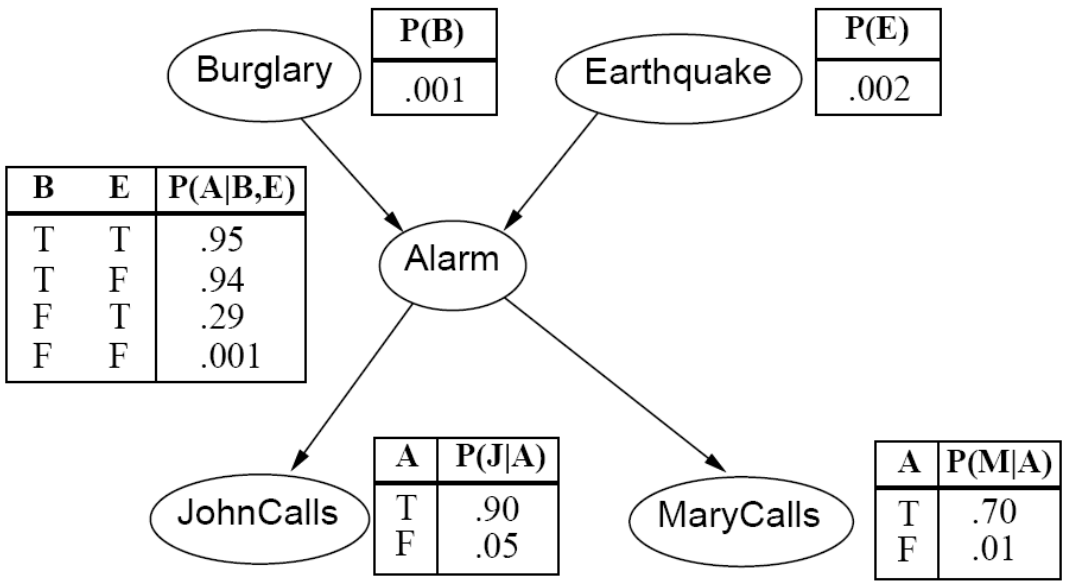
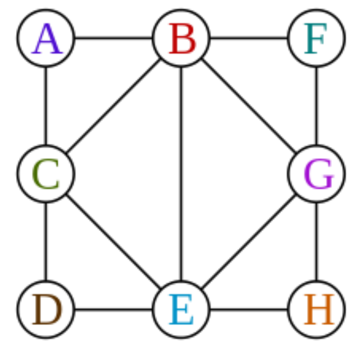
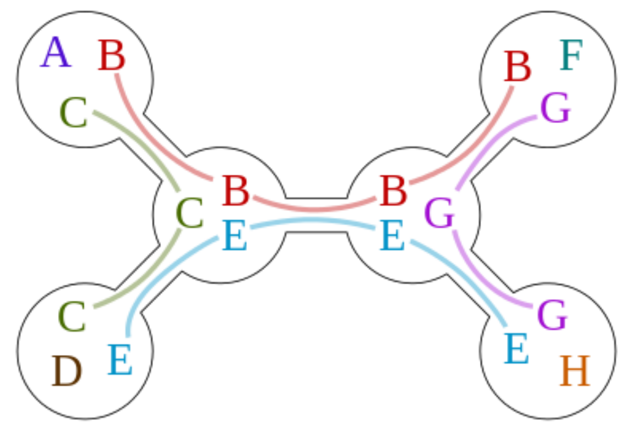
CS 440/ECE448 Lecture 19: Bayes Net Inference

Mark Hasegawa-Johnson, 3/2019

Including slides by Svetlana Lazebnik, 11/2016







Bayes Network Inference & Learning

Bayes net is a **memory-efficient model** of dependencies among:

- Query *variables:* **X**
- *Evidence* (*observed*) variables and their values: **E** = **e**
- *Unobserved* variables: **Y**

Inference problem: answer questions about the query variables given the evidence variables

- This can be done using the posterior distribution P(**X** | **E** = **e**)
- The posterior can be derived from the full joint P(**X**, **E**, **Y**)
- How do we make this **computationally efficient?**

Learning problem: given some training examples, how do we learn the parameters of the model?

- Parameters = p(variable|parents), for each variable in the net

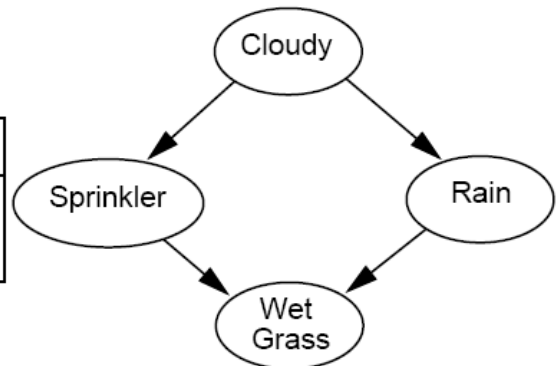
Outline

- Inference Examples
- Inference Algorithms
  - Trees: Sum-product algorithm
  - Poly-trees: Junction tree algorithm
  - Graphs: No polynomial-time algorithm
- Parameter Learning

Practice example 1

- Variables: *Cloudy, Sprinkler, Rain, Wet Grass*

P(C)
.50



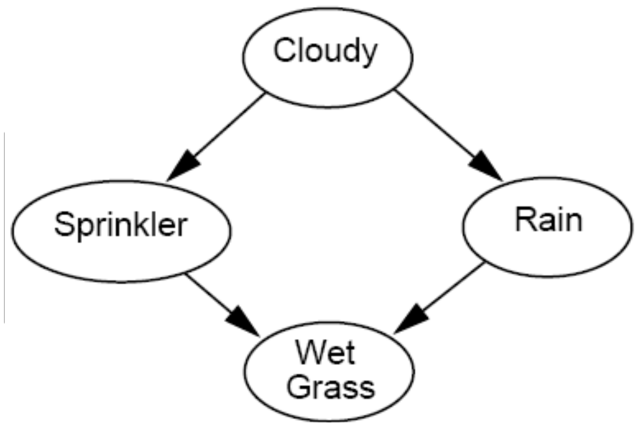
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Practice example 1

- Given that the grass is wet, what is the probability that it has rained?



$$P(r \mid w) = \frac{P(r,w)}{P(w)} = \frac{\sum_{C=c,S=s} P(c,s,r,w)}{\sum_{C=c,S=s,R=r} P(c,s,r,w)}$$

$$= \frac{\sum_{C=c,S=s} P(c)P(s \mid c)P(r \mid c)P(w \mid r,s)}{\sum_{C=c,S=s,R=r} P(c)P(s \mid c)P(r \mid c)P(w \mid r,s)}$$

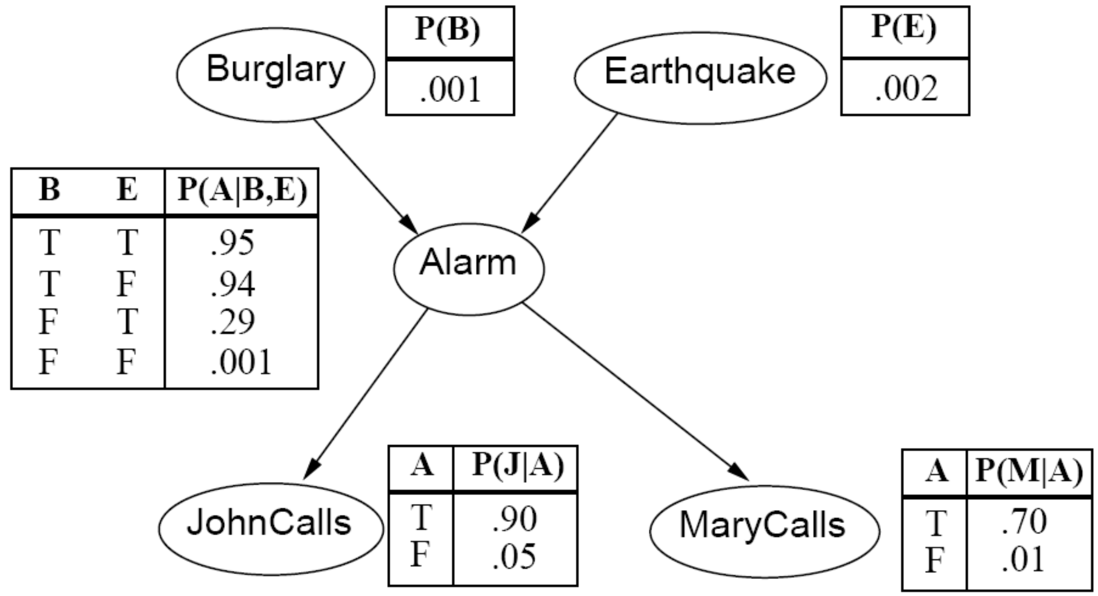
Practice Example #2

- Suppose you have an observation, for example, “Jack called” (J=1)
- You want to know: was there a burglary?
- You need

$$P(B = 1|J = 1) = \frac{P(B, J = 1)}{\sum_b P(B = b, J = 1)}$$

- So you need to compute the table P(B,J) for all possible settings of (B,J)

Bayes Net Inference: The Hard Way



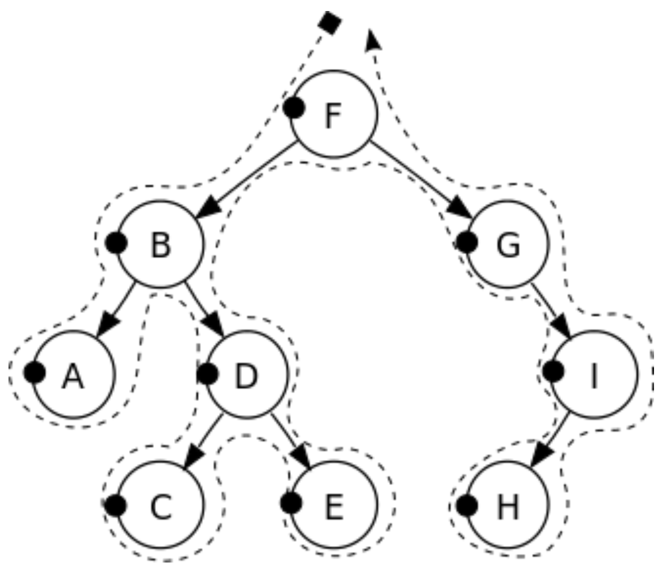
1. P(B,E,A,J,M)=P(B)P(E)P(A|B,E)P(J|A)P(M|A)
2. $P(B,J) = \sum_E \sum_A \sum_M P(B,E,A,J,M)$

Exponential complexity (#P-hard, actually): N variables, each of which has K possible values $\Rightarrow$ $O\{K^N\}$ time complexity

Is there an easier way?

- Tree-structured Bayes nets: the sum-product algorithm
    - Quadratic complexity, $O\{NK^3\}$
- Polytrees: the junction tree algorithm
    - Pseudo-polynomial complexity, $O\{NK^M\}$, for M<N
- Arbitrary Bayes nets: #P complete, $O\{K^N\}$
    - The SAT problem is a Bayes net!
- Parameter Learning

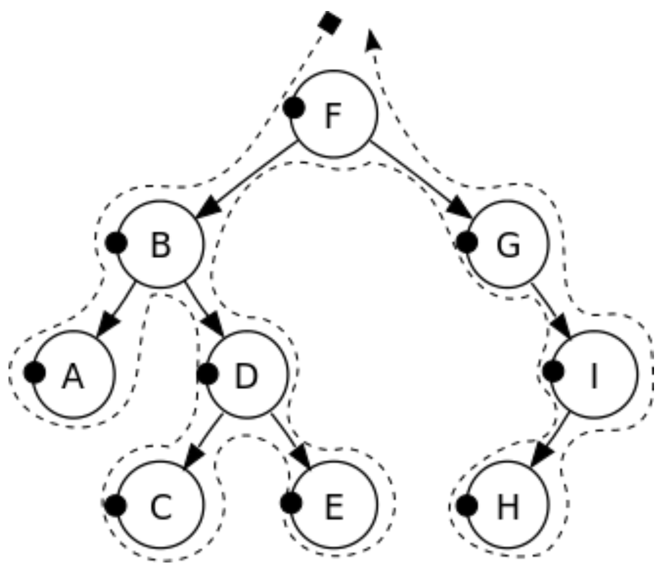
1. Tree-Structured Bayes Nets



- Suppose these are all binary variables.
- We observe E=1
- We want to find P(H=1|E=1)
- Means that we need to find both P(H=0,E=1) and P(H=1,E=1) because

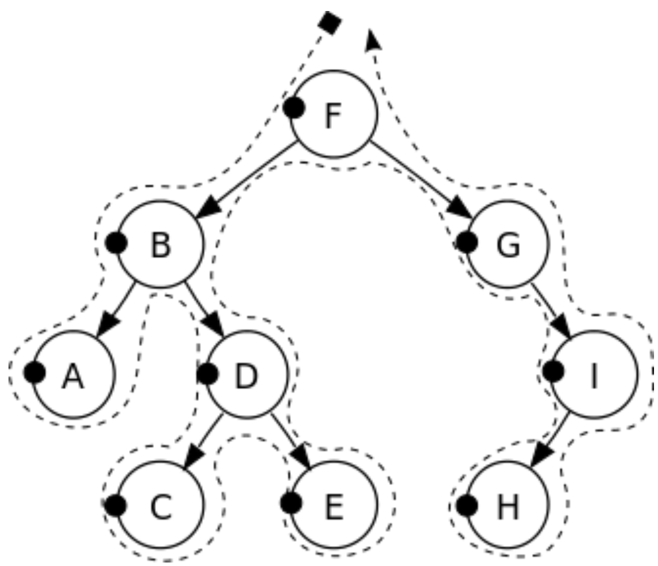
$$P(H=1|E=1)=\frac{P(H=1,E=1)}{\sum_h P(H=h,E=1)}$$

The Sum-Product Algorithm (Belief Propagation)



- Find the only undirected path from the evidence variable to the query variable (EDBFGIH)
- Find the directed root of this path P(F)
- Find the joint probabilities of root and evidence: P(F=0,E=1) and P(F=1,E=1)
- Find the joint probabilities of query and evidence: P(H=0,E=1) and P(H=1,E=1)
- Find the conditional probability P(H=1|E=1)

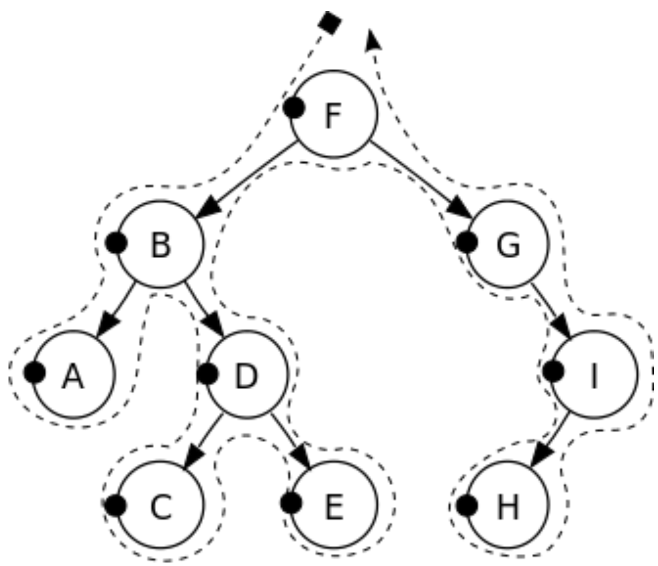
The Sum-Product Algorithm (Belief Propagation)



Starting with the root P(F), we find P(F,E) by alternating product steps and sum steps:

1. Product: $P(B,D,F)=P(F)P(B|F)P(D|B)$
2. Sum: $P(D,F) = \sum_{B=0}^{1} P(B,D,F)$
3. Product: $P(D,E,F)=P(D,F)P(E|D)$
4. Sum: $P(E,F) = \sum_{D=0}^{1} P(D,E,F)$

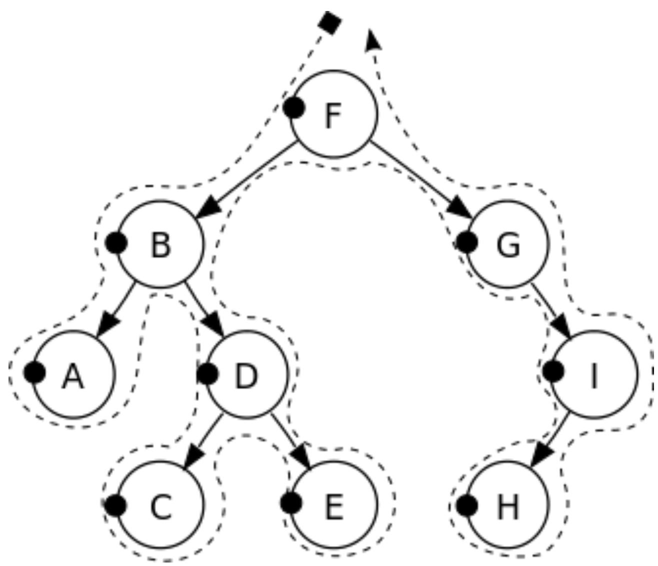
The Sum-Product Algorithm (Belief Propagation)



Starting with the root P(E,F), we find P(E,H) by alternating product steps and sum steps:

1. Product: P(E,F,G)=P(E,F)P(G|F)
2. Sum: $P(E,G) = \sum_{F=0}^{1} P(E,F,G)$
3. Product: P(E,G,I)=P(E,G)P(I|G)
4. Sum: $P(E,I) = \sum_{G=0}^{1} P(E,G,I)$
5. Product: P(E,H,I)=P(E,I)P(I|G)
6. Sum: $P(E,H) = \sum_{I=0}^{1} P(E,H,I)$

Time Complexity of Belief Propagation



- Each product step generates a table with 3 variables
- Each sum step reduces that to a table with 2 variables
- If each variable has K values, and if there are $O\{N\}$ variables on the path from evidence to query, then time complexity is $O\{NK^3\}$

Time Complexity of Bayes Net Inference

- Tree-structured Bayes nets: the sum-product algorithm
    - Quadratic complexity, $O\{NK^3\}$
- Polytrees: the junction tree algorithm
    - Pseudo-polynomial complexity, $O\{NK^M\}$, for M<N
- Arbitrary Bayes nets: #P complete, $O\{K^N\}$
    - The SAT problem is a Bayes net!
- Parameter Learning

2. The Junction Tree Algorithm

a. Moralize the graph (identify each variable’s Markov blanket)
b. Triangulate the graph (eliminate undirected cycles)
c. Create the junction tree (form cliques)
d. Run the sum-product algorithm on the junction tree

2.a. Markov Blanket

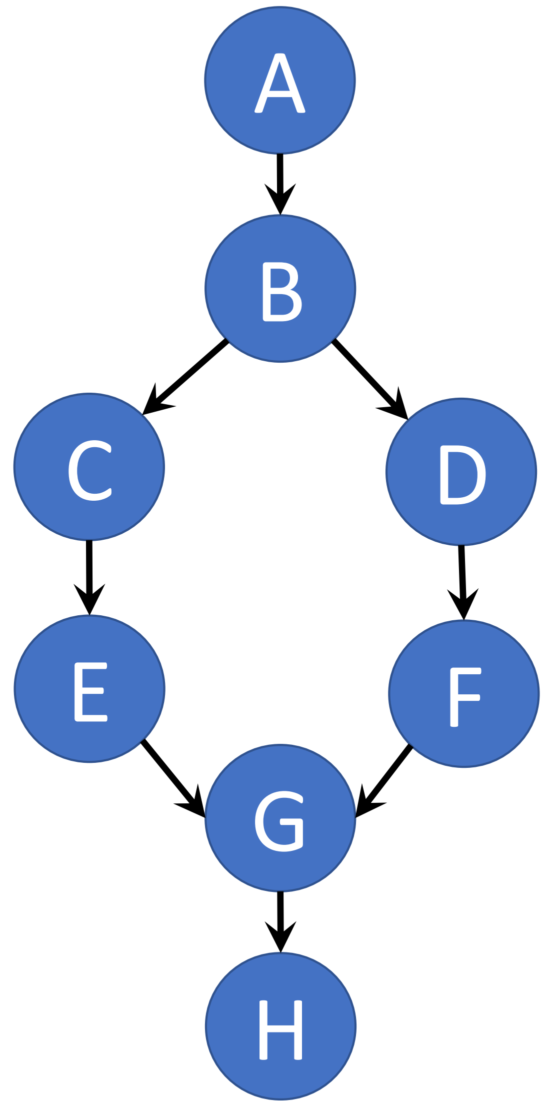
- Suppose there is a Bayes net with variables A,B,C,D,E,F,G,H
- The "Markov blanket" of variable F is D,E,G if

$$P(F|A,B,C,D,E,G,H)$$
$$= P(F|D,E,G)$$

2.a. Markov Blanket

- Suppose there is a Bayes net with variables A,B,C,D,E,F,G,H
- The "Markov blanket" of variable F is D,E,G if

$$P(F|A,B,C,D,E,G,H)$$
$$= P(F|D,E,G)$$

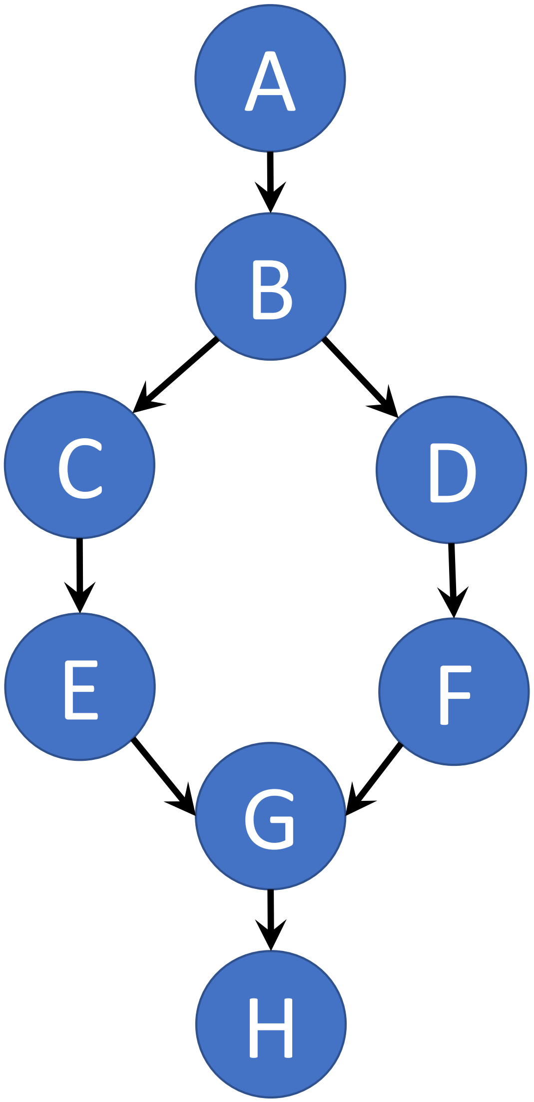


2.a. Markov Blanket

- The "Markov blanket" of variable F is D,E,G if

$$P(F|A,B,C,D,E,G,H)$$
$$= P(F|D,E,G)$$

- How can we prove that?
- $P(A,\ldots,H) = P(A)P(B|A)\ \ldots$
- Which of those terms include F?



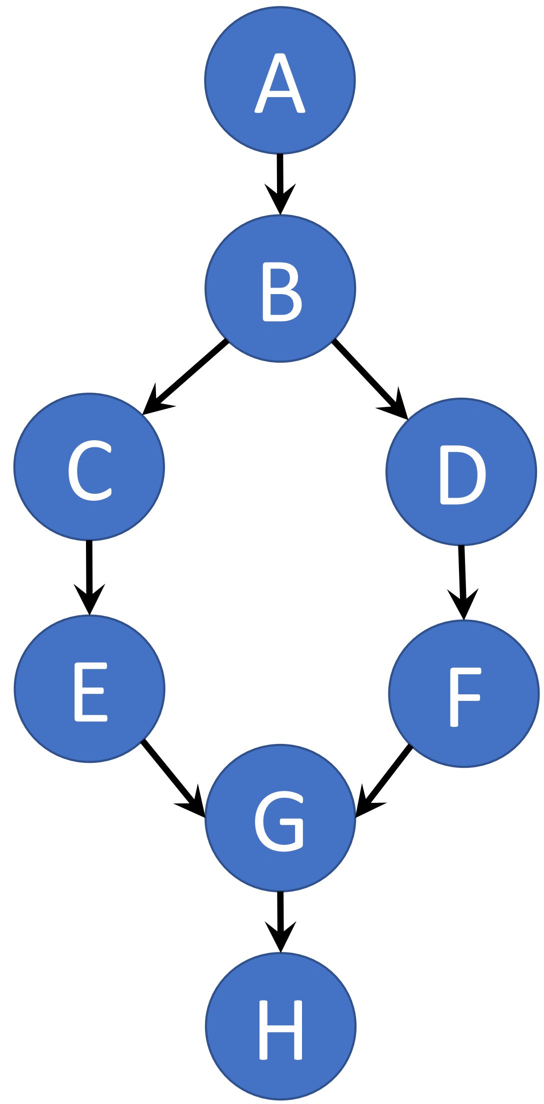
2.a. Markov Blanket

- Which of those terms include F?
- Only these two:

$$P(F|D)$$

and

$$P(G|E,F)$$



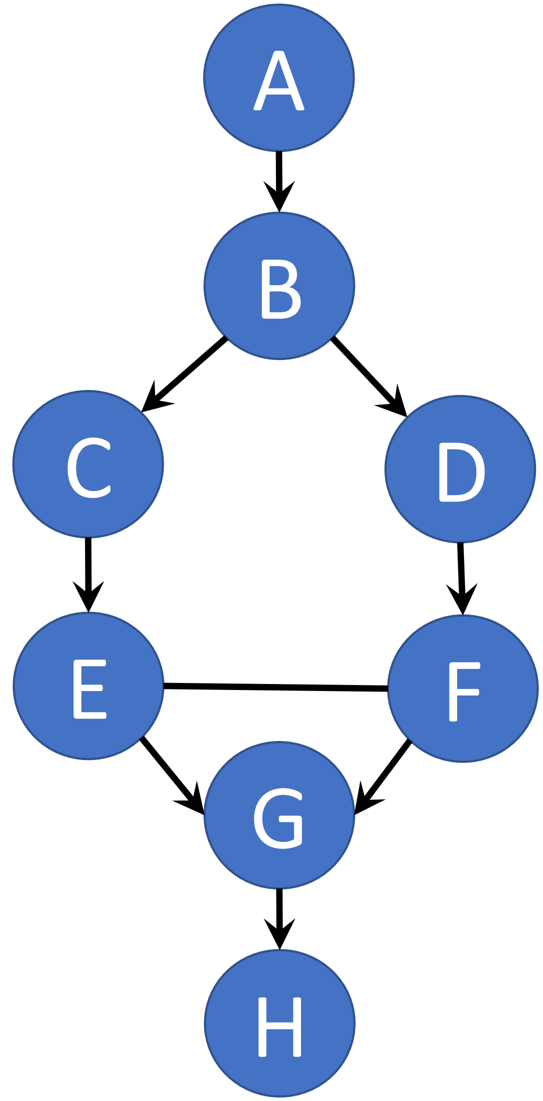
2.a. Markov Blanket

The Markov Blanket of variable F includes only its immediate family members:

- Its parent, D
- Its child, G
- The other parent of its child, E

Because P(F|A,B,C,D,E,G,H)

= P(F|D,E,G)

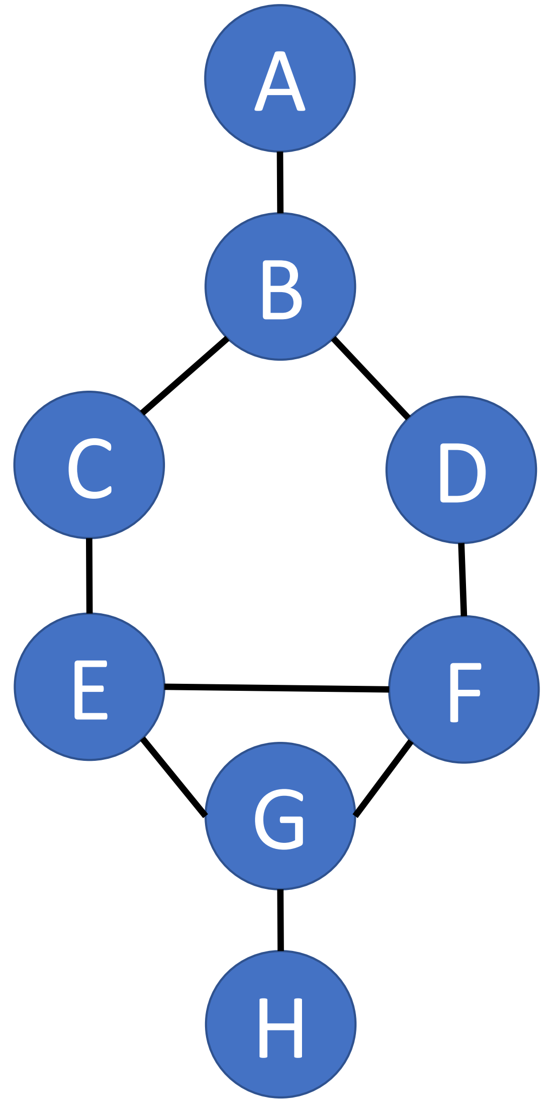


2.a. Moralization

"Moralization" =

1. If two variables have a child together, force them to get married.
2. Get rid of the arrows (not necessary any more).

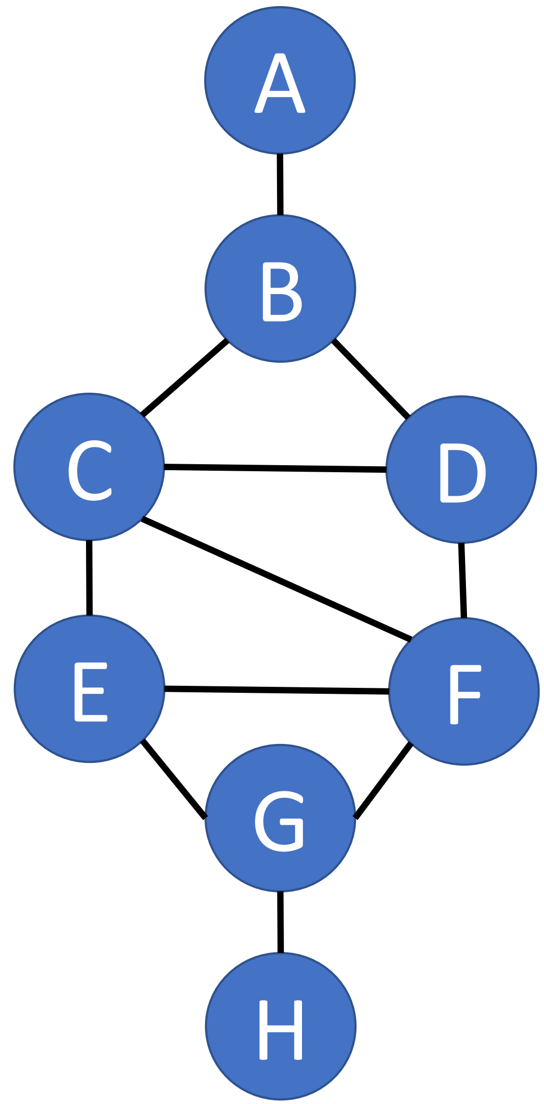
Result: Markov blanket = the set of variables to which a variable is connected.



2.b. Triangulation

Triangulation = draw edges so that there is no unbroken cycle of length > 3.

There are usually many different ways to do this. For example, here's one:

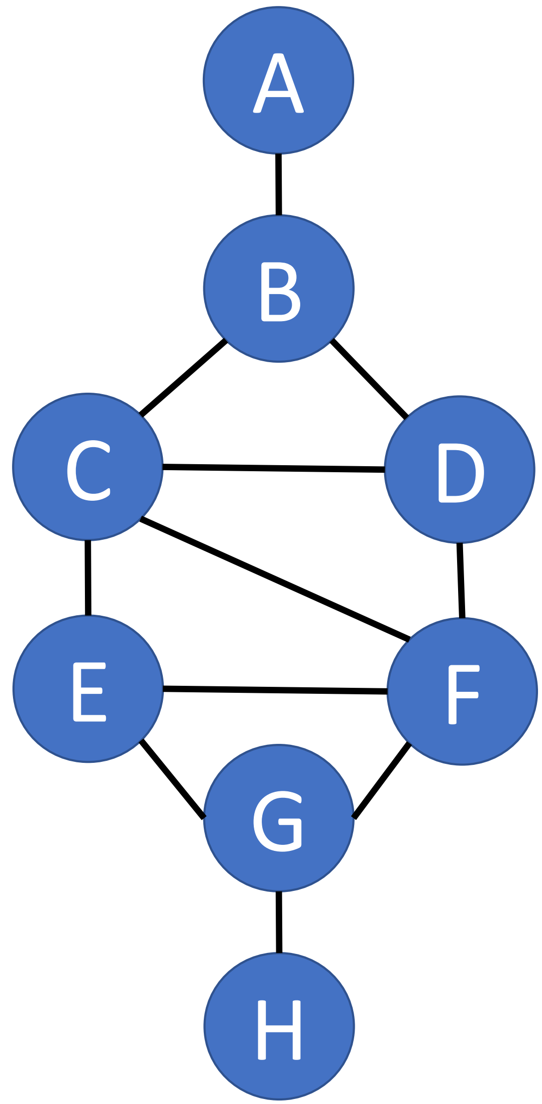


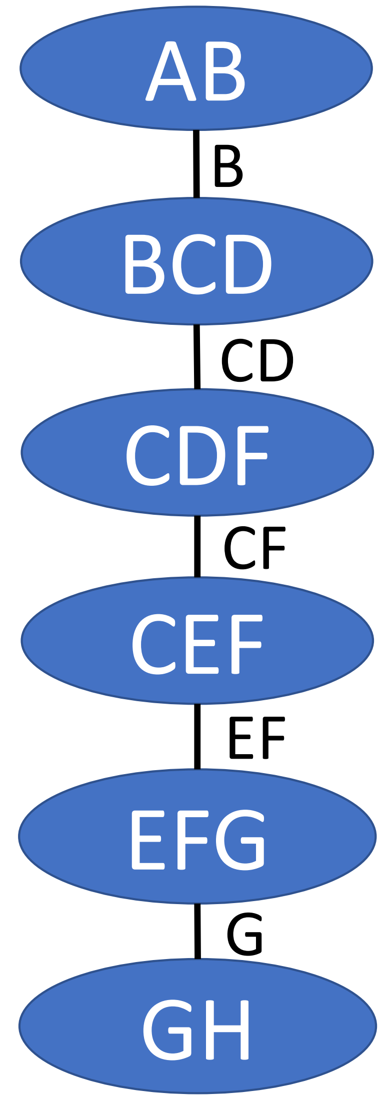
2.c. Form Cliques

Clique = a group of variables, all of whom are members of each other's immediate family.

Junction Tree = a tree in which

- Each node is a clique from the original graph,
- Each edge is an "intersection set," naming the variables that overlap between the two cliques.





2.d. Sum-Product

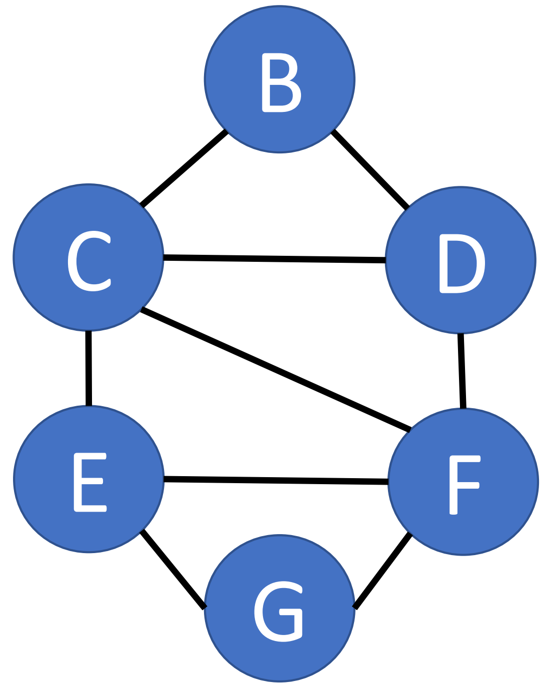
Suppose we need P(B,G):

1. Product: P(B,C,D,F)=P(B)P(C|B)P(D|B)P(F|D)
2. Sum: $P(B,C,F) = \sum_D P(B,C,D,F)$
3. Product: P(B,C,E,F)=P(B,C,F)P(E|C)
4. Sum: $P(B,E,F) = \sum_C P(B,C,E,F)$
5. Product: P(B,E,F,G) = P(B,E,F)P(G|E,F)
6. Sum: $P(B,G) = \sum_E \sum_F P(B,E,F,G)$

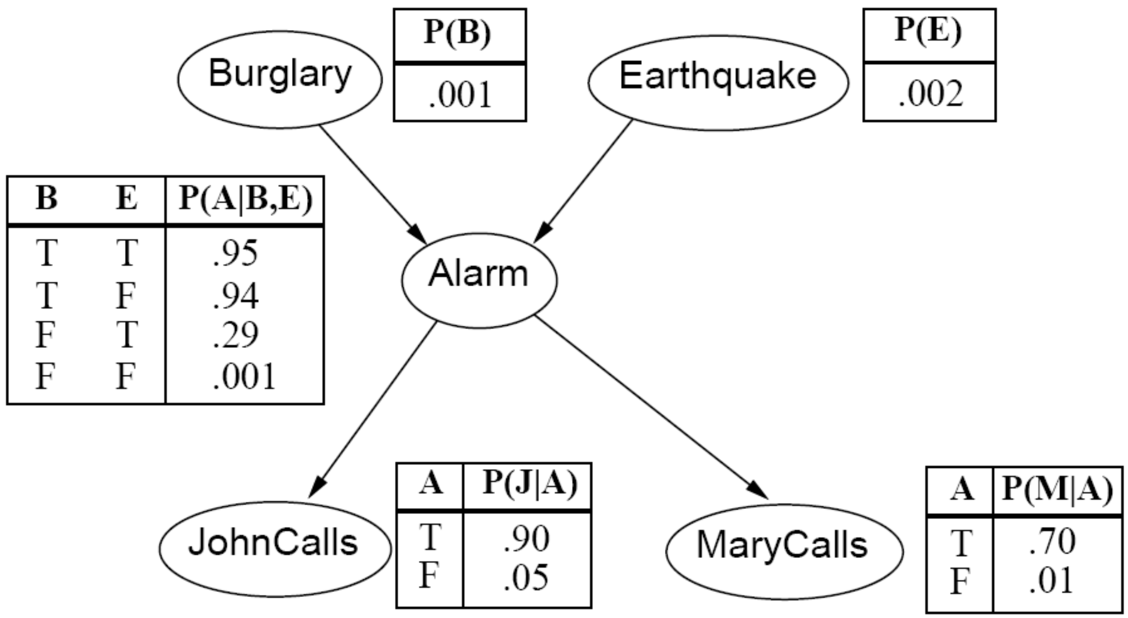
Complexity: $O\{NK^M\}$, where N=# cliques,

K = # values for each variable,

M = 1 + # variables in the largest clique



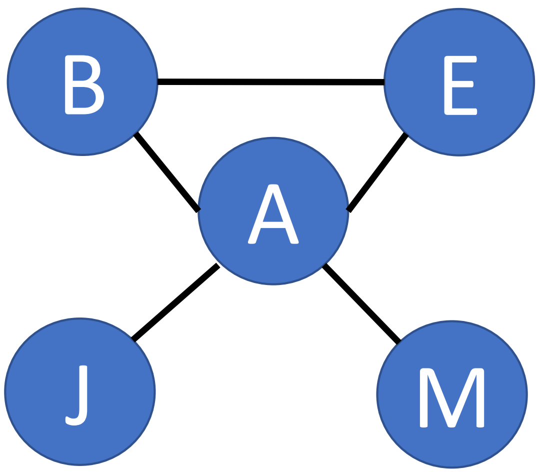
Junction Tree: Sample Test Question



Consider the burglar alarm example.

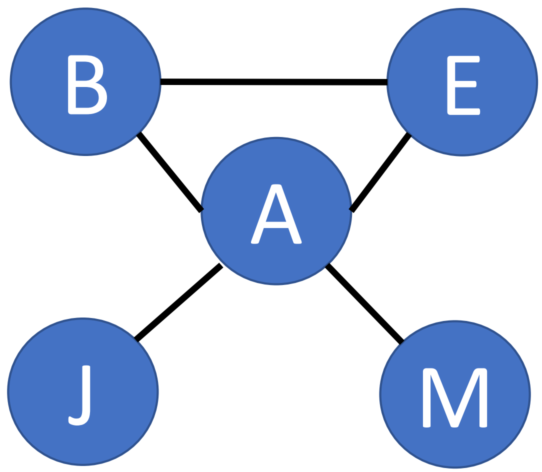
a. Moralize this graph
b. Is it already triangulated? If not, triangulate it.
c. Draw the junction tree

Solution



a. Moralize this graph

Solution

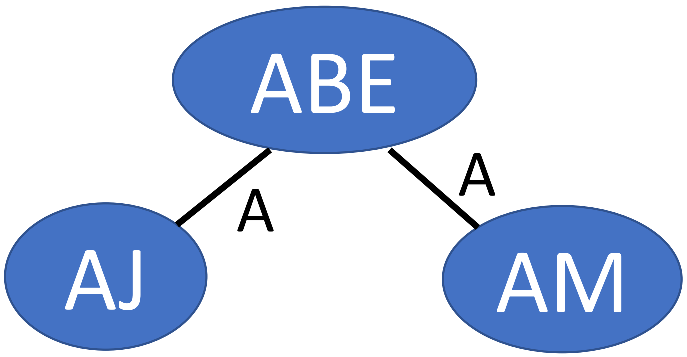


b. Is it already triangulated?

Answer: yes. There is no unbroken cycle of length > 3.

Solution

c. Draw the junction tree



Time Complexity of Bayes Net Inference

- Tree-structured Bayes nets: the sum-product algorithm
  - Quadratic complexity, $O\{NK^3\}$
- Polytrees: the junction tree algorithm
  - Pseudo-polynomial complexity, $O\{NK^M\}$, for M<N
- Arbitrary Bayes nets: #P complete, $O\{K^N\}$
  - The SAT problem is a Bayes net!
- Parameter Learning

Bayesian network inference

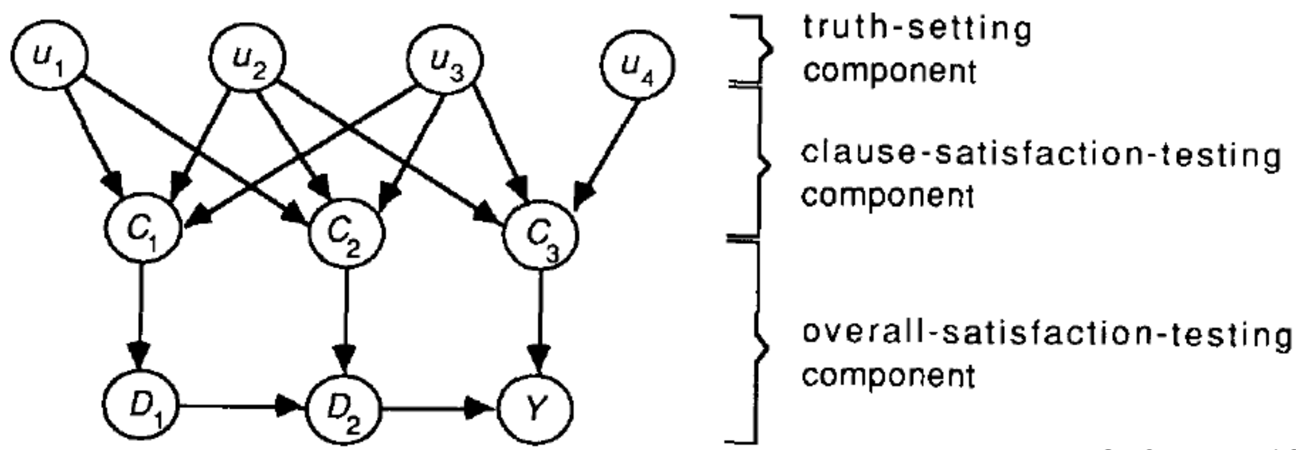
- In full generality, NP-hard
  - More precisely, #P-hard: equivalent to counting satisfying assignments
- We can reduce satisfiability to Bayesian network inference
  - Decision problem: is P(Y) > 0?

$$Y = (U_1 \vee U_2 \vee U_3) \wedge (\neg U_1 \vee \neg U_2 \vee U_3) \wedge (U_2 \vee \neg U_3 \vee U_4)$$

Bayesian network inference

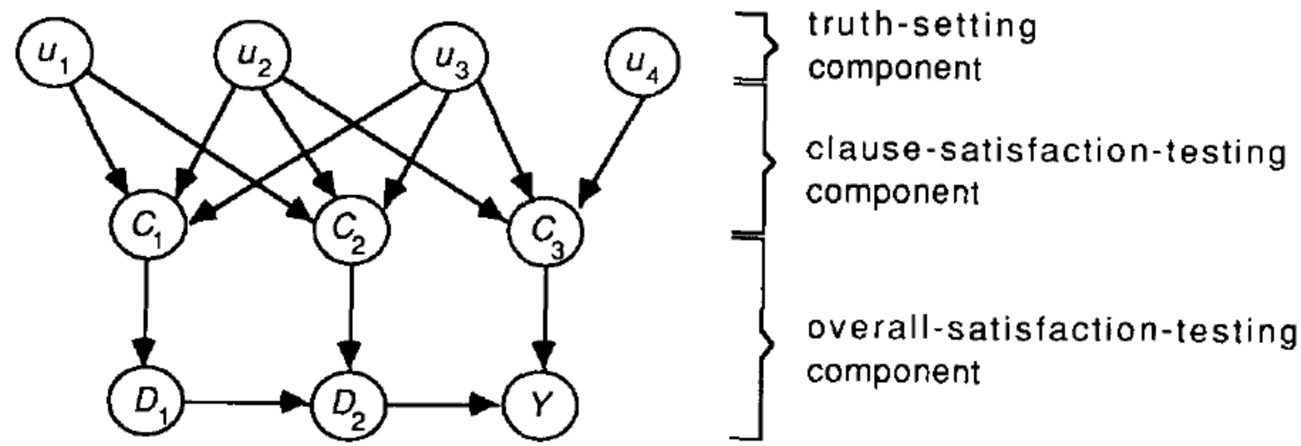
- In full generality, NP-hard
  - More precisely, #P-hard: equivalent to counting satisfying assignments
- We can reduce satisfiability to Bayesian network inference
  - Decision problem: is P(Y) > 0?

$$Y = \underbrace{(U_1 \vee U_2 \vee U_3)}_{C_1} \wedge \underbrace{(\neg U_1 \vee \neg U_2 \vee U_3)}_{C_2} \wedge \underbrace{(U_2 \vee \neg U_3 \vee U_4)}_{C_3}$$



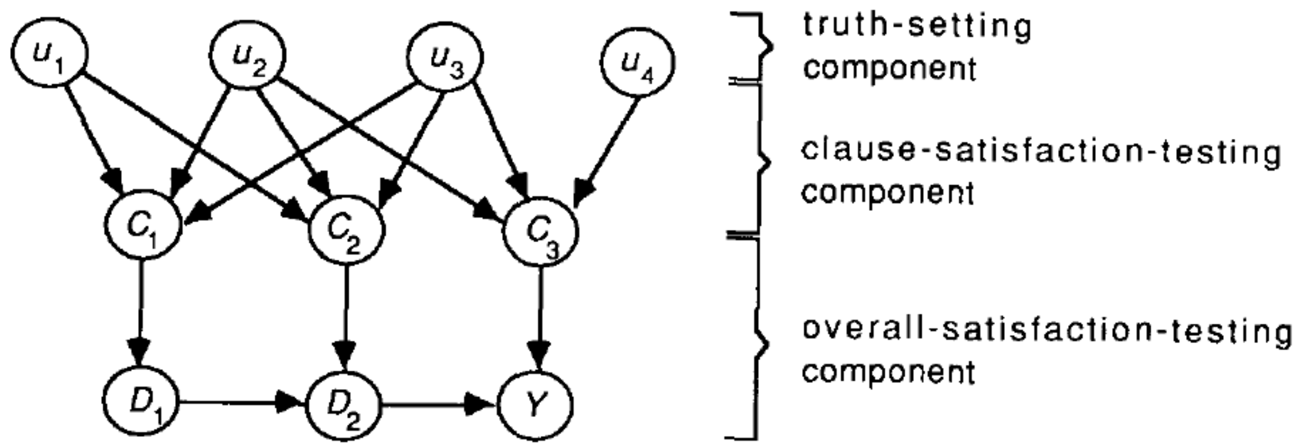
G. Cooper, 1990

Bayesian network inference



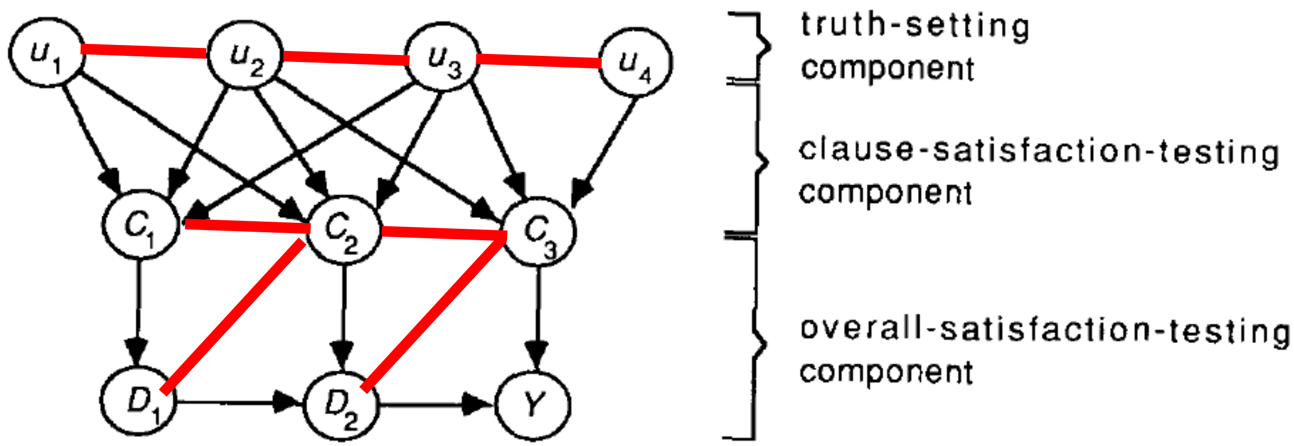
$$P(U_1, U_2, U_3, U_4, C_1, C_2, C_3, D_1, D_2, Y) =$$
$$P(U_1)P(U_2)P(U_3)P(U_4)$$
$$P(C_1 \mid U_1, U_2, U_3)P(C_2 \mid U_1, U_2, U_3)P(C_3 \mid U_2, U_3, U_4)$$
$$P(D_1 \mid C_1)P(D_2 \mid D_1, C_2)P(Y \mid D_2, C_3)$$

Bayesian network inference



Why can't we use the junction tree algorithm to efficiently compute Pr(Y)?

Bayesian network inference



Why can't we use the junction tree algorithm to efficiently compute Pr(Y)?
Answer: after we moralize and triangulate, the size of the largest clique (u2u3c1c2c3) is $M \approx N$, same order of magnitude as the original problem

Time Complexity of Bayes Net Inference

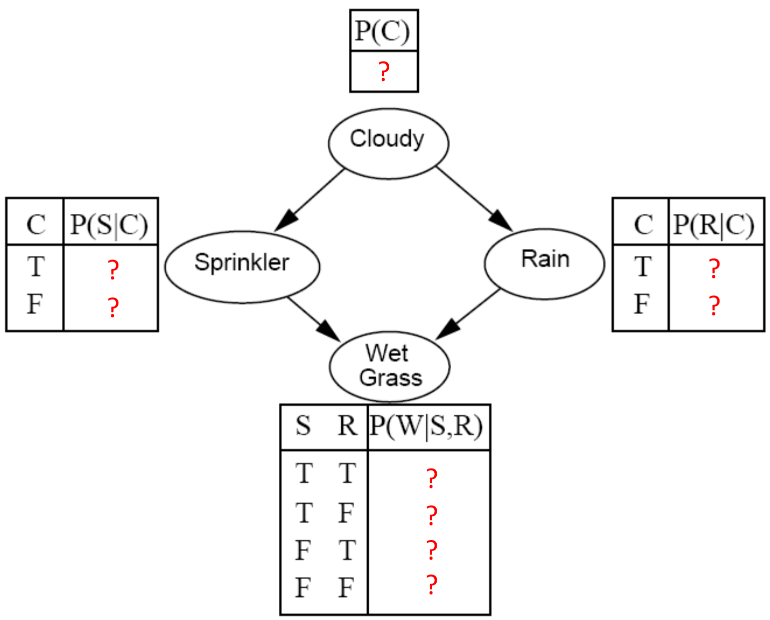
- Tree-structured Bayes nets: the sum-product algorithm
  - Quadratic complexity, $O\{NK^3\}$
- Polytrees: the junction tree algorithm
  - Pseudo-polynomial complexity, $O\{NK^M\}$, for M<N
- Arbitrary Bayes nets: #P complete, $O\{K^N\}$
  - The SAT problem is a Bayes net!
- Parameter Learning

Parameter learning

- **Inference problem**: given values of evidence variables $\mathbf{E} = \mathbf{e}$, answer questions about query *variables* $\mathbf{X}$ using the posterior $P(\mathbf{X} \mid \mathbf{E} = \mathbf{e})$
- **Learning problem:** estimate the parameters of the probabilistic model $P(\mathbf{X} \mid \mathbf{E})$ given a *training sample* $\{(\mathbf{x}_1,\mathbf{e}_1), \ldots, (\mathbf{x}_n,\mathbf{e}_n)\}$

Parameter learning: complete data

- Suppose we know the network structure (but not the parameters), and have a training set of *complete* observations



Training set

Sample	C	S	R	W
1	T	F	T	T
2	F	T	F	T
3	T	F	F	F
4	T	T	T	T
5	F	T	F	T
6	T	F	T	F
…	…	…	….	…

Parameter learning

- Suppose we know the network structure (but not the parameters), and have a training set of *complete* observations
- Example:

$$P(S = T | C = T) = \frac{\#\text{samples with } S = T, C = T}{\#\text{ samples with } C = T} = \frac{1}{4}$$

Training set

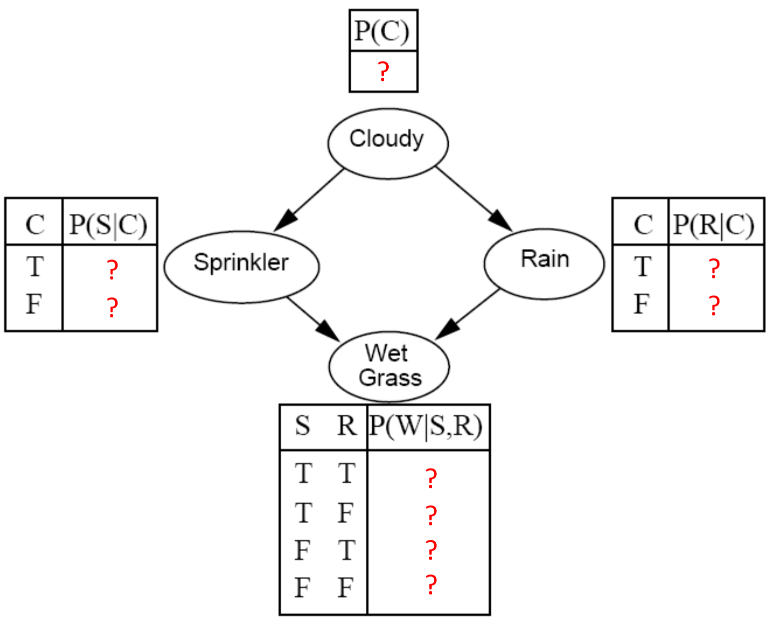
Sample	**C**	**S**	**R**	**W**
1	T	F	T	T
2	F	T	F	T
3	T	F	F	F
4	T	T	T	T
5	F	T	F	T
6	T	F	T	F
…	…	…	….	…

Parameter learning

- Suppose we know the network structure (but not the parameters), and have a training set of *complete* observations
  - P(X | Parents(X)) is given by the observed frequencies of the different values of X for each combination of parent values

Parameter learning: missing data

- Suppose we know the network structure (but not the parameters), and have a training set, but the training set is ***missing some observations***.

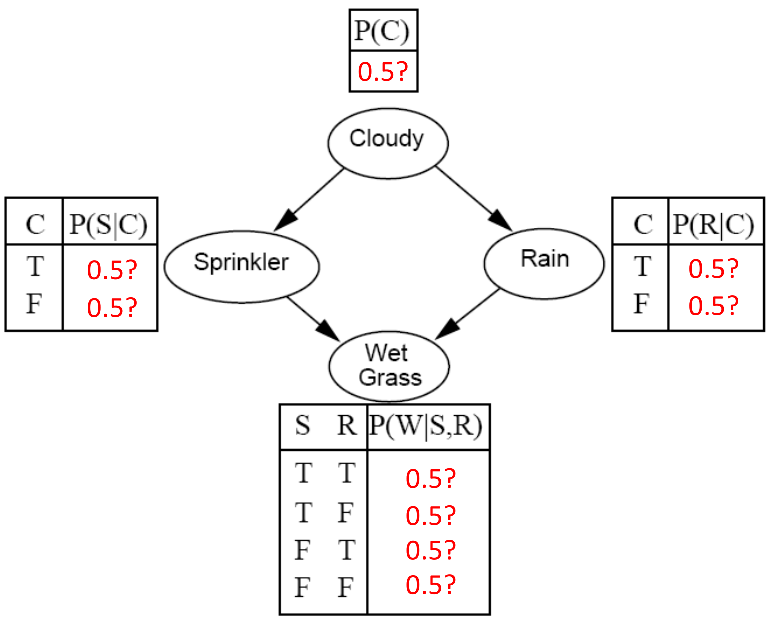


Training set

Sample	C	S	R	W
1	?	F	T	T
2	?	T	F	T
3	?	F	F	F
4	?	T	T	T
5	?	T	F	T
6	?	F	T	F
…	…	…	….	…

Missing data: the EM algorithm

- The EM algorithm starts ("Expectation Maximization") starts with an initial guess for each parameter value.
- We try to improve the initial guess, using the algorithm on the next two slides:
  - E-step
  - M-step



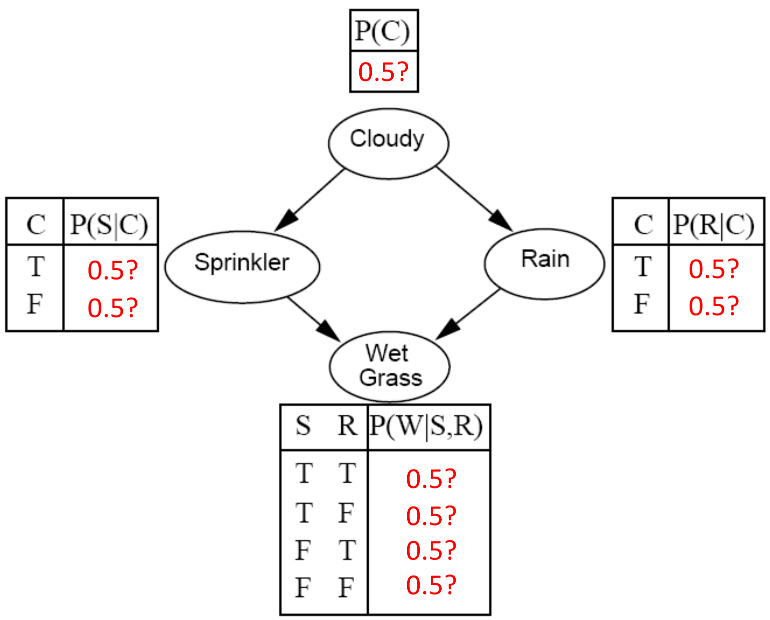
Training set

Sample	C	S	R	W
1	?	F	T	T
2	?	T	F	T
3	?	F	F	F
4	?	T	T	T
5	?	T	F	T
6	?	F	T	F
…	…	…	….	…

Missing data: the EM algorithm

- E-Step (Expectation): Given the model parameters, replace each of the missing numbers with a probability (a number between 0 and 1) using

$$P(C = 1|S, R, W) = \frac{P(C = 1, S, R, W)}{P(C = 1, S, R, W) + P(C = 0, S, R, W)}$$



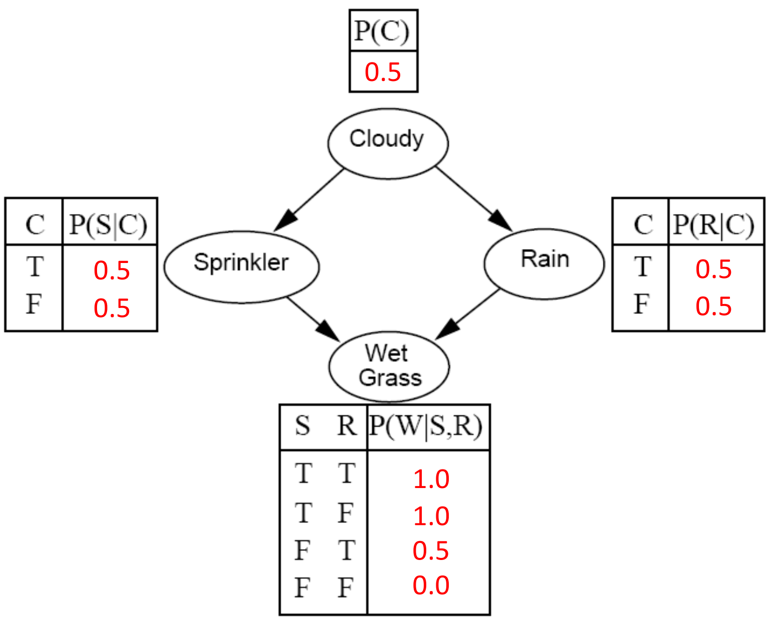
Training set

Sample	C	S	R	W
1	0.5?	F	T	T
2	0.5?	T	F	T
3	0.5?	F	F	F
4	0.5?	T	T	T
5	0.5?	T	F	T
6	0.5?	F	T	F
…	…	…	….	…

Missing data: the EM algorithm

- M-Step (Maximization): Given the missing data estimates, replace each of the missing model parameters using

$$P(\text{Variable} = \text{T}|\text{Parents} = \text{value}) = \frac{E[\#\text{ times Variable} = T, \text{Parents} = \text{value}]}{E[\#\text{times Parents} = \text{value}]}$$

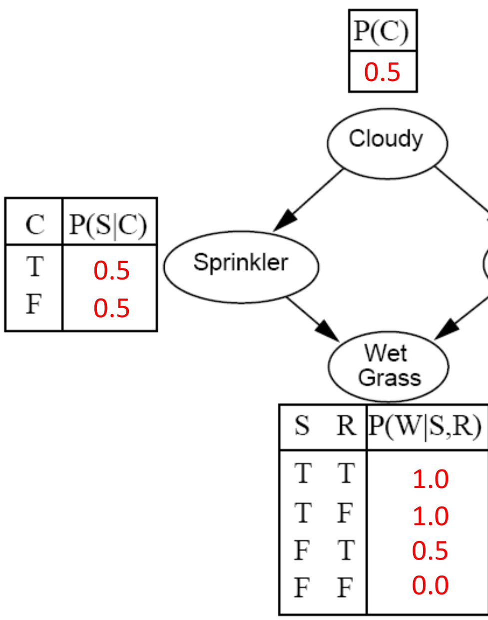


Training set

Sample	C	S	R	W
1	0.5?	F	T	T
2	0.5?	T	F	T
3	0.5?	F	F	F
4	0.5?	T	T	T
5	0.5?	T	F	T
6	0.5?	F	T	F
…	…	…	….	…

Missing data: the EM algorithm

- Iterate back and forth between E-step and M-step until the model converges.



Training set

Sample	C	S	R	W
1	0.5?	F	T	T
2	0.5?	T	F	T
3	0.5?	F	F	F
4	0.5?	T	T	T
5	0.5?	T	F	T
6	0.5?	F	T	F
…	…	…	….	…

Summary: Bayesian networks

- Structure
- Parameters
- Inference
- Learning