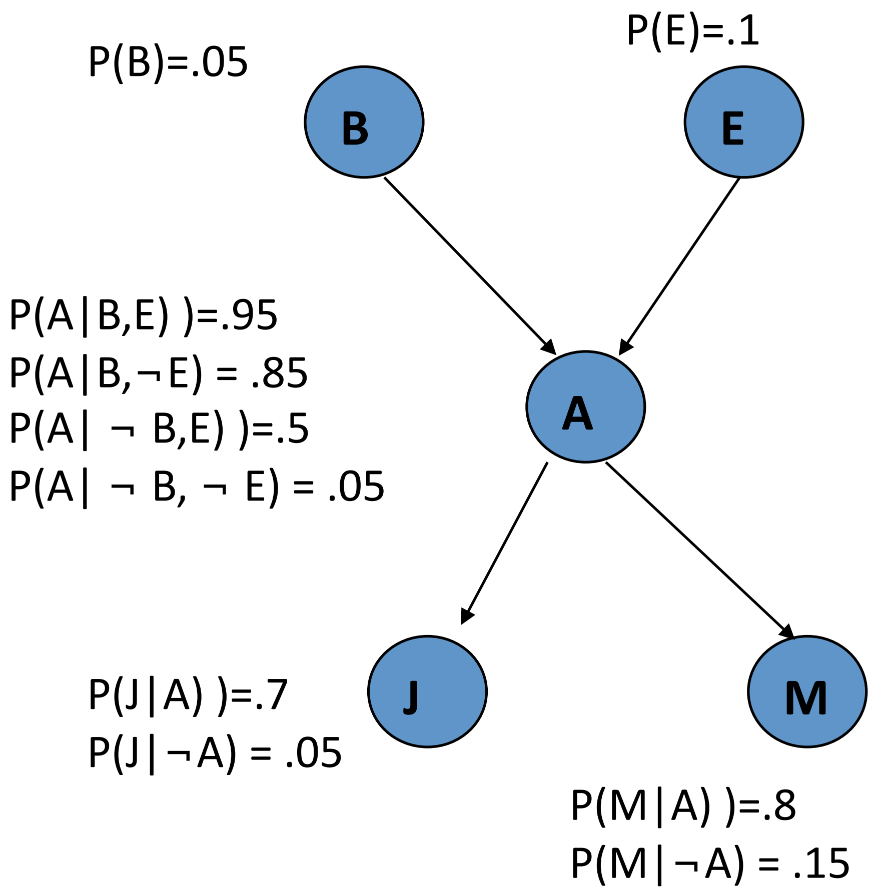
1. Enumeration
2. Variable elimination
3. Stochastic inference

Stochastic inference

- We can easily sample the joint distribution to obtain possible instances

1. Sample the free variable
2. For all other variables:
   - If all parents have been sampled, sample based on conditional distribution

We end up with a new set of assignments for B,E,A,J and M which are a random sample from the joint

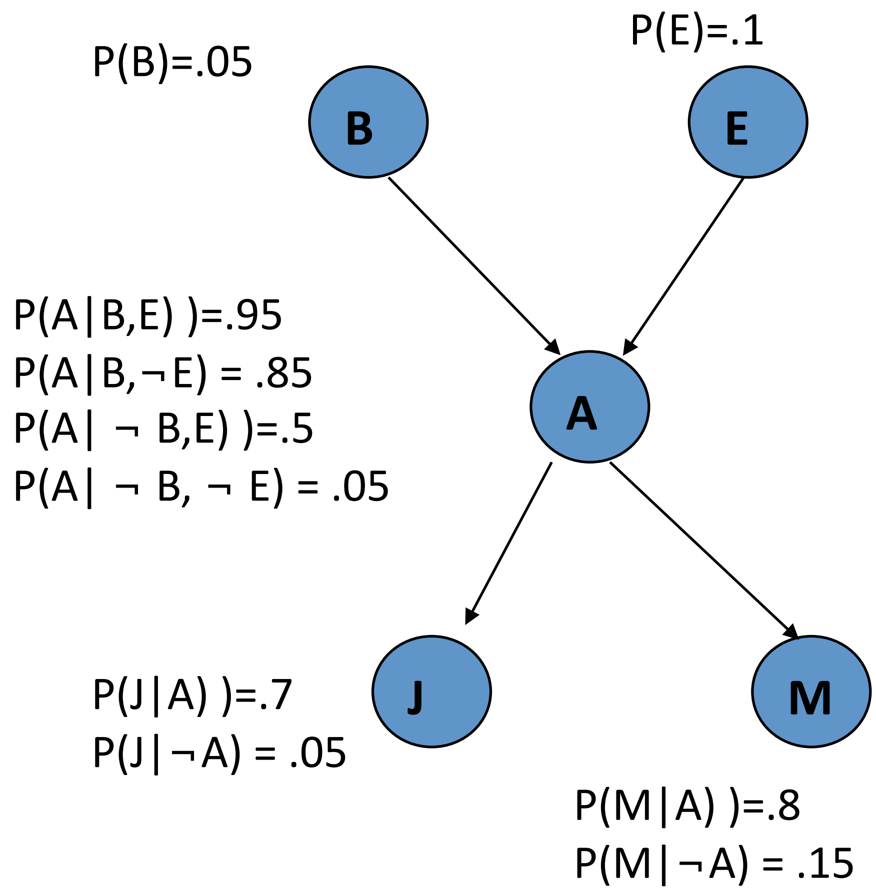


Stochastic inference

- We can easily sample the joint distribution to obtain possible instances

1. Sample the free variable
2. For all other variables:
   - If all parents have been sampled, sample based on conditional distribution

It is always possible to carry out this sampling procedure, why?



Using sampling for inference

- Let’s revisit our problem: Compute $P(B \mid J, \neg M)$
- Looking at the samples we can count:

  - N: total number of samples

  - N_c : total number of samples in which the condition holds $(J, \neg M)$

  - N_B: total number of samples where the joint is true $(B, J, \neg M)$
- For a large enough N
  - $N_c / N \approx P(J, \neg M)$
  - $N_B / N \approx P(B, J, \neg M)$
- And so, we can set

$$P(B \mid J, \neg M) = P(B, J, \neg M) / P(J, \neg M) \approx N_B / N_c$$

Using sampling for inference

- Lets revisit our problem: Compute P(B | J,¬M)
- Looking at the sa[mples we can count:]

  - N: total number of s
  - N_c : total number of
  - N_B: total number of
- For a large enough

  - N_c / N ≈ P(J,¬M)
  - N_B / N ≈ P(B,J,¬M)
- And so, we can set

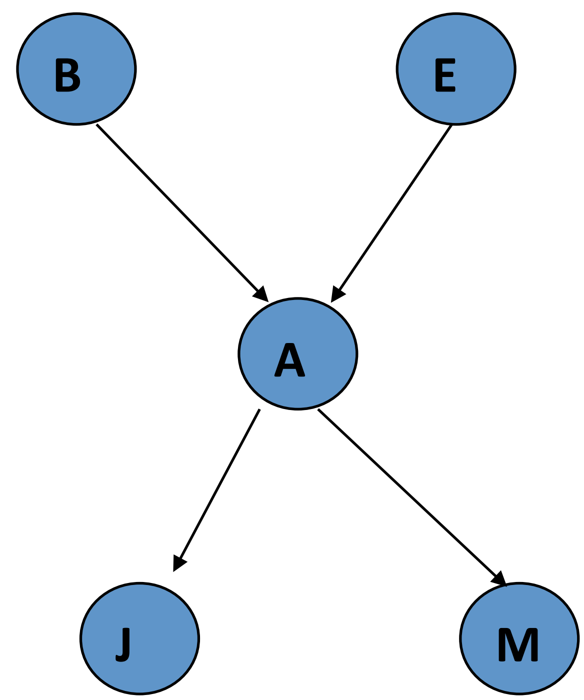
P(B | J,¬M) = P(B,J,¬M) / P(J,¬M) ≈ N_B / N_c

Problem: What if the condition rarely happens?

We would need lots and lots of samples, and most would be wasted

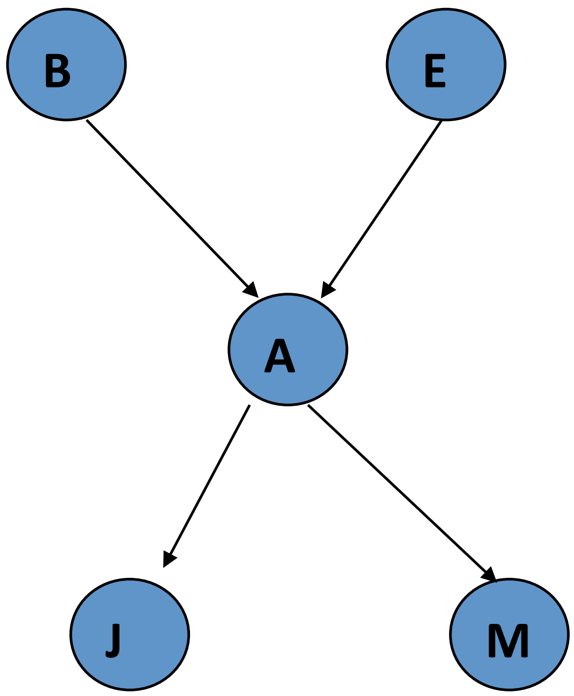
Weighted sampling

- Compute P(B | J,¬M)
- We can manually set the value of J to 1 and M to 0
- This way, all samples will contain the correct values for the conditional variables
- Problems?



Weighted sampling

- Compute P(B | J,¬M)
- Given an assignment to parents, we assign a value of 1 to J and 0 to M.
- We record the *probability* of this assignment ($w = p_1 * p_2$) and we weight the new joint sample by *w*



Weighted sampling algorithm for computing P(B | J,¬M)

- Set $N_B, N_c = 0$
- Sample the joint setting the values for *J* and *M*, compute the weight, *w,* of this sample
- $N_c = N_c + w$
- If $B = 1$, $N_B = N_B + w$

- After many iterations, set

$$P(B \mid J, \neg M) = N_B / N_c$$

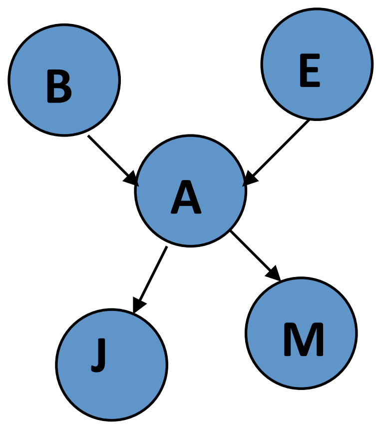
Other inference methods

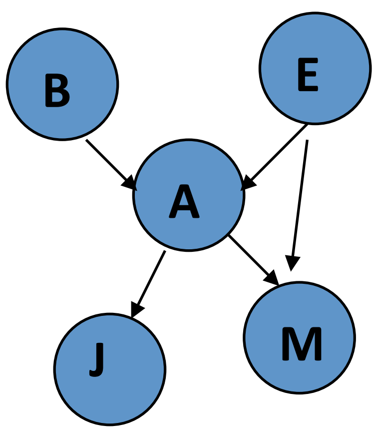
- Convert network to a polytree

  - In a polytree no two nodes have more than one path between them

  - We can convert arbitrary networks to a polytree by clustering (grouping) nodes. For such a graph there is a algorithm which is linear in the number of nodes

  - However, converting into a polytree can result in an exponential increase in the size of the CPTs





Important points

- Bayes rule
- Joint distribution, independence, conditional independence
- Attributes of Bayesian networks
- Constructing a Bayesian network
- Inference in Bayesian networks