CS 2750 Machine Learning
Lecture 17

Bayesian belief networks III (learning and inference)

Milos Hauskrecht
milos@pitt.edu
5329 Sennott Square

Bayesian belief networks (BBNs)

Bayesian belief networks (late 80s, beginning of 90s)

Key features:

- Represent the full joint distribution over the variables more compactly with a **smaller number of parameters**.
- Take advantage of **conditional and marginal independences** among random variables
- **X and Y are independent** $P(X,Y) = P(X)P(Y)$
- **X and Y are conditionally independent given Z**

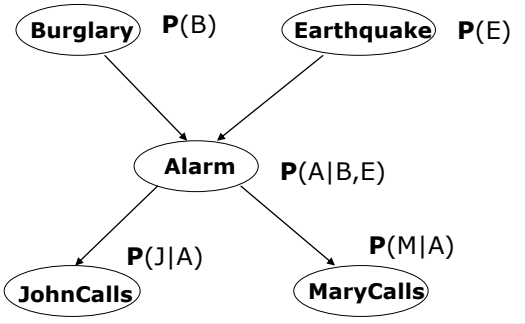
$$P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z)$$

$$P(X \mid Y,Z) = P(X \mid Z)$$

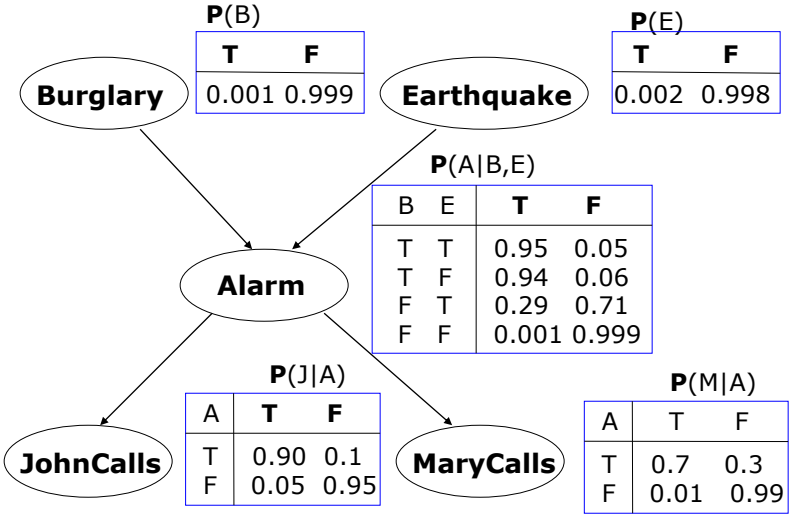
Bayesian belief network

Belief network structure:

- **Nodes** = random variables
  Burglary, Earthquake, Alarm, Mary calls and John calls
- **Links** = direct (causal) dependencies between variables.
  The chance of Alarm being is influenced by Earthquake, The chance of John calling is affected by the Alarm



Bayesian belief network: parameters

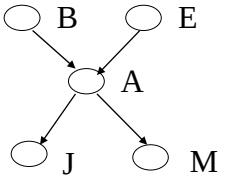


Full joint distribution in BBNs

Full joint distribution is defined in terms of local conditional distributions (obtained via the chain rule):

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

Example:



Assume the following assignment of values to random variables

$$B = T, E = T, A = T, J = T, M = F$$

Then its probability is:

$$P(B = T, E = T, A = T, J = T, M = F) =$$

$$P(B = T)P(E = T)P(A = T \mid B = T, E = T)P(J = T \mid A = T)P(M = F \mid A = T)$$

Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

- **What did we save?**

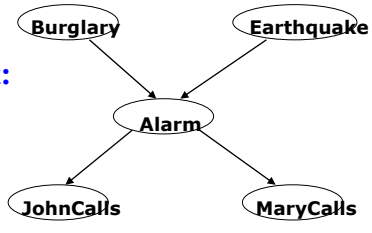
Alarm example: 5 binary (True, False) variables

# of parameters of the full joint:

$$2^5 = 32$$

One parameter depends on the rest:

$$2^5 - 1 = 31$$



# of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$

One parameter in every conditional depends on the rest:

$$2^2 + 2(2) + 2(1) = 10$$

Learning of BBN

Learning.

- **Learning of parameters of conditional probabilities**
- **Learning of the network structure**

Variables:

- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

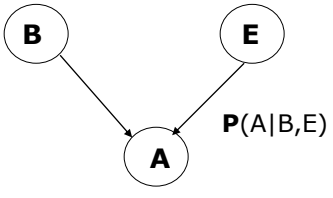
Next:

- **Learning of the parameters of BBN**
- **Values for all variables are observable**

Estimation of parameters of BBN

- **Idea:** decompose the estimation problem for the full joint over a large number of variables to a set of smaller estimation problems corresponding to local parent-variable conditionals.
- **Example:** Assume A,E,B are binary with *True*, *False* values

Learning of P(A|B,E) = 4 estimation problems



- **P**(A|B=T,E=T)
- **P**(A|B=T,E=F)
- **P**(A|B=F,E=T)
- **P**(A|B=F,E=F)

- **Assumption that enables the decomposition:** parameters of conditional distributions are independent

---

Estimates of parameters of BBN

- Two assumptions that permit the decomposition:
  - **Sample independence**

$$P(D \mid \Theta, \xi) = \prod_{u=1}^{N} P(D_u \mid \Theta, \xi)$$

  - **Parameter independence**

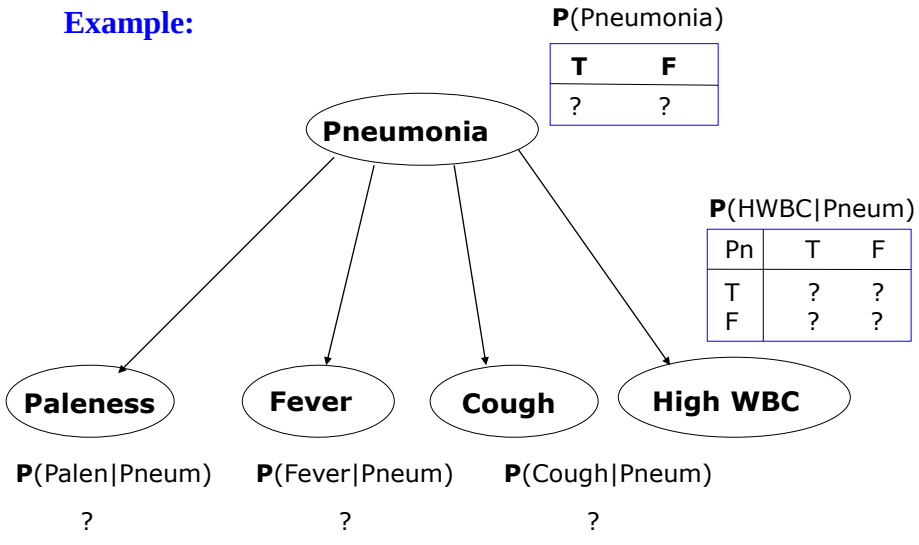
# of nodes
# of parents' values

$$p(\Theta \mid D, \xi) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\theta_{ij} \mid D, \xi)$$

Parameters of **each conditional** (one for every assignment of values to parent variables) can be learned independently

Learning of BBN parameters. Example.

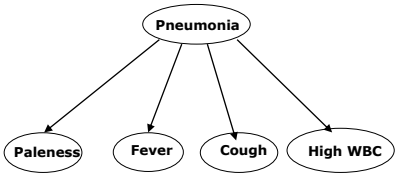
Example:



Learning of BBN parameters. Example.

Data D (different patient cases):

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



Estimates of parameters of BBN

- Much like multiple **coin toss or roll of a dice** problems.
- A "smaller" learning problem corresponds to the learning of exactly one conditional distribution
- **Example:**

$$\mathbf{P}(Fever \mid Pneumonia = T)$$

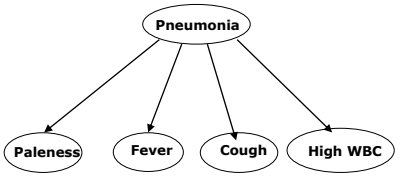
- **Problem:** How to pick the data to learn?

Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 1: Select data points with Pneumonia=T

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F

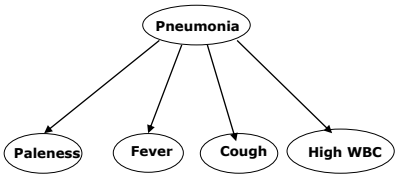


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 1: Ignore the rest

Pal	Fev	Cou	HWB	Pneu
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T

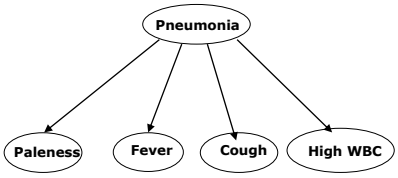


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 2: Select values of the random variable defining the distribution of Fever

Pal	Fev	Cou	HWB	Pneu
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T

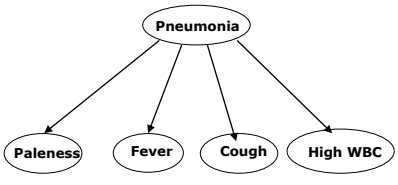


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 2: Ignore the rest

Fev
F
F
T
T
T

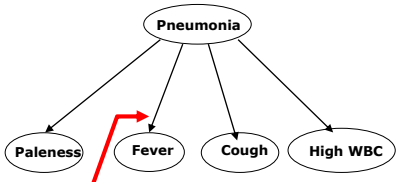


Learning of BBN parameters. Example.

Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 3a: Learning the ML estimate

Fev
F
F
T
T
T



$\mathbf{P}(Fever \mid Pneumonia = T)$

	T	F
Pneum =T	0.6	0.4

Learning of BBN parameters. Bayesian learning.

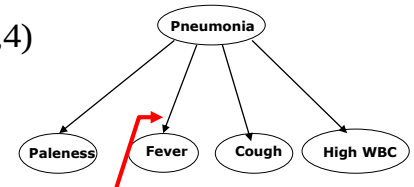
Learn: $\mathbf{P}(Fever \mid Pneumonia = T)$

Step 3b: Learning the Bayesian posterior

Assume the prior

$\theta_{Fever|Pneumonia=T} \sim Beta(3,4)$

Fev
F
F
T
T
T



Posterior:

$\theta_{Fever|Pneumonia=T} \sim Beta(6,6)$

$\theta^{MAP}_{Fever|Pneumonia=T} = \frac{6-1}{6+6-2} = 0.5$

MAP estimates

	T	F
Pneum =T	0.5	0.5

Estimates of parameters of BBN

Much like multiple **coin toss or roll of a dice** problems.

- A "smaller" learning problem corresponds to the learning of exactly one conditional distribution

Example:

$$\mathbf{P}(Fever \mid Pneumonia = T)$$

Problem: How to pick the data to learn?

Answer:

1. Select data points with Pneumonia=T
   (ignore the rest)
2. Focus on (select) only values of the random variable defining the distribution (Fever)
3. Learn the parameters of the local conditionals the same way as we learned the parameters of a biased coin or a die

---

Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
- Simplifies the representation and learning of a model
- Can be used for solving various **inference tasks**:
  - **Diagnostic task. (from effect to cause)**

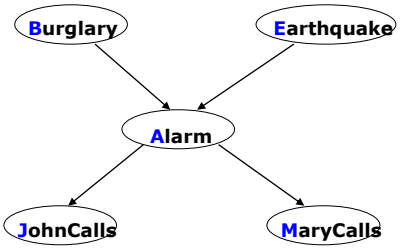
    $$\mathbf{P}(Burglary \mid JohnCalls = T)$$
  - **Prediction task. (from cause to effect)**

    $$\mathbf{P}(JohnCalls \mid Burglary = T)$$
  - **Other probabilistic queries** (queries on joint distributions).

    $$\mathbf{P}(Alarm)$$
- **Main question:** Can we take advantage of independences to construct special algorithms and speeding up the inference?

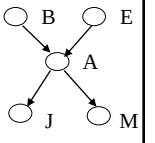
Inference in Bayesian network

- **Bad news:**
  - Exact inference problem in BBNs is NP-hard (Cooper)
  - Approximate inference is NP-hard (Dagum, Luby)
- **But** very often we can achieve significant improvements
- Assume our Alarm network



- Assume we want to compute: $P(J=T)$

Inference in Bayesian networks



Computing: $P(J=T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J=T) =$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(B=b, E=e, A=a, J=T, M=m)$$

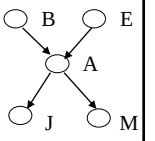
$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T \mid A=a)P(M=m \mid A=a)P(A=a \mid B=b, E=e)P(B=b)P(E=e)$$

Computational cost:

Number of additions: ?

Number of products: ?

Inference in Bayesian networks



Computing: $P(J=T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J=T) =$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(B=b,E=e,A=a,J=T,M=m)$$

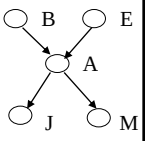
$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

Computational cost:

Number of additions: **15** (adding 16 terms)

Number of products: ?

Inference in Bayesian networks



Computing: $P(J=T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals

$$P(J=T) =$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(B=b,E=e,A=a,J=T,M=m)$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

Computational cost:

Number of additions: **15**

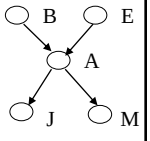
Number of products: 16*4=**64** (4 multiplications repeated 16 times)

Inference in Bayesian networks

How to compute sums and products more efficiently?

$$\sum_x af(x) = a\sum_x f(x)$$

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

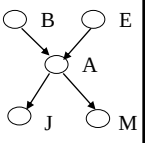
$$P(J=T) =$$
$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

- Use variable e and its sum and

$$\sum_x af(x) = a\sum_x f(x)$$

$$= \sum_{b\in T,F}\sum_{a\in T.F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(B=b)[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)]$$

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way

(multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

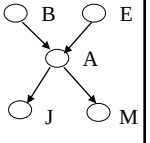
$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{a\in T.F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]$$

• • •

$$=\sum_{a\in T,F}P(J=T\mid A=a)[\sum_{m\in T,F}P(M=m\mid A=a)][\sum_{b\in T,F}P(B=b)[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)]]$$

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way

(multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

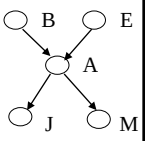
$$=\sum_{a\in T,F}P(J=T\mid A=a)\left[\sum_{m\in T,F}P(M=m\mid A=a)\right]\left[\sum_{b\in T,F}P(B=b)\left[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)\right]\right.$$

1

Computational cost:

Number of additions: ?

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T) =$$

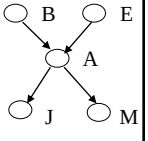
$$= \sum_{a \in T,F} P(J=T \mid A=a) \left[\sum_{m \in T,F} P(M=m \mid A=a) \right] \left[\sum_{b \in T,F} P(B=b) \left[\sum_{e \in T,F} P(A=a \mid B=b, E=e) P(E=e) \right] \right]$$

2*1

Computational cost:

Number of additions: ?

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T) =$$

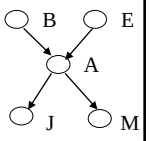
$$= \sum_{a \in T,F} P(J=T \mid A=a) \left[\sum_{m \in T,F} P(M=m \mid A=a) \right] \left[\sum_{b \in T,F} P(B=b) \left[\sum_{e \in T,F} P(A=a \mid B=b, E=e) P(E=e) \right] \right]$$

2*2*1

Computational cost:

Number of additions: ?

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

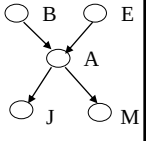
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

2*1 2*2*1

Computational cost:

Number of additions: ?

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

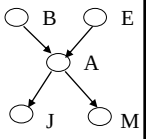
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

1 2*1 2*1 2*2*1

Computational cost:

Number of additions: **?**

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way

(multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

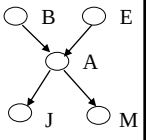
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

1 2*1 2*1 2*2*1

Computational cost:

Number of additions: 1+2*[1+1+2*1]=**9**

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way

(multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

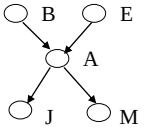
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

1

Computational cost:

Number of products: ?

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

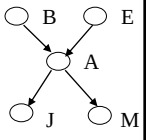
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

2*2 *2*1

Computational cost:

Number of products: ?

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

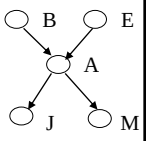
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

2*2 2*2*1 2*2 *2*1

Computational cost:

Number of products: ?

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

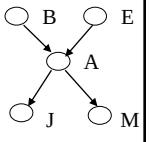
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

2*2 2*2*1 2*2 *2*1

Computational cost:

Number of products: 2*[2+2*(1+2*1)]=**16**

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
  (multiplications by constants can be taken out of the sum)

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{a\in T.F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]$$

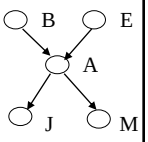
$$=\sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]\left[\sum_{b\in T,F} P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]\right]$$

Computational cost:

Number of additions: 1+2*[1+1+2*1]=**9**

Number of products: 2*[2+2*(1+2*1)]=**16**

Variable elimination

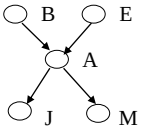


- **Variable elimination:**
  - Similar idea but interleave sum and products one variable at the time during inference
  - E.g. Query $P(J=T)$ requires to eliminate A,B,E,M and this can be done in different order

$$P(J=T)=$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

Variable elimination



Assume order: M, E, B,A to calculate $P(J=T)$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F}P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)\left[\sum_{m\in T,F}P(M=m\mid A=a)\right]$$

$$=\sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)\ \ 1$$

$$=\sum_{a\in T,F}\sum_{b\in T,F}P(J=T\mid A=a)P(B=b)\left[\sum_{e\in T,F}P(A=a\mid B=b,E=e)P(E=e)\right]$$

$$=\sum_{a\in T,F}\sum_{b\in T,F}P(J=T\mid A=a)P(B=b)\,\tau_1(A=a,B=b)$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)\left[\sum_{e\in T,F}P(B=b)\,\tau_1(A=a,B=b)\right]$$

$$=\sum_{a\in T,F}P(J=T\mid A=a)\ \ \tau_2(A=a)\quad=\boxed{P(J=T)}$$

Inference in Bayesian network

- **Exact inference algorithms:**
  - **Variable elimination**
  - Recursive decomposition (Cooper, Darwiche)
  - Symbolic inference (D'Ambrosio)
  - Belief propagation algorithm (Pearl)
  - Clustering and joint tree approach (Lauritzen, Spiegelhalter)
  - Arc reversal (Olmsted, Schachter)

- **Approximate inference algorithms:**
  - **Monte Carlo methods:**
    - Forward sampling, Likelihood sampling
  - Variational methods