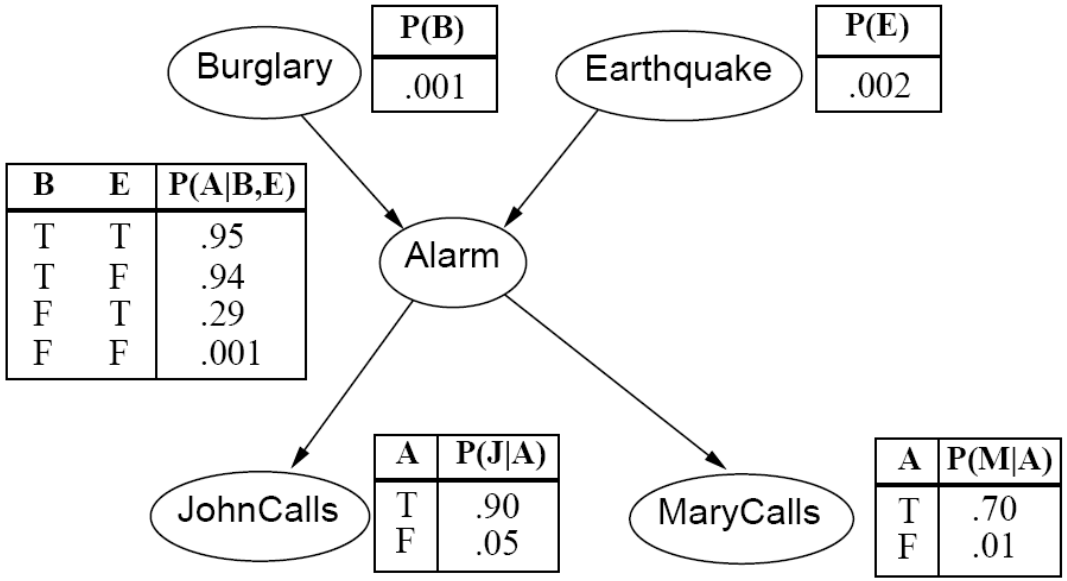
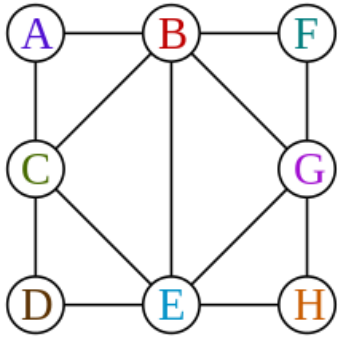
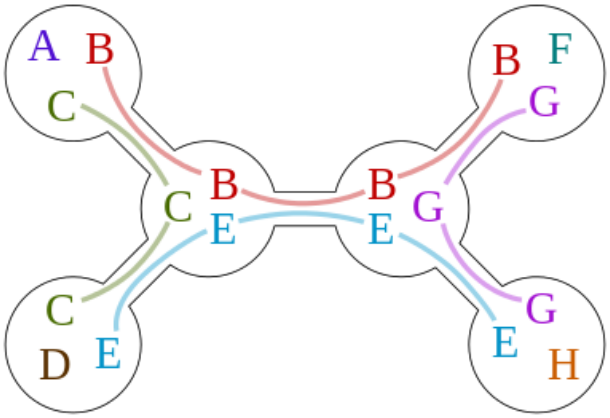
CS 440/ECE448 Lecture 19: Bayes Net Inference

Mark Hasegawa-Johnson, 3/2019 modified by Julia Hockenmaier 3/2019

Including slides by Svetlana Lazebnik, 11/2016



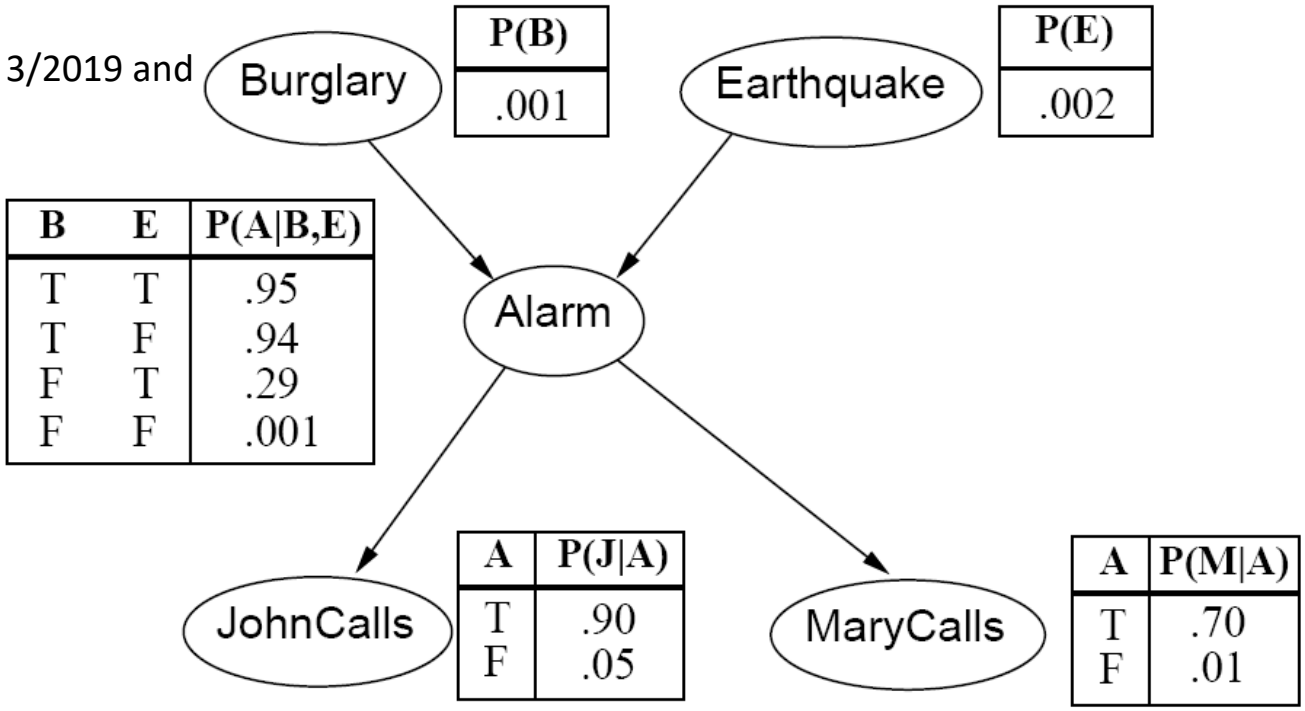




CS440/ECE448 Lecture 19: Bayesian Networks and Bayes Net Inference

Slides by Svetlana Lazebnik, 10/2016

Modified by Mark Hasegawa-Johnson, 3/2019 and Julia Hockenmaier 3/2019



Today's lecture

- **Bayesian Networks** (Bayes Nets)
  - A graphical representation of probabilistic models
  - Capture conditional (in)dependencies between random variables
- **Inference and Learning in Bayes Nets**
  - Inference = Reasoning
  - Learning = Parameter estimation

Review: Bayesian inference

A general scenario:

Query *variables:* X

***Evidence* (*observed*) variables and their values**: **E = e**

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

This can be done using the posterior distribution P(**X** | **E** = **e**)

Example of a useful question: **Which X is true?**

More formally: what value of **X** has the least probability of being wrong?

Answer: **MPE = MAP** (argmin P(error) = argmax P(X=x|E=e))

Today: What if P(X,E) is complicated?

- Very, very common problem: P(X,E) is complicated because both X and E depend on some hidden variable Y
- SOLUTION:
  - Represent the dependencies as a graph
  - When your algorithm performs inference, make sure it does so in the order of the graph
- FORMALISM: Bayesian Network

Bayesian Inference with Hidden Variables

- **A general scenario:**
  - **Query *variables:* X**
  - ***Evidence* (*observed*) variables and their values**: **E = e**
  - ***Hidden (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**)
  - In turn, the posterior needs to be derived from the full joint P(**X**, **E**, **Y**)

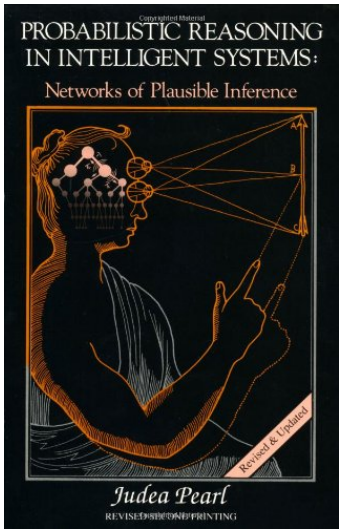
$$P(\boldsymbol{X} \mid \boldsymbol{E} = \boldsymbol{e}) = \frac{P(\boldsymbol{X}, \boldsymbol{e})}{P(\boldsymbol{e})} \propto \sum_{\boldsymbol{y}} P(\boldsymbol{X}, \boldsymbol{e}, \boldsymbol{y})$$

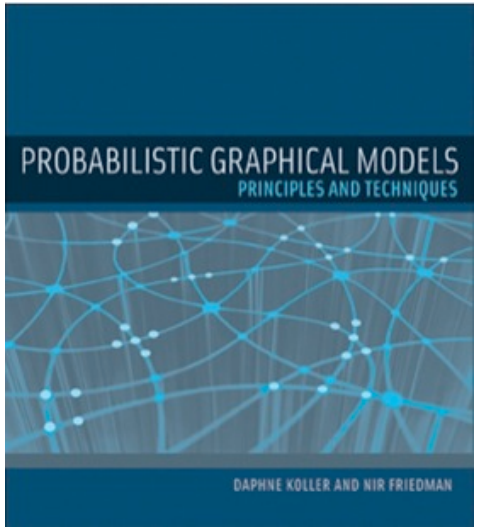
- Bayesian networks are a tool for representing joint probability distributions efficiently

Bayesian Networks

Bayesian networks

- More commonly called *graphical models*
- A way to depict conditional independence relationships between random variables
- A compact specification of full joint distributions





Independence

- Random variables X and Y are **independent** $(X \perp Y)$ if $P(X,Y) = P(X) \times P(Y)$

  NB.: Since X and Y are R.V.s (not individual events), $P(X,Y) = P(X) \times P(Y)$ is an abbreviation for: $\forall x \forall y\ P(X=x,Y=y) = P(X=x) \times P(Y=y)$

- X and Y are ***conditionally*** **independent** given Z $(X \perp Y \mid Z)$ if $P(X,Y \mid Z) = P(X \mid Z) \times P(Y \mid Z)$

  The value of X depends on the value of Z, and the value of Y depends on the value of Z, so X and Y are not independent.

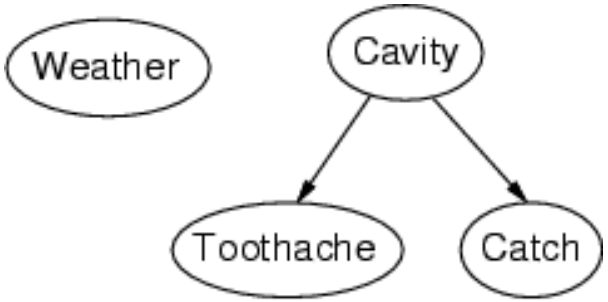
Bayesian networks

- **Insight:** (Conditional) independence assumptions are essential for probabilistic modeling

- **Bayes Net:** a directed graph which represents the joint distribution of a number of random variables in a directed graph
  - Nodes = random variables
  - Directed edges = dependencies

Bayesian networks

- **Nodes:** random variables
- **Edges:** dependencies
  - An edge from one variable (parent) to another (child) indicates direct influence (conditional probabilities)
  - Edges must form a directed, *acyclic* graph
  - Each node is conditioned on its parents: P(X | Parents(X))
    These conditional distributions are the parameters of the network
- **Each node is conditionally independent of its non-descendants given its parent**



We have four random variables
Weather is independent of cavity, toothache and catch
Toothache and catch both depend on cavity.

Conditional independence and the joint distribution

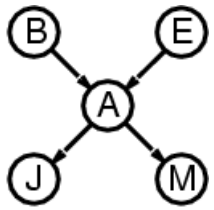
- **Key property: each node is conditionally independent of its *non-descendants* given its *parents***
- Suppose the nodes $X_1, \ldots, X_n$ are sorted in topological order of the graph (i.e. if X_i is a parent of X_j, $i < j$)
- To get the joint distribution $P(X_1, \ldots, X_n)$, use chain rule (step 1 below) and then take advantage of independencies (step 2)

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P\left(X_i \mid X_1,\ldots,X_{i-1}\right)$$

$$= \prod_{i=1}^{n} P\left(X_i \mid Parents(X_i)\right)$$

The joint probability distribution

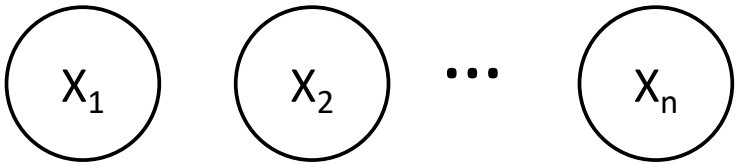
$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P\left(X_i \mid Parents(X_i)\right)$$



P(j, m, a, ¬b,¬e) = P(¬b) P(¬e) P(a|¬b,¬e) P(j|a) P(m|a)

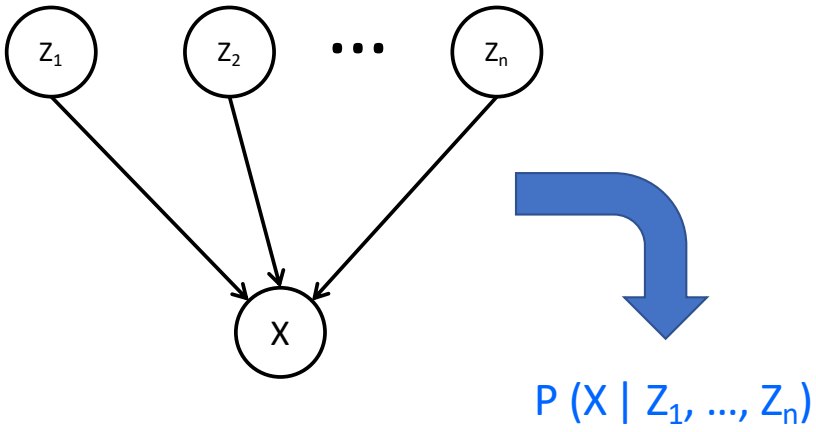
Example: N independent coin flips

- Complete independence: no interactions:
  P(X1) P(X2) P(X3)



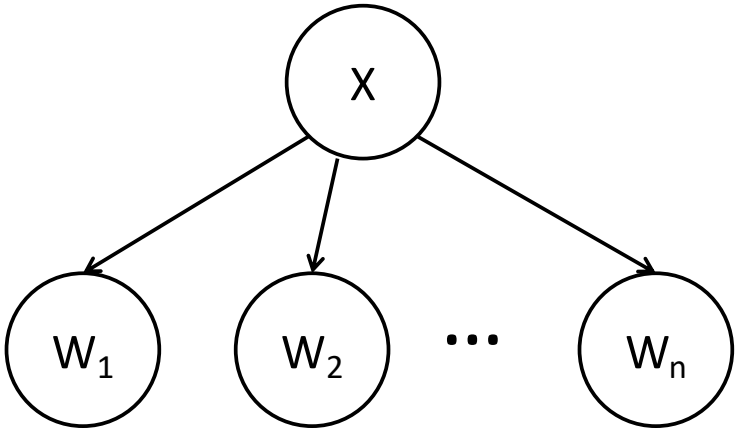
Conditional probability distributions

- To specify the full joint distribution, we need to specify a *conditional* distribution for each node given its parents: P (X | Parents(X))



Naïve Bayes document model

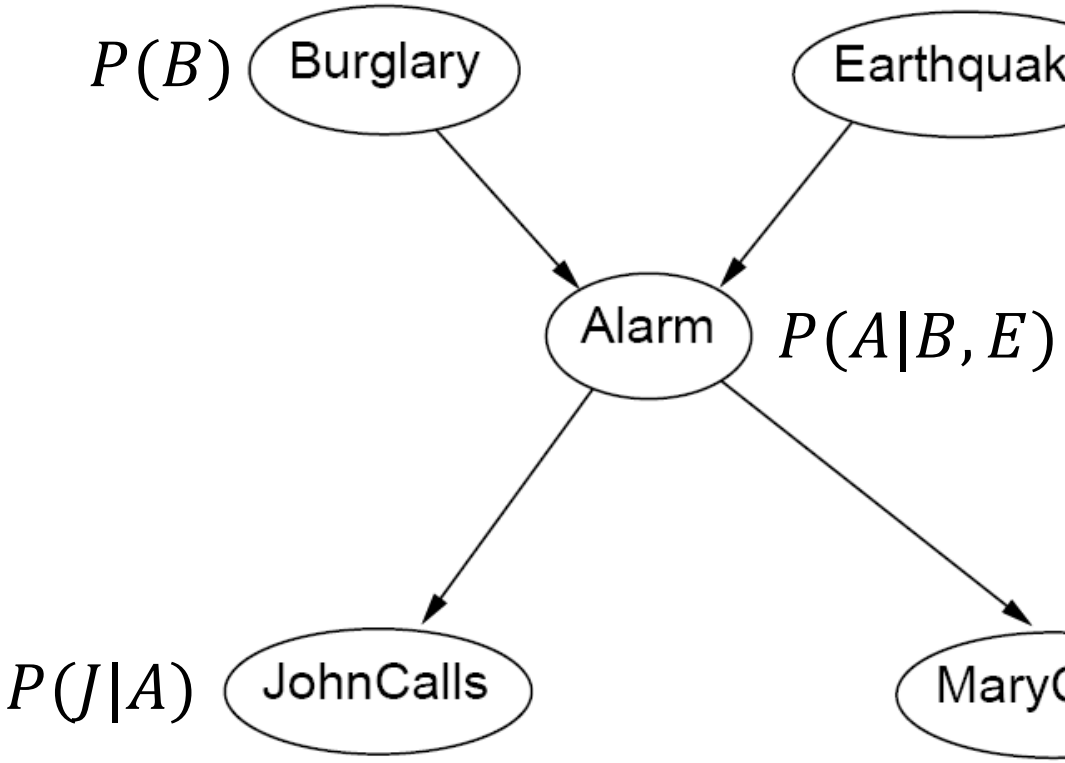
- Random variables:
  - X: document class
  - W_1, …, W_n: words in the document
- Dependencies: $P(X)\ P(W_1 \mid X) \ldots P(W_n \mid X)$



Example: Los Angeles Burglar Alarm

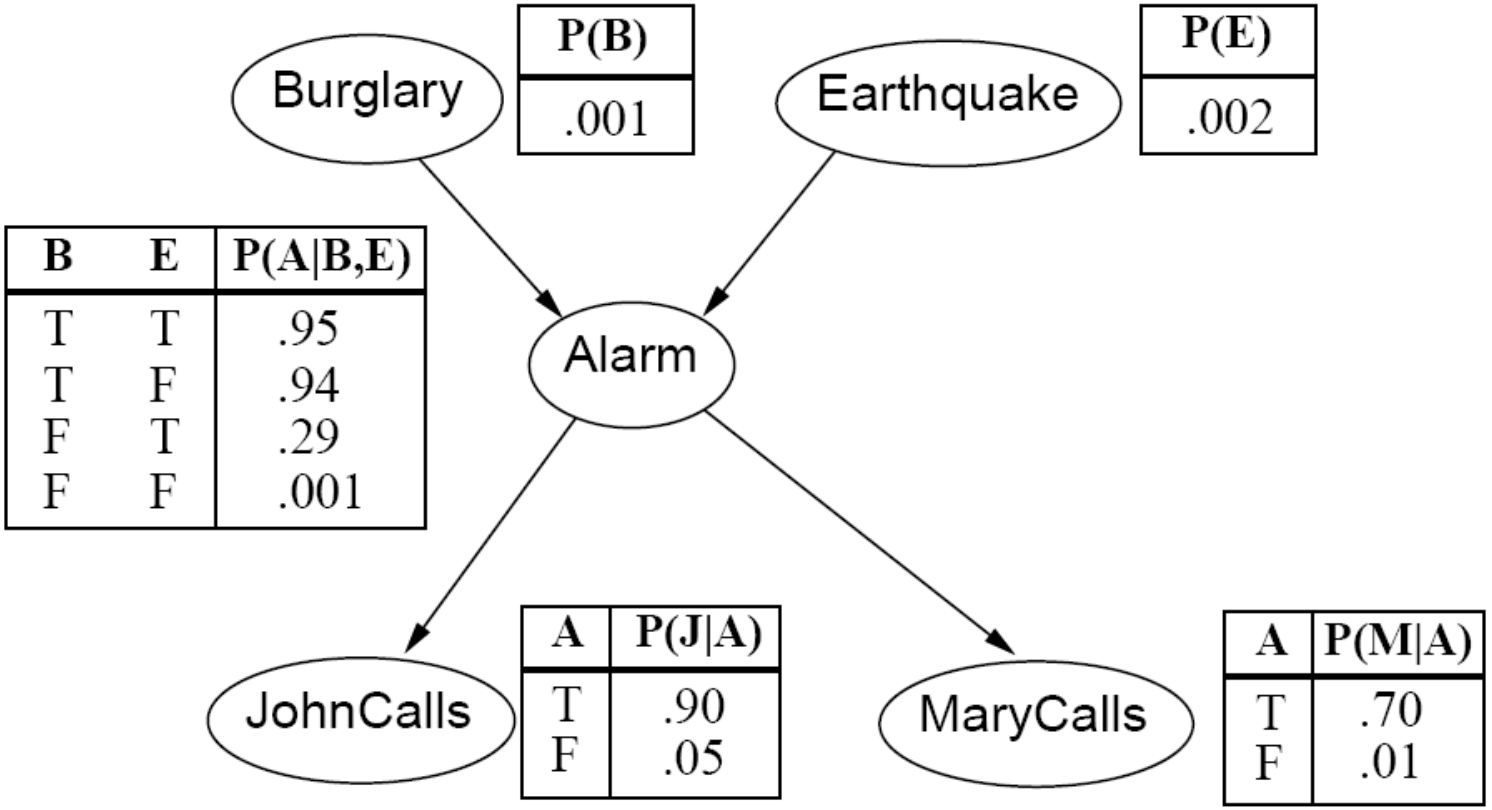
- I have a burglar alarm that is sometimes set off by minor earthquakes. My two neighbors, John and Mary, promised to call me at work if they hear the alarm
  - Example inference task: suppose Mary calls and John doesn't call. What is the probability of a burglary?
- What are the random variables?
  - *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*
- What are the direct influence relationships?
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call

Example: Burglar Alarm



- A "model" is a complete specification of the dependencies.
- The conditional probability tables are the *model parameters.*

Example: Burglar Alarm

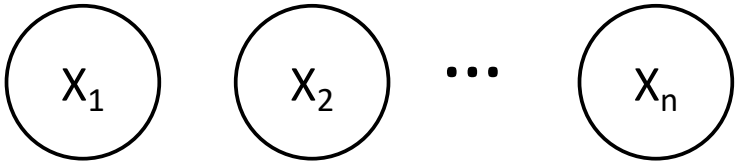


Outline

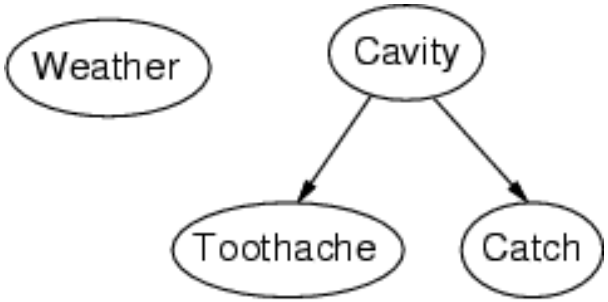
- Review: Bayesian inference
- Bayesian network: graph semantics
- The Los Angeles burglar alarm example
- Conditional independence ≠ Independence
- Constructing a Bayesian network: Structure learning
- Constructing a Bayesian network: Hire an expert

Independence

- By saying that X_i and X_j are independent, we mean that
$$\mathrm{P}(X_j, X_i) = \mathrm{P}(X_i)\mathrm{P}(X_j)$$
- X_i and X_j are independent if and only if they have no common ancestors
- Example: *independent coin flips*



- Another example: Weather is independent of all other variables in this model.

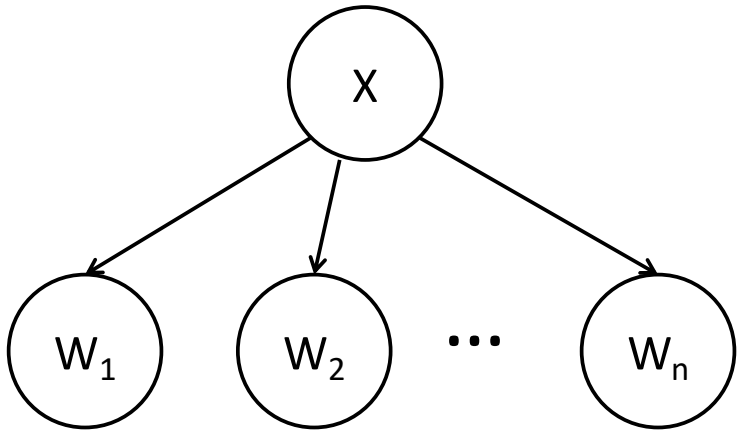


Conditional independence

- By saying that W_i and W_j are conditionally independent given X, we mean that

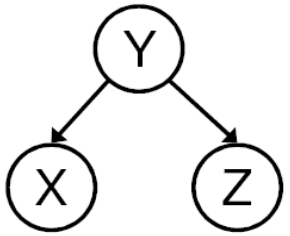
$$\mathrm{P}(W_i, W_j | X) = \mathrm{P}(W_i | X)\mathrm{P}(W_j | X)$$

- W_i and W_j are conditionally independent given X if and only if they have no common ancestors other than the ancestors of X.
- Example: *naïve Bayes model:*



Conditional independence ≠ Independence

Common cause: Conditionally Independent



Y: Project due

X: Newsgroup busy

Z: Lab full

Are X and Z **independent**? **No**

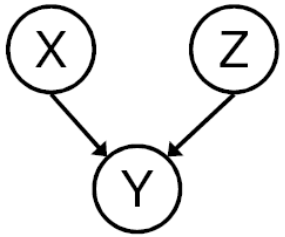
$$P(Z,X) = \sum_Y P(Z|Y)P(X|Y)P(Y)$$

$$P(Z)P(X) = \left(\sum_Y P(Z|Y)P(Y)\right)\left(\sum_Y P(X|Y)P(Y)\right)$$

Are they **conditionally independent given** Y? **Yes**

$$P(Z,X|Y) = P(Z|Y)P(X|Y)$$

Common effect: Independent



X: Raining

Z: Ballgame

Y: Traffic

Are X and Z **independent**? **Yes**

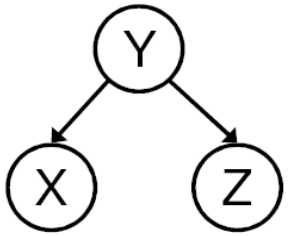
$$P(X,Z) = P(X)P(Z)$$

Are they **conditionally independent given** Y? **No**

$$P(Z,X|Y) = \frac{P(Y|X,Z)P(X)P(Z)}{P(Y)} \neq P(Z|Y)P(X|Y)$$

Conditional independence ≠ Independence

Common cause: Conditionally Independent



Y: Project due

X: Newsgroup busy

Z: Lab full

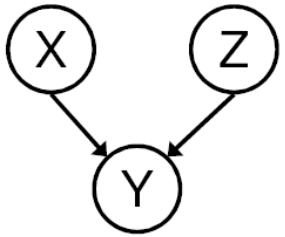
Are X and Z independent? **No**

Knowing X tells you about Y, which tells you about Z.

Are they conditionally independent given Y? **Yes**

If you already know Y, then X gives you no useful information about Z.

Common effect: Independent



X: Raining

Z: Ballgame

Y: Traffic

Are X and Z independent? **Yes**

Knowing X tells you nothing about Z.

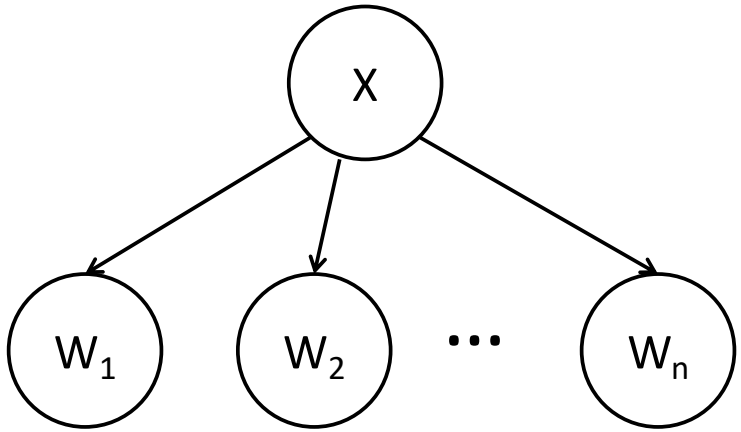
Are they conditionally independent given Y? **No**

If Y is true, then either X or Z must be true.

Knowing that X is false means Z must be true.

We say that X "explains away" Z.

Conditional independence ≠ Independence

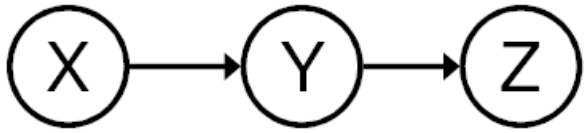


Being conditionally independent given X does NOT mean that W_i and W_j are independent. Quite the opposite. For example:

- The document topic, X, can be either "sports" or "pets", equally probable.
- W_1=1 if the document contains the word "food," otherwise W_1=0.
- W_2=1 if the document contains the word "dog," otherwise W_2=0.
- Suppose you don't know X, but you know that W_2=1 (the document has the word "dog"). Does that change your estimate of p(W_1=1)?

Conditional independence

Another example: *causal chain*



X: Low pressure

Y: Rain

Z: Traffic

- X and Z are conditionally independent given Y, because they have no common ancestors other than the ancestors of Y.

- Being conditionally independent given Y does NOT mean that X and Z are independent. Quite the opposite. For example, suppose $\mathrm{P}(X) = 0.5$, $\mathrm{P}(Y|X) = 0.8$, $\mathrm{P}(Y|\neg X) = 0.1$, $\mathrm{P}(Z|Y) = 0.7$, and $\mathrm{P}(Z|\neg Y) = 0.4$. Then we can calculate that $\mathrm{P}(Z|X) = 0.64$, but $\mathrm{P}(Z) = 0.535$

Outline

- Review: Bayesian inference
- Bayesian network: graph semantics
- The Los Angeles burglar alarm example
- Conditional independence ≠ Independence
- Constructing a Bayesian network: Structure learning
- Constructing a Bayesian network: Hire an expert

Constructing a Bayes Network: Two Methods

1. "Structure Learning" a.k.a. "Analysis of Causality:"
    1. Suppose you know the variables, but you don't know which variables depend on which others. You can learn this from data.
    2. This is an exciting new area of research in statistics, where it goes by the name of "analysis of causality."
    3. … but it's almost always harder than method #2. You should know how to do this in very simple examples (like the Los Angeles burglar alarm), but you don't need to know how to do this in the general case.
2. "Hire an Expert:"
    1. Find somebody who knows how to solve the problem.
    2. Get her to tell you what are the important variables, and which variables depend on which others.
    3. THIS IS ALMOST ALWAYS THE BEST WAY.

Constructing Bayesian networks: Structure Learning

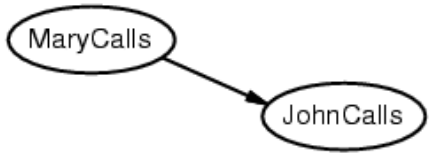
1. Choose an ordering of variables X_1, … , X_n
2. For i = 1 to n
   - add X_i to the network
   - Check your training data. If there is any variable X_1, … ,X_{i-1} that CHANGES the probability of X_i=1, then add that variable to the set Parents(X_i) such that
     $P(X_i \mid Parents(X_i)) = P(X_i \mid X_1, \ldots X_{i-1})$
3. Repeat the above steps for every possible ordering (complexity: n!).
4. Choose the graph that has the smallest number of edges.

Example

- Suppose we choose the ordering M, J, A, B, E

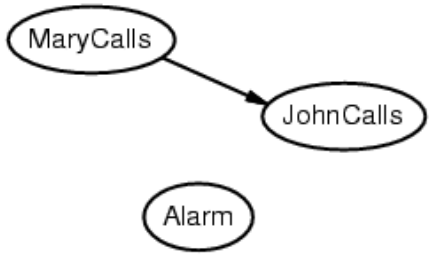
Example

- Suppose we choose the ordering M, J, A, B, E



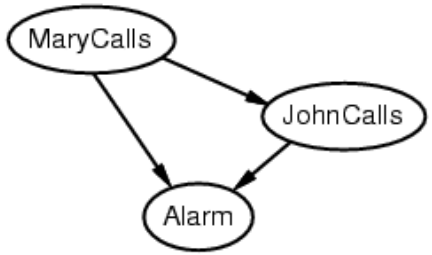
Example

- Suppose we choose the ordering M, J, A, B, E



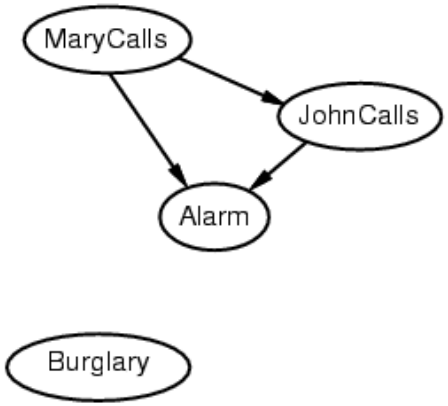
Example

- Suppose we choose the ordering M, J, A, B, E



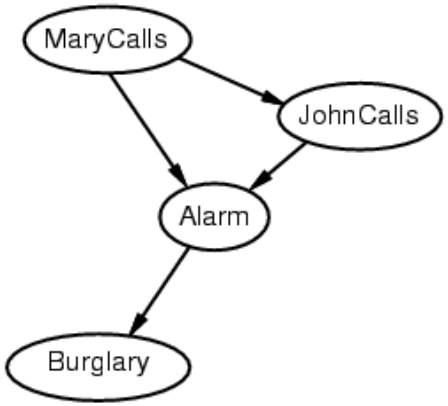
Example

- Suppose we choose the ordering M, J, A, B, E



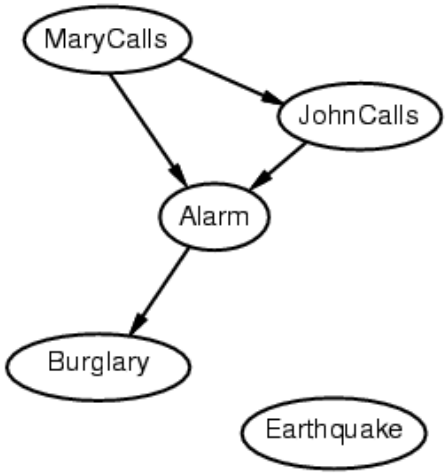
Example

- Suppose we choose the ordering M, J, A, B, E



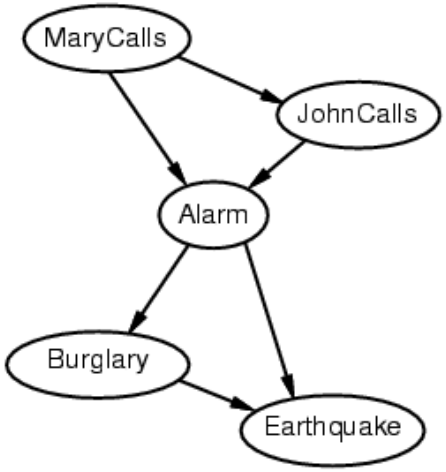
Example

- Suppose we choose the ordering M, J, A, B, E

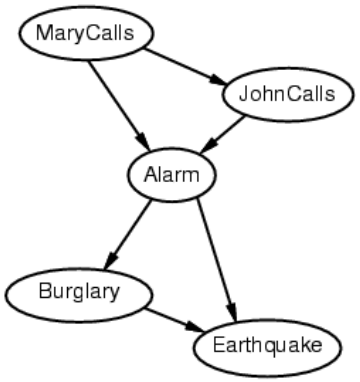


Example

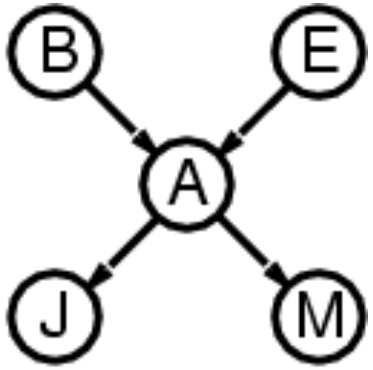
- Suppose we choose the ordering M, J, A, B, E



Example contd.



versus

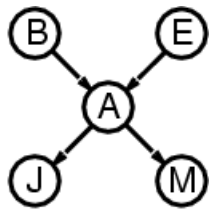


- Deciding conditional independence is hard in noncausal directions
  - The causal direction seems much more natural
- Network is less compact: 1 + 2 + 4 + 2 + 4 = 13 numbers needed (vs. 1+1+4+2+2=10 for the causal ordering)

Why store it in causal order? A: Saves memory

- Suppose we have a Boolean variable X_i with k Boolean parents. How many rows does its conditional probability table have?
  - 2^k rows for all the combinations of parent values
  - Each row requires one number for $P(X_i = \text{true} \mid \text{parent values})$
- If each variable has no more than k parents, how many numbers does the complete network require?
  - $O(n \cdot 2^k)$ numbers – vs. $O(2^n)$ for the full joint distribution
- How many nodes for the burglary network?

  1 + 1 + 4 + 2 + 2 = 10 numbers (vs. 2^5-1 = 31)

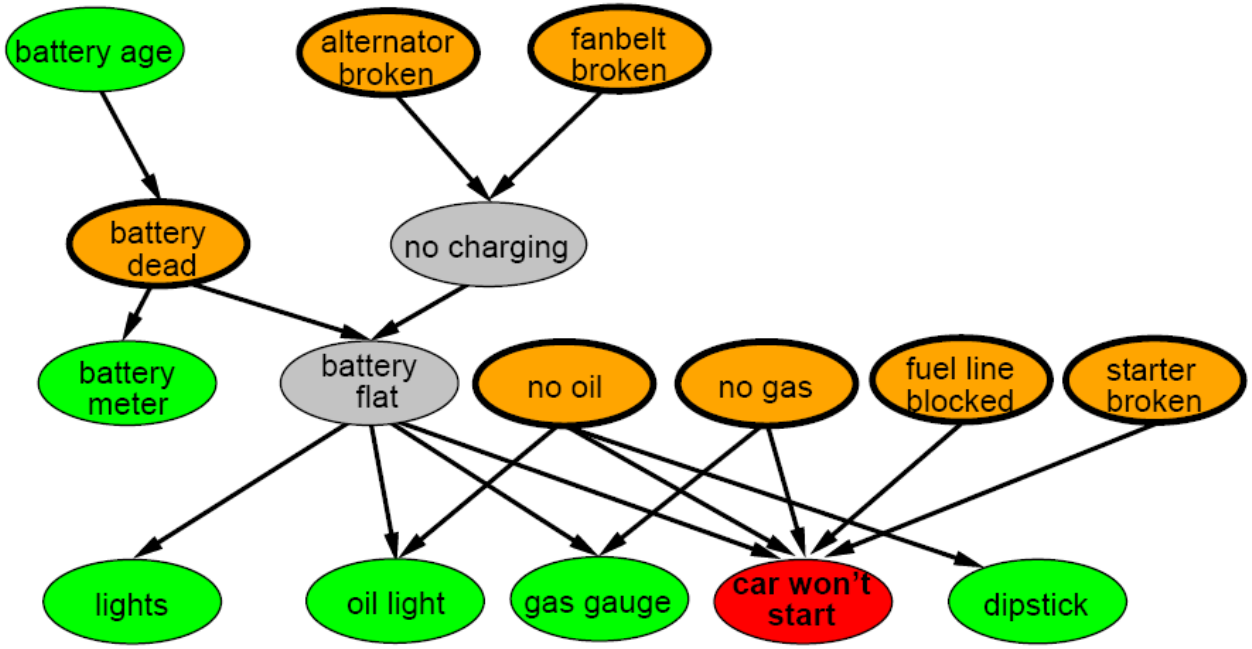


Outline

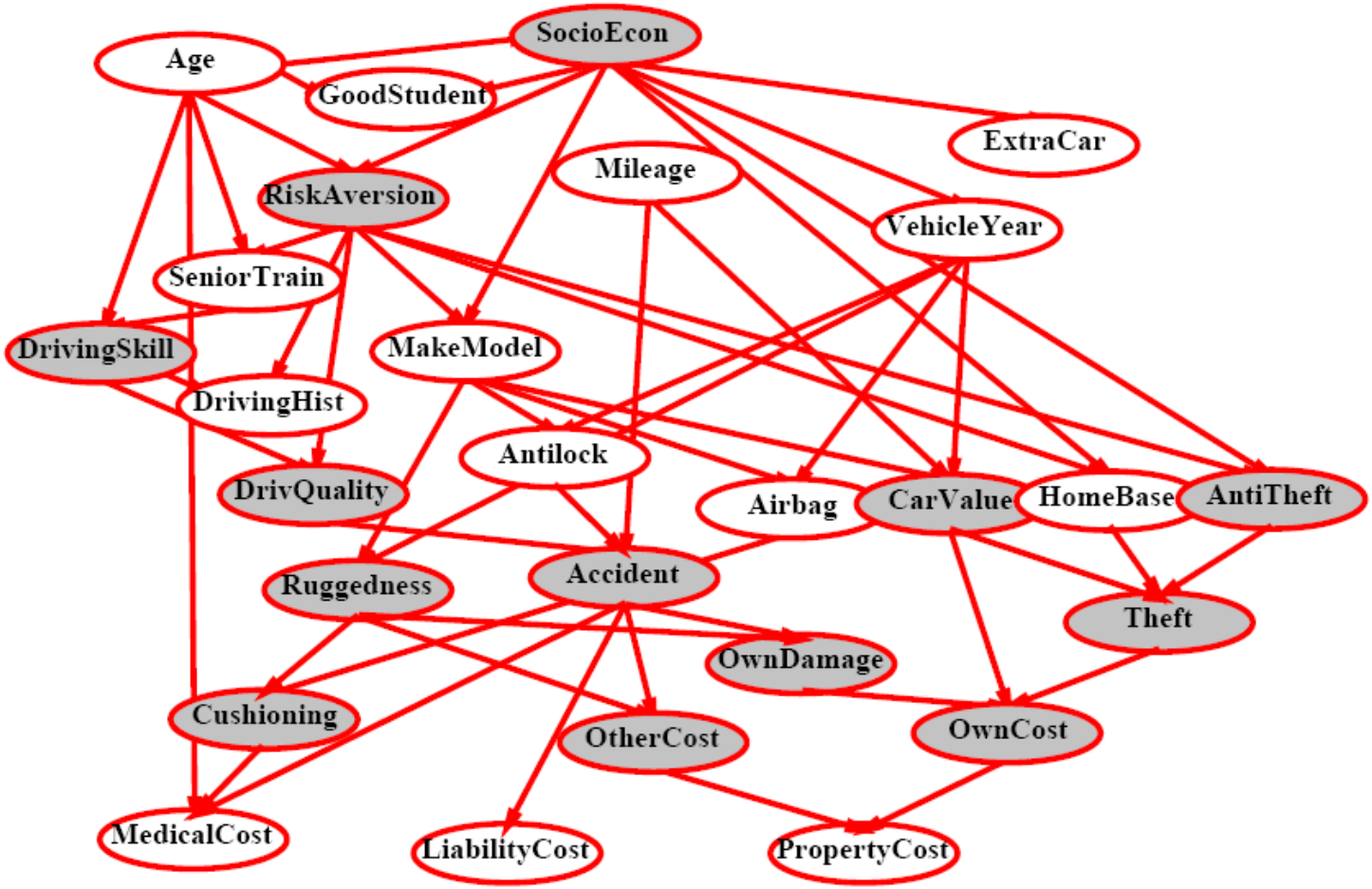
- Review: Bayesian inference
- Bayesian network: graph semantics
- The Los Angeles burglar alarm example
- Conditional independence ≠ Independence
- Constructing a Bayesian network: Structure learning
- Constructing a Bayesian network: Hire an expert

A more realistic Bayes Network: Car diagnosis

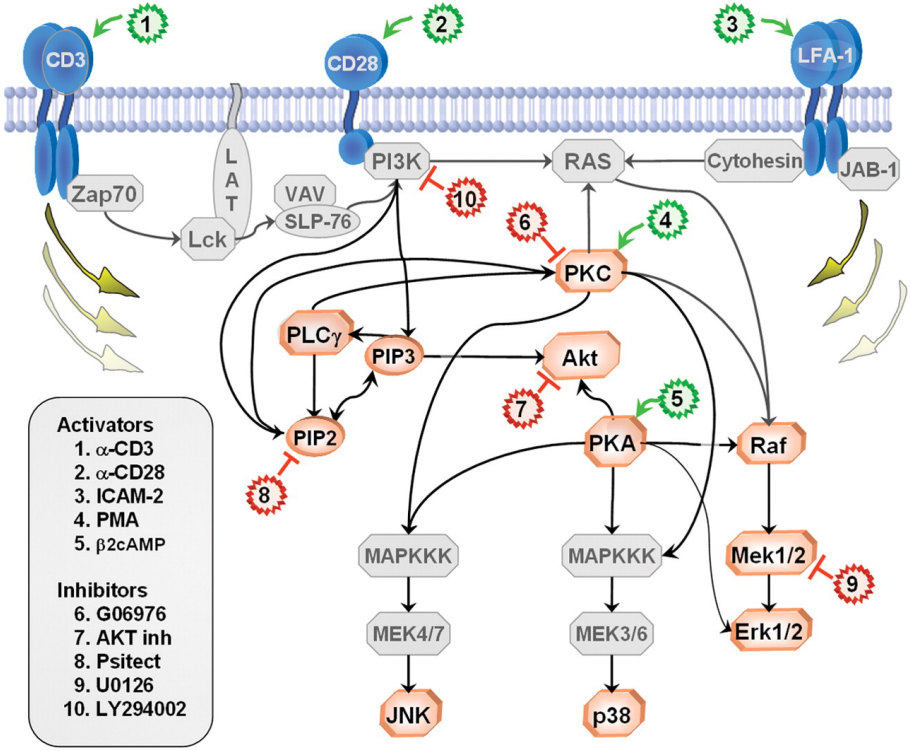
- Initial observation: car won't start
- Orange: "broken, so fix it" nodes
- Green: testable evidence
- Gray: "hidden variables" to ensure sparse structure, reduce parameters



Car insurance



In research literature…



Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan (22 April 2005) ***Science*** **308** (5721), 523.

In research literature…

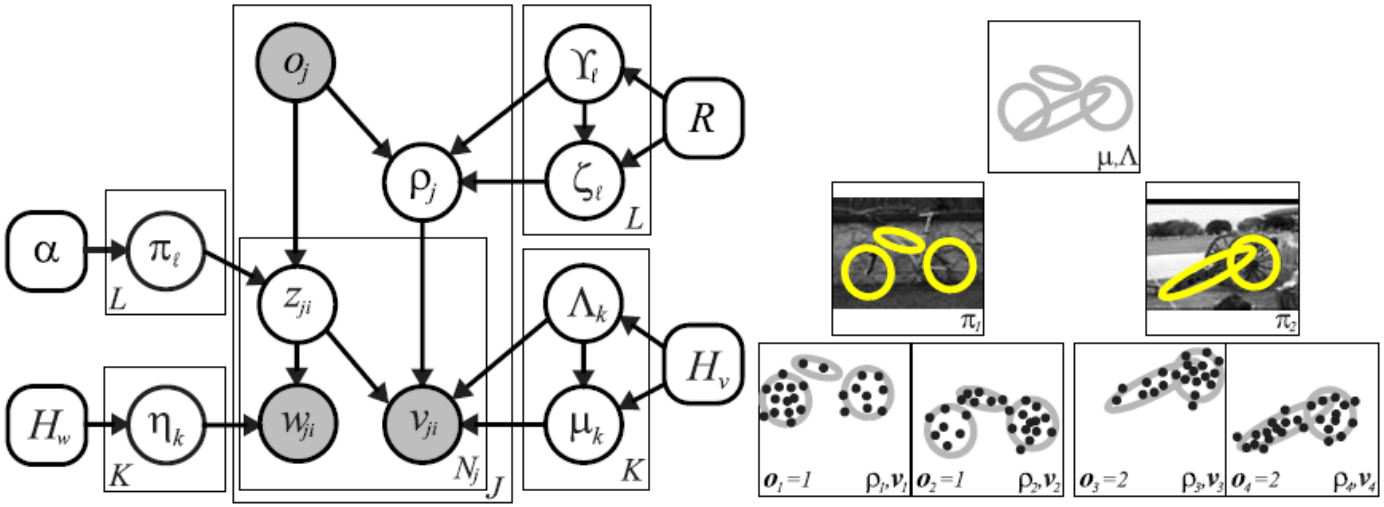


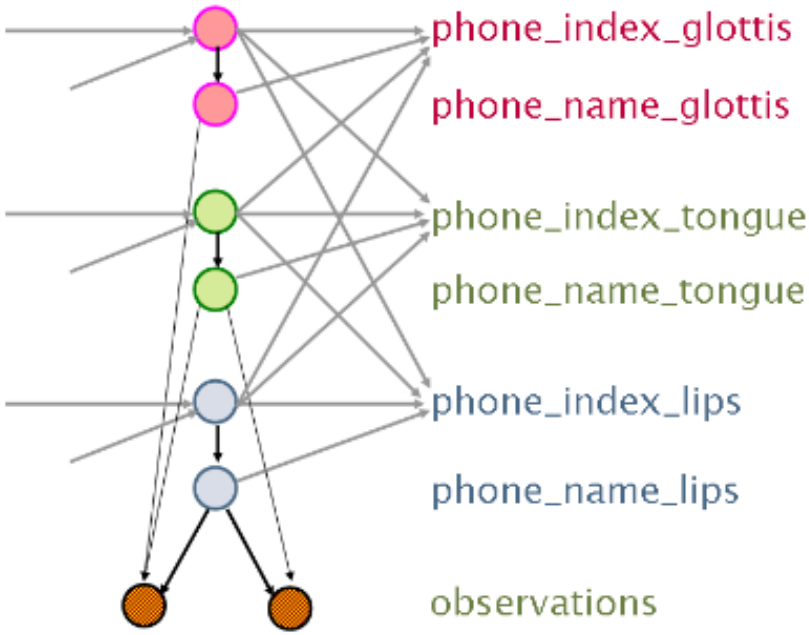
Fig. 3 A parametric, fixed-order model which describes the visual appearance of L object categories via a common set of K shared parts. The j^{th} image depicts an instance of object category o_j, whose position is determined by the reference transformation ρ_j. The appearance w_{ji} and position v_{ji}, relative to ρ_j, of visual features are determined by assignments $z_{ji} \sim \pi_{o_j}$ to latent parts. The cartoon example illustrates how a wheel part might be shared among two categories, *bicycle* and *cannon*. We show feature positions (but not appearance) for two hypothetical samples from each category

Describing Visual Scenes Using Transformed Objects and Parts

E. Sudderth, A. Torralba, W. T. Freeman, and A. Willsky.

International Journal of Computer Vision, No. 1-3, May 2008, pp. 291-330.

In research literature…

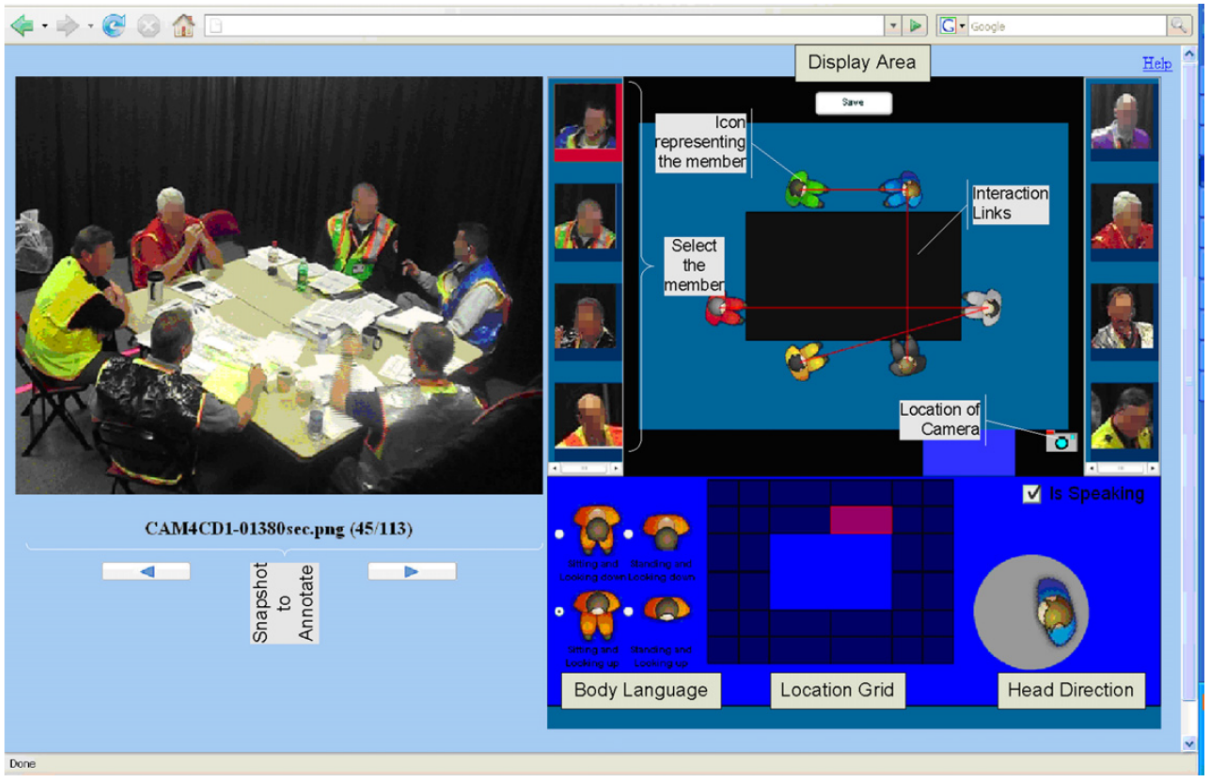


Audiovisual Speech Recognition with Articulator Positions as Hidden Variables

Mark Hasegawa-Johnson, Karen Livescu, Partha Lal and Kate Saenko

International Congress on Phonetic Sciences 1719:299-302, 2007

In research literature…

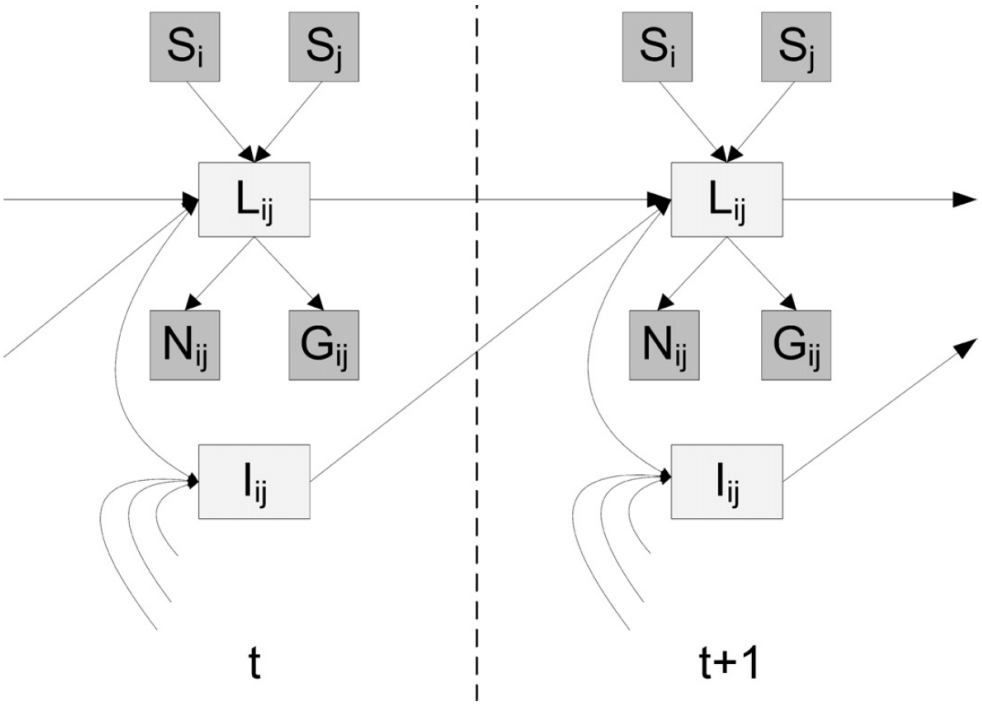


[Detecting interaction links in a collaborating group using manually annotated data]

S. Mathur, M.S. Poole, F. Pena-Mora, M. Hasegawa-Johnson, N. Contractor

Social Networks 10.1016/j.socnet.2012.04.002

In research literature…



- **Link:** $L_{ij} = 1$ if #i is listening to #j.
- **Indirect:** $I_{ij} = 1$ if #i and #j are both listening to the same person.
- **Speaking**: $S_i = 1$ if the i'th person is speaking.
- **Gaze:** $G_{ij} = 1$ if #i is looking at #j.
- **Neighborhood:** $N_{ij} = 1$ if they're near one another

Detecting interaction links in a collaborating group using manually annotated data

S. Mathur, M.S. Poole, F. Pena-Mora, M. Hasegawa-Johnson, N. Contractor

Social Networks 10.1016/j.socnet.2012.04.002

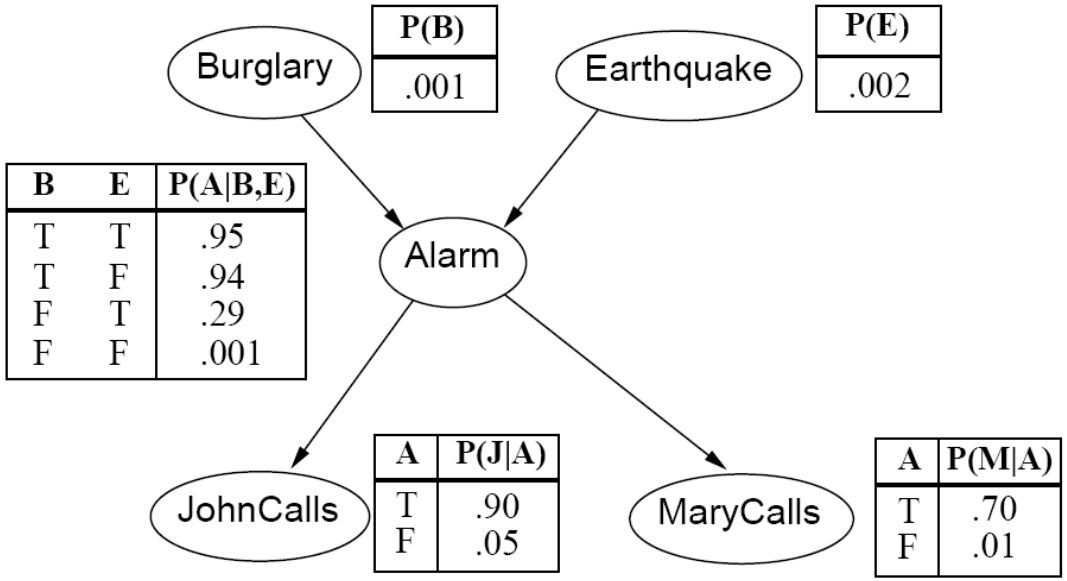
Summary

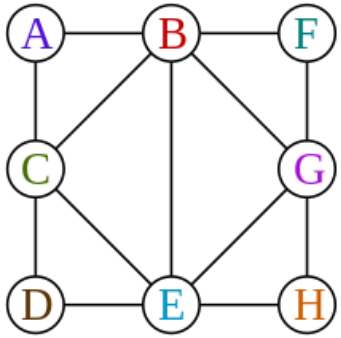
- Bayesian networks provide a natural representation for (causally induced) conditional independence
- Topology + conditional probability tables
- Generally easy for domain experts to construct

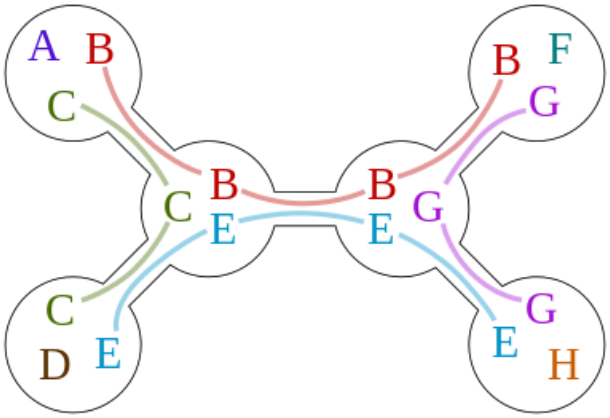
CS 440/ECE448 Lecture 19: Bayes Net Inference

Mark Hasegawa-Johnson, 3/2019 modified by Julia Hockenmaier 3/2019

Including slides by Svetlana Lazebnik, 11/2016







Bayes Net Inference and Learning

Bayes Network Inference & Learning

Bayes net is a **memory-efficient model** of dependencies among a set of random variables.

Inference problem: answer questions about the **query variables X** given the **evidence variables and their values E=e** as well as some **unobserved (hidden) variables Y**.

- We want to know the **posterior** distribution $P(\mathbf{X} \mid \mathbf{E} = \mathbf{e})$
- The posterior can be derived from the **full joint** $P(\mathbf{X}, \mathbf{E}, \mathbf{Y})$
- How do we make this **computationally efficient?**

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

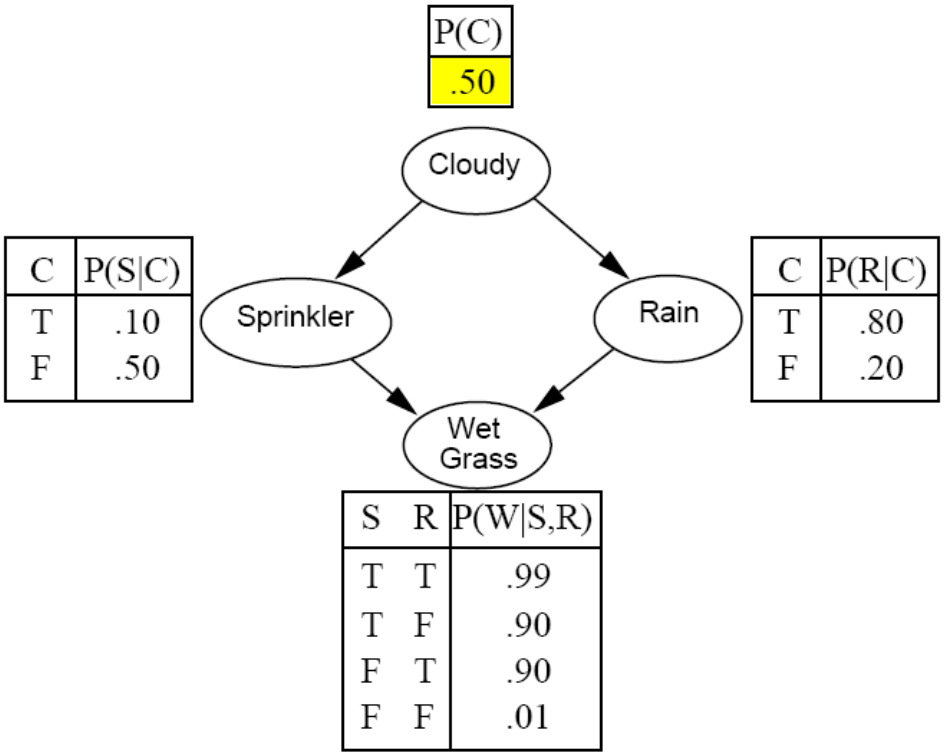
- Parameters = p(variable|parents), for each variable in the net

Outline

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

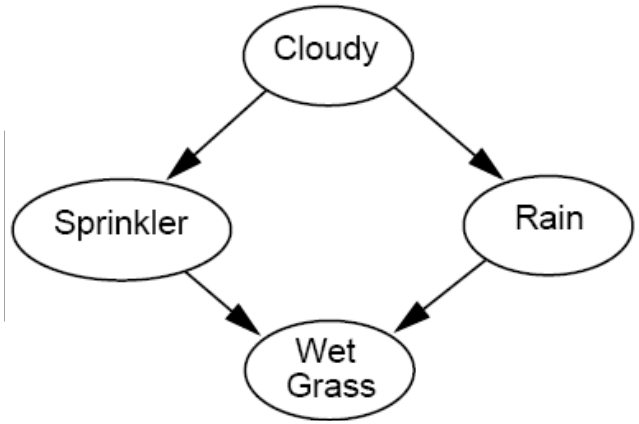
Practice example 1

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



Practice example 1

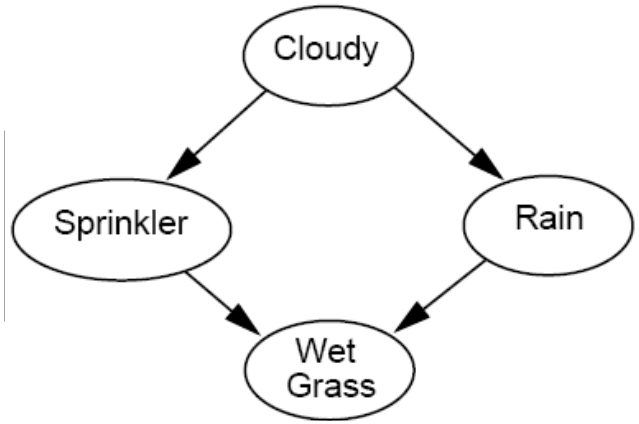
- Given that the grass is wet, what is the probability that it has rained?



$$P(r \mid w) = \frac{P(r,w)}{P(w)}$$

Practice example 1

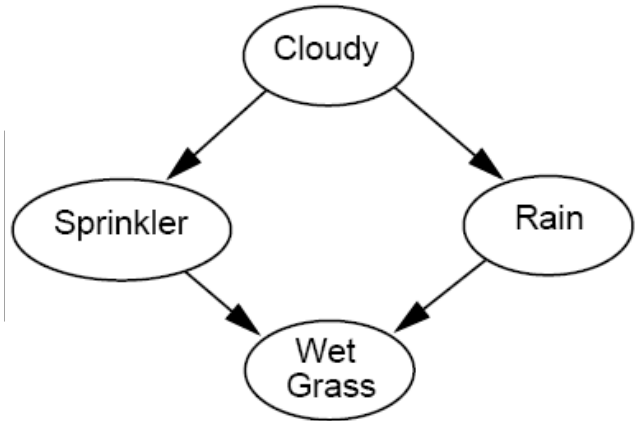
- Given that the grass is wet, what is the probability that it has rained?



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

Practice example 1

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



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

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

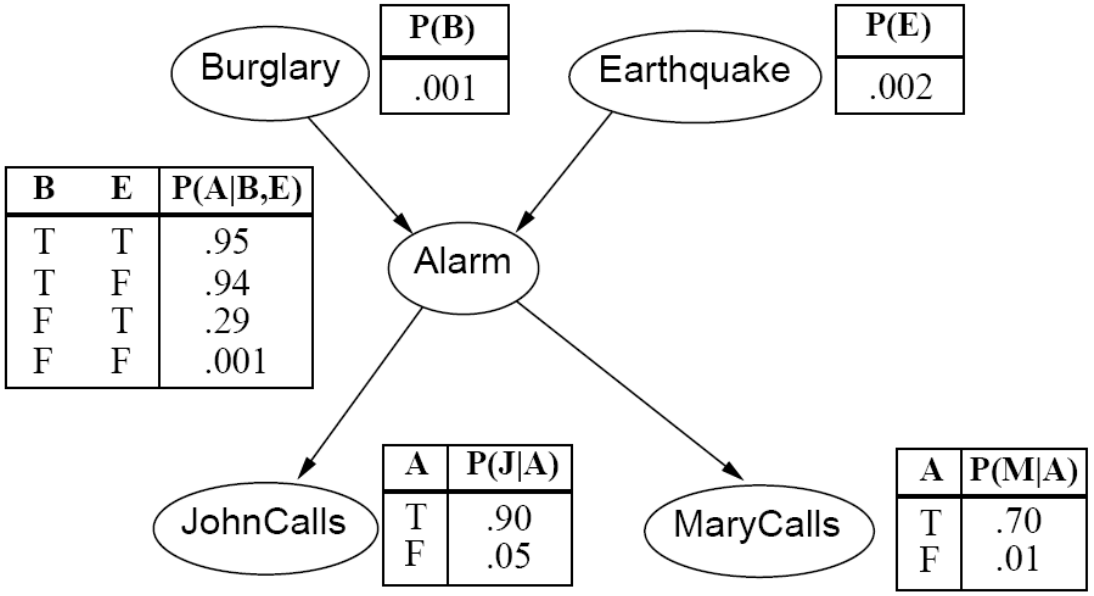
Practice Example #2

- 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 $\mathrm{P(B,J)}$ for all possible settings of $\mathrm{(B,J)}$

Bayes Net Inference: The Hard Way



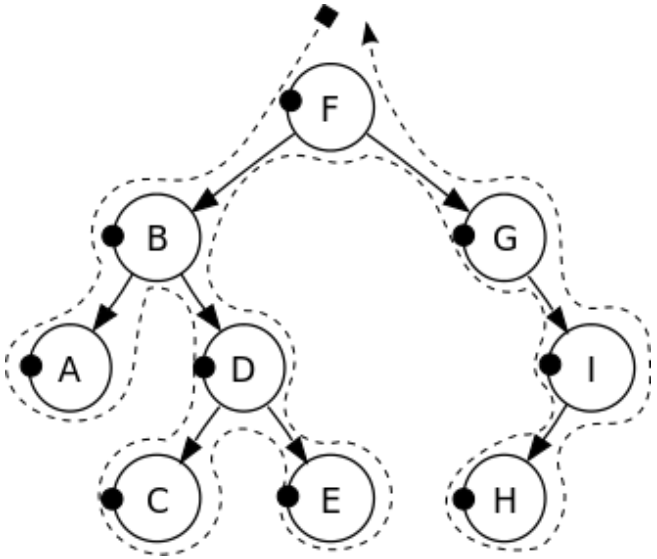
1. $\mathrm{P(B, E, A, J, M) = P(B)\ P(E)\ P(A|B,E)\ P(J|A)\ P(M|A)}$
2. $\mathrm{P(B, J) = \sum_E \sum_A \sum_M P(B, E, A, J, M)}$

Exponential complexity (#P-hard, actually): N variables, each of which has K possible values ⇒ $O\{K^N\}$ 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, $\boldsymbol{O\{K^N\}}$**
  - The SAT problem is a Bayes net!

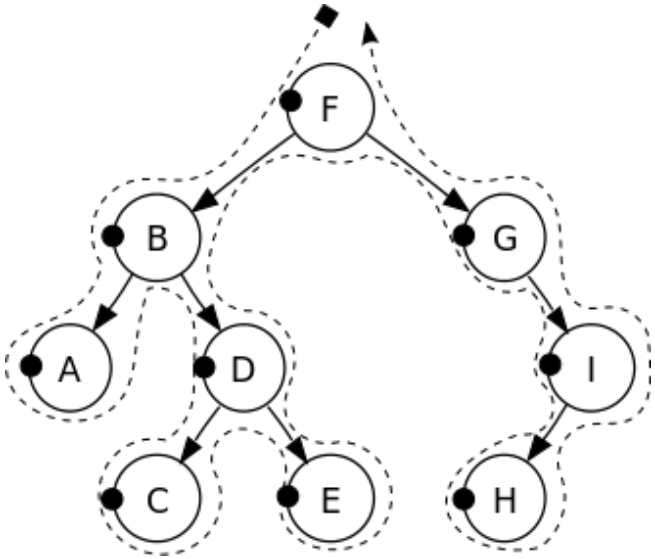
1. Tree-Structured Bayes Nets



- 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

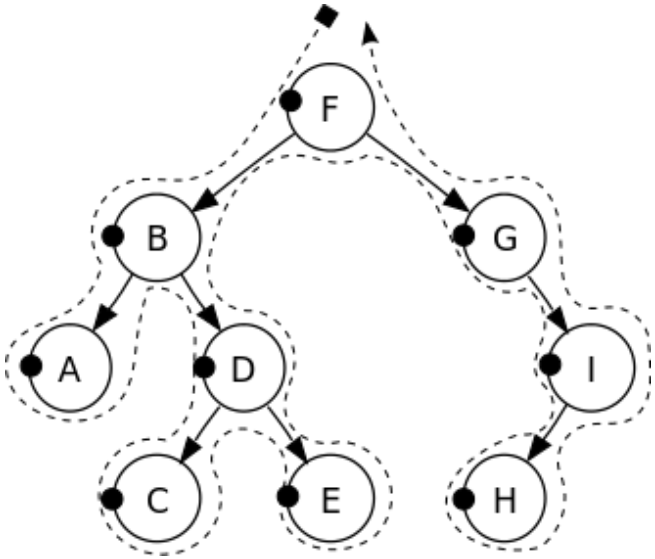
$$P(H = 1|E = 1) = \frac{P(H = 1, E = 1)}{\sum_h P(H = h, E = 1)}$$

The Sum-Product Algorithm (Belief Propagation)



- Find the **only undirected path from the evidence variable to the query variable** (E-D-B-F-G-I-H)
- 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)

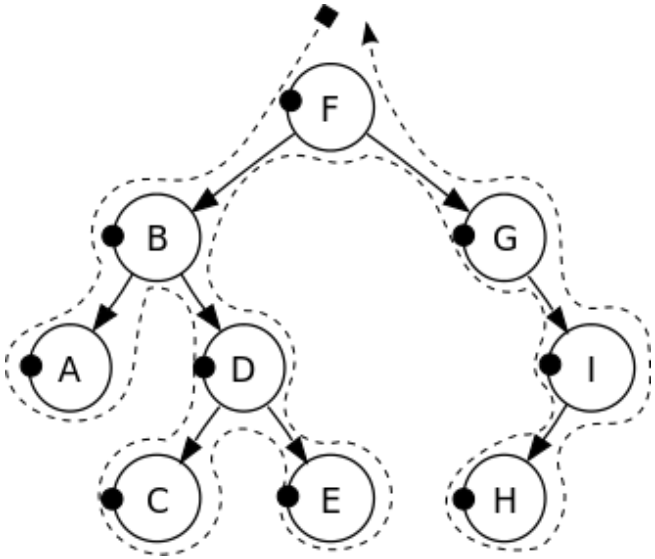
The Sum-Product Algorithm (Belief Propagation)



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

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

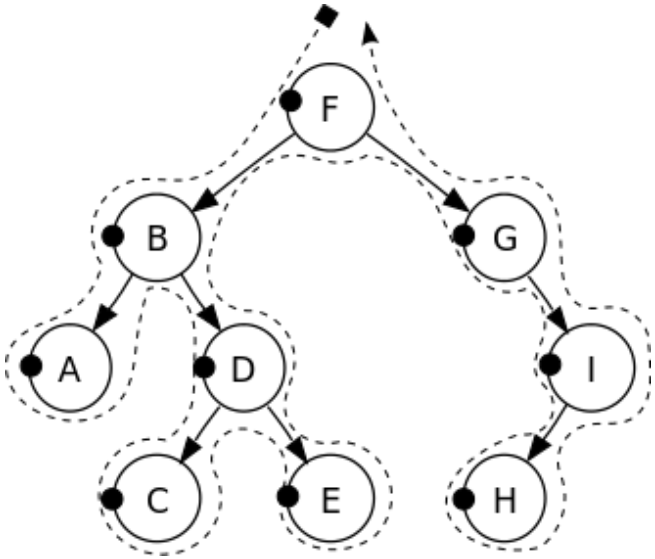
The Sum-Product Algorithm (Belief Propagation)



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

1. Product: $\mathrm{P(E,F,G)=P(E,F)P(G|F)}$
2. Sum: $P(E,G) = \sum_{F=0}^{1} P(E,F,G)$
3. Product: $\mathrm{P(E,G,I)=P(E,G)P(I|G)}$
4. Sum: $P(E,I) = \sum_{G=0}^{1} P(E,G,I)$
5. Product: $\mathrm{P(E,H,I)=P(E,I)P(I|G)}$
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, $\boldsymbol{O\{K^N\}}$**
  - The SAT problem is a Bayes net!

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

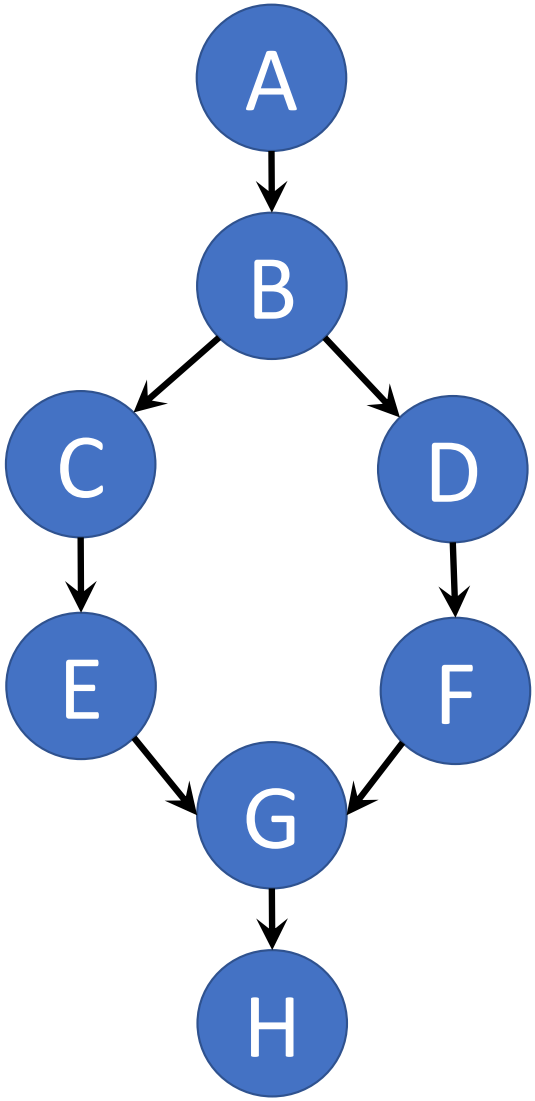
- Suppose there is a Bayes net with variables A,B,C,D,E,F,G,H
- The “Markov blanket” of variable F is D,E,G if

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

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

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



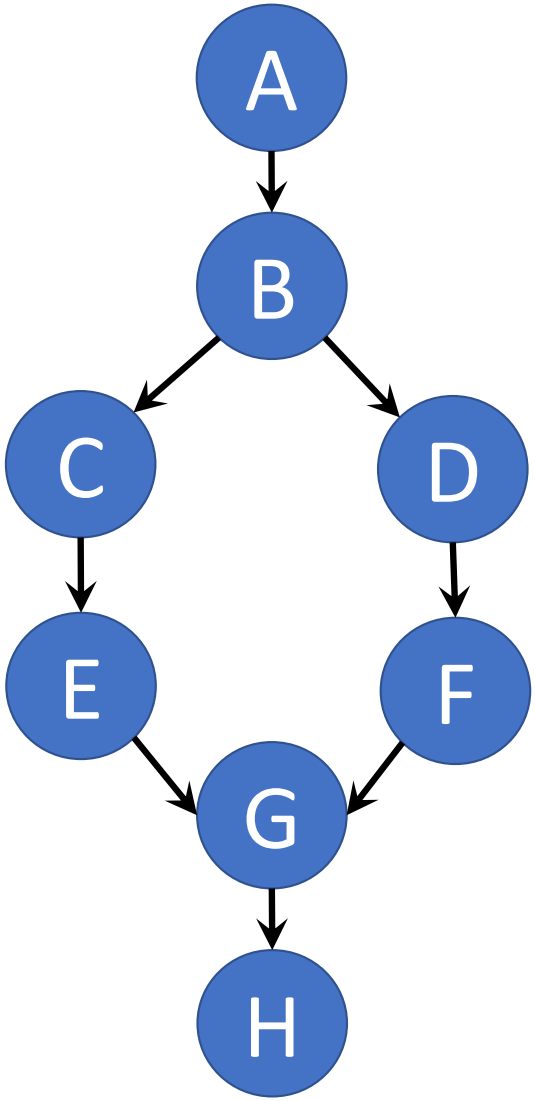
2.a. Markov Blanket

- 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?
- $P(A,\ldots,H) = P(A)P(B|A)$ …
- Which of those terms include F?



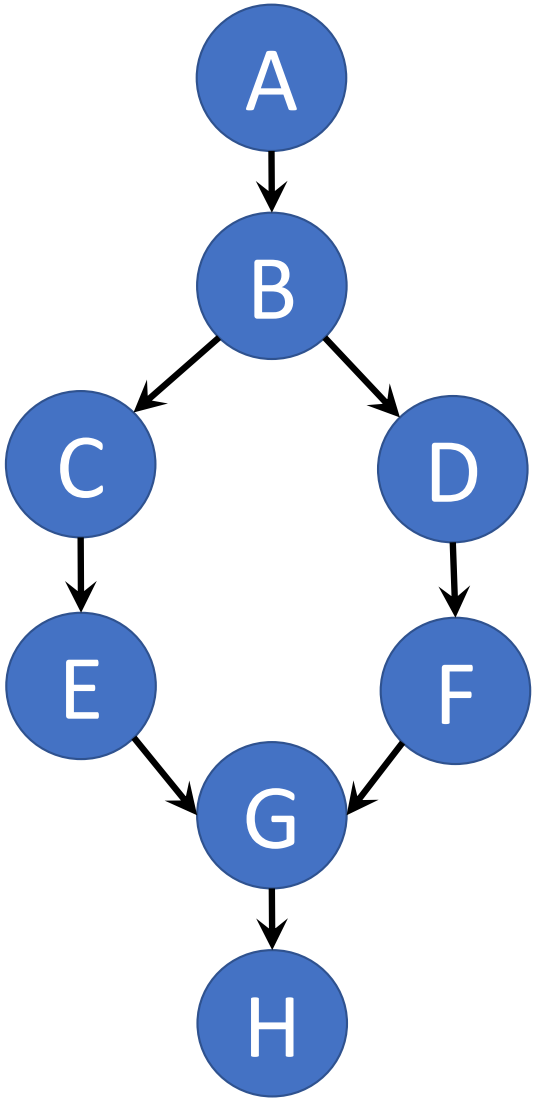
2.a. Markov Blanket

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

$$P(F|D)$$

and

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



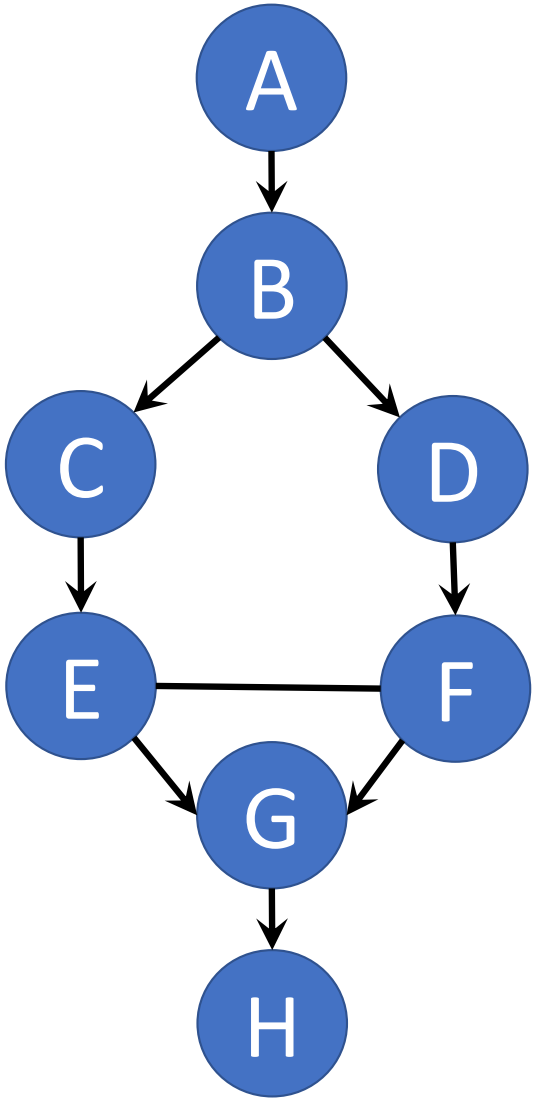
2.a. Markov Blanket

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)

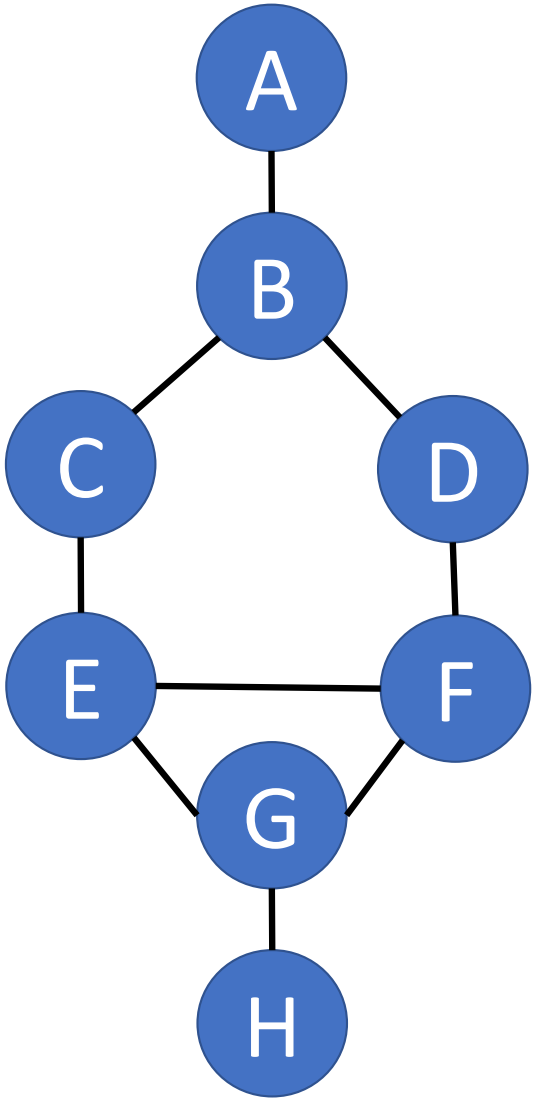


2.a. Moralization

"**Moralization**" =

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

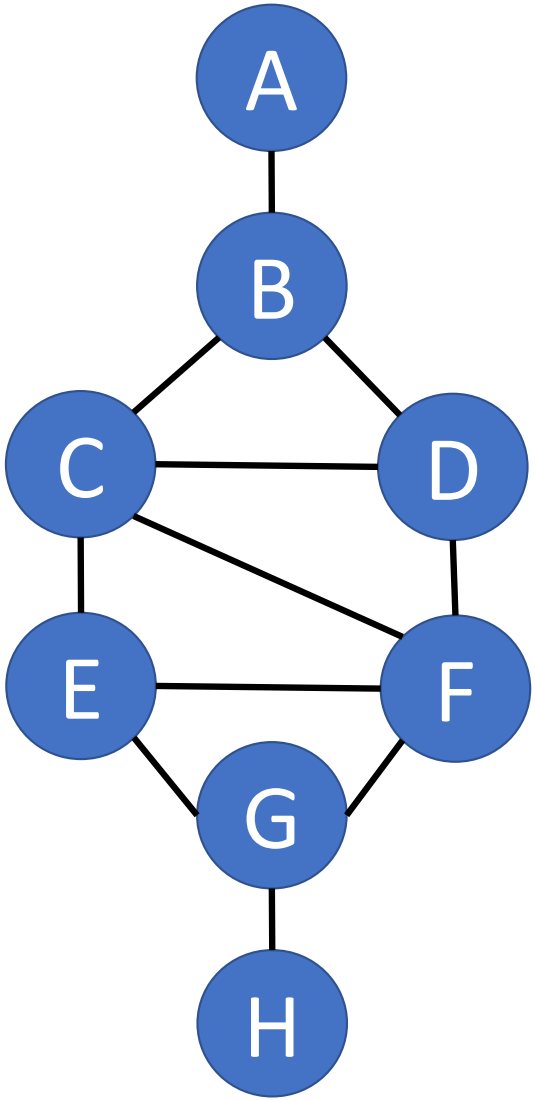
Result: **Markov blanket = the set of variables to which a variable is connected.**



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:

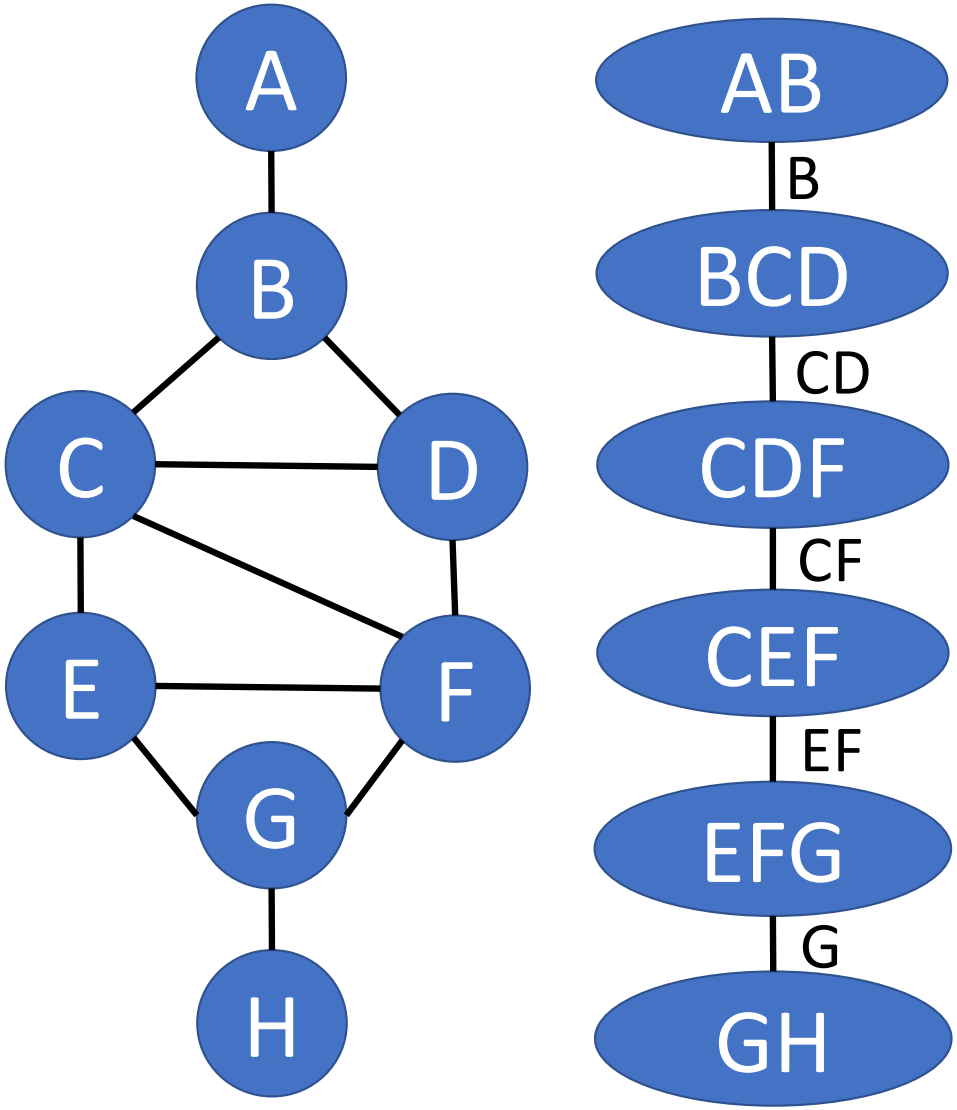


2.c. Form Cliques

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

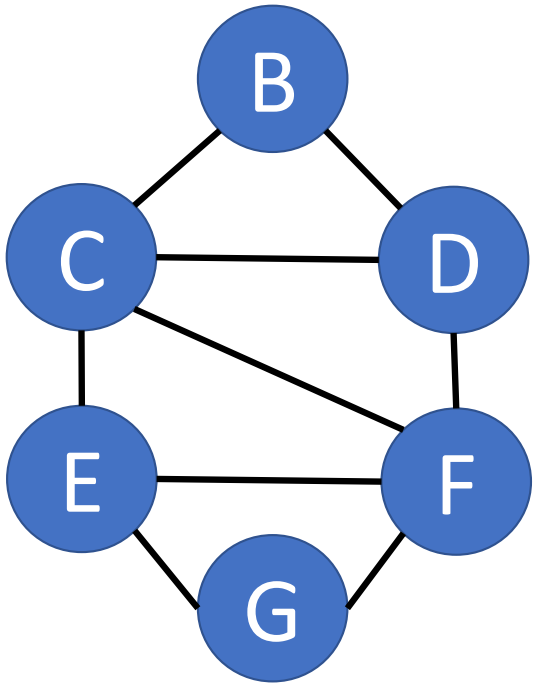
Suppose we need P(B,G):

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

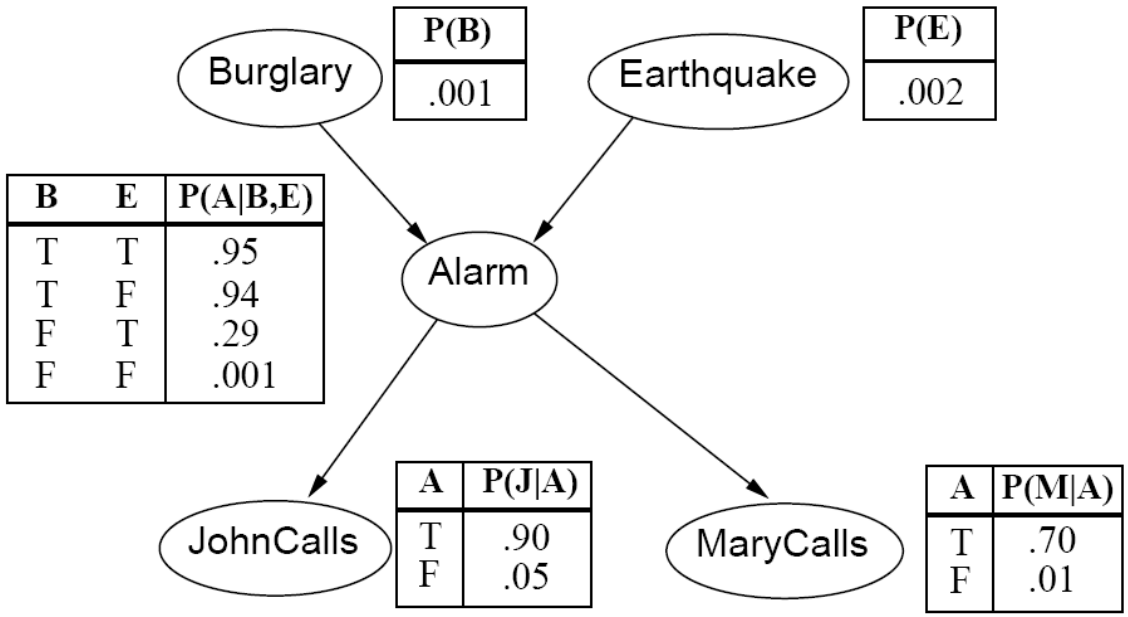
Complexity: $O\{NK^M\}$, where N=# cliques,

K = # values for each variable,

M = 1 + # variables in the largest clique



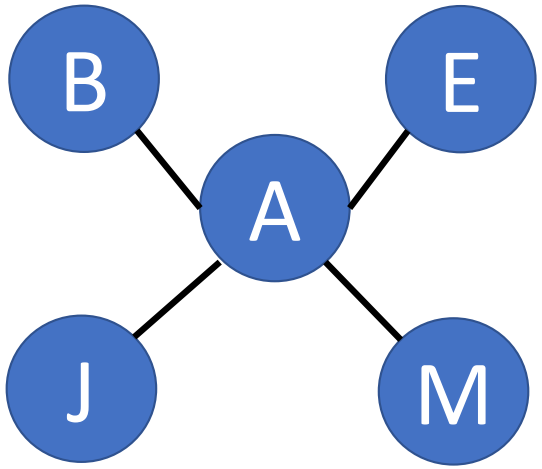
Junction Tree: Sample Test Question



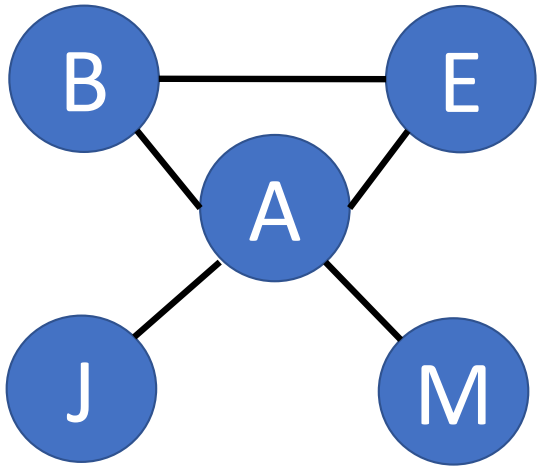
Consider the burglar alarm example.

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

Solution

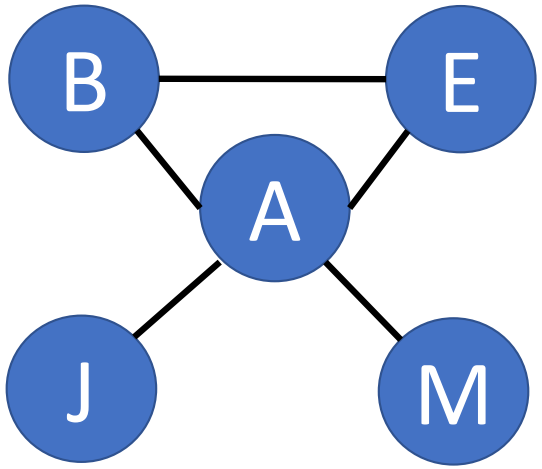


Solution



a. Moralize this graph

Solution

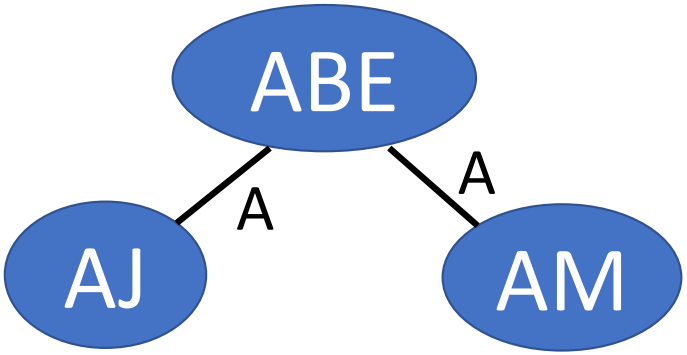


b. Is it already triangulated?

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, $\boldsymbol{O\{K^N\}}$**
  - The SAT problem is a Bayes net!

Bayesian network inference

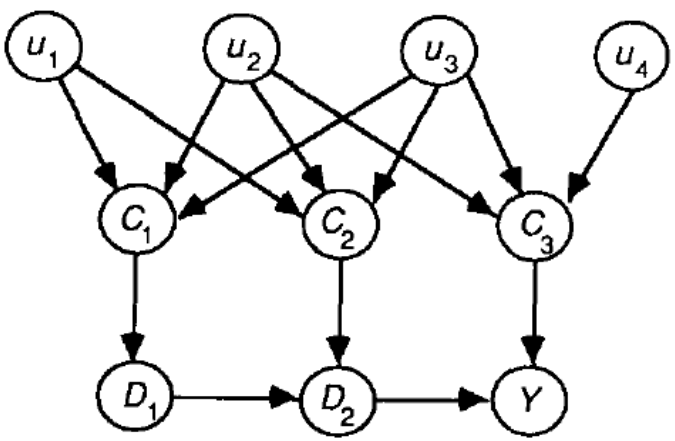
- In full generality, NP-hard
  - More precisely, #P-hard: equivalent to counting satisfying assignments
- We can reduce satisfiability to Bayesian network inference
  - Decision problem: is P(Y) > 0?

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

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?

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



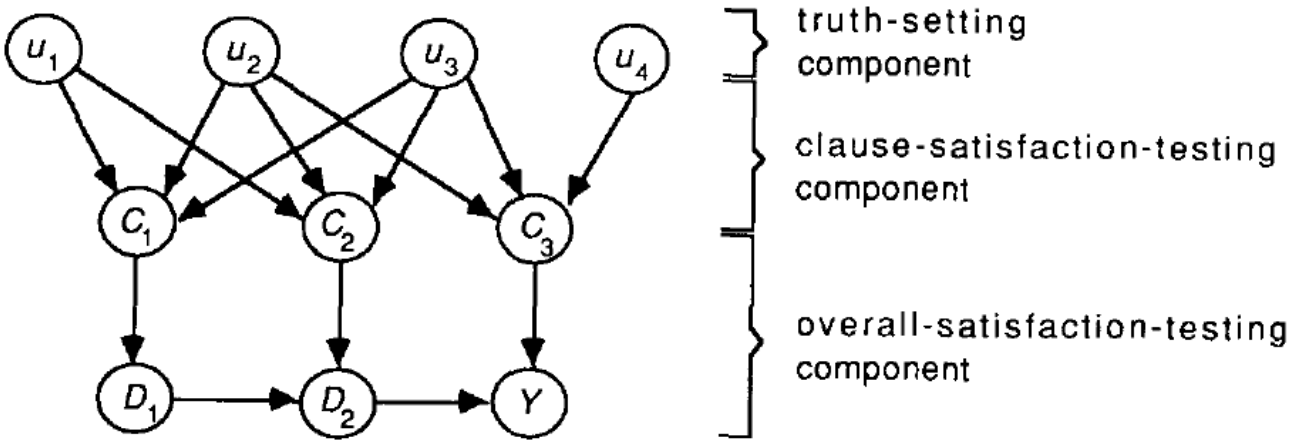
truth-setting component

clause-satisfaction-testing component

overall-satisfaction-testing component

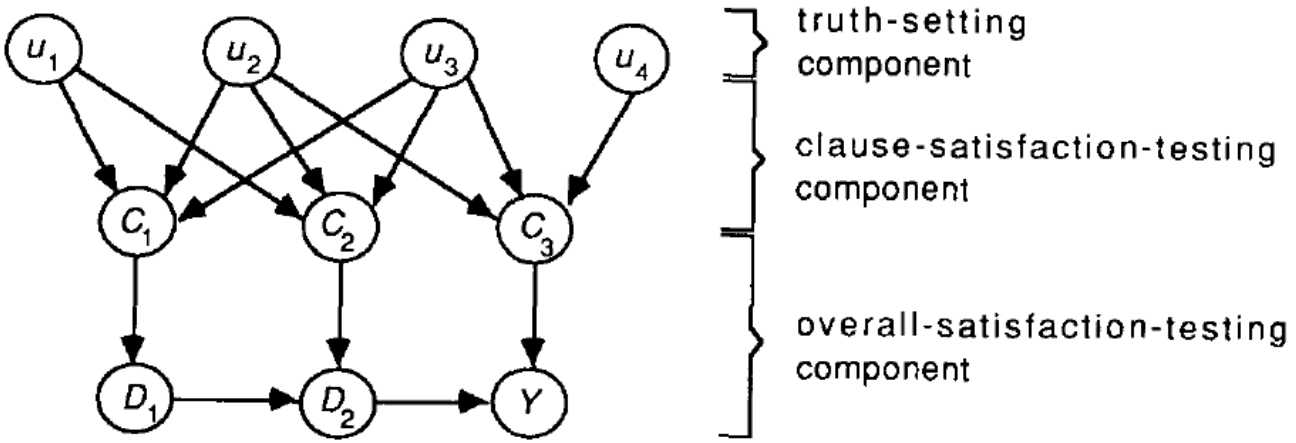
G. Cooper, 1990

Bayesian network inference



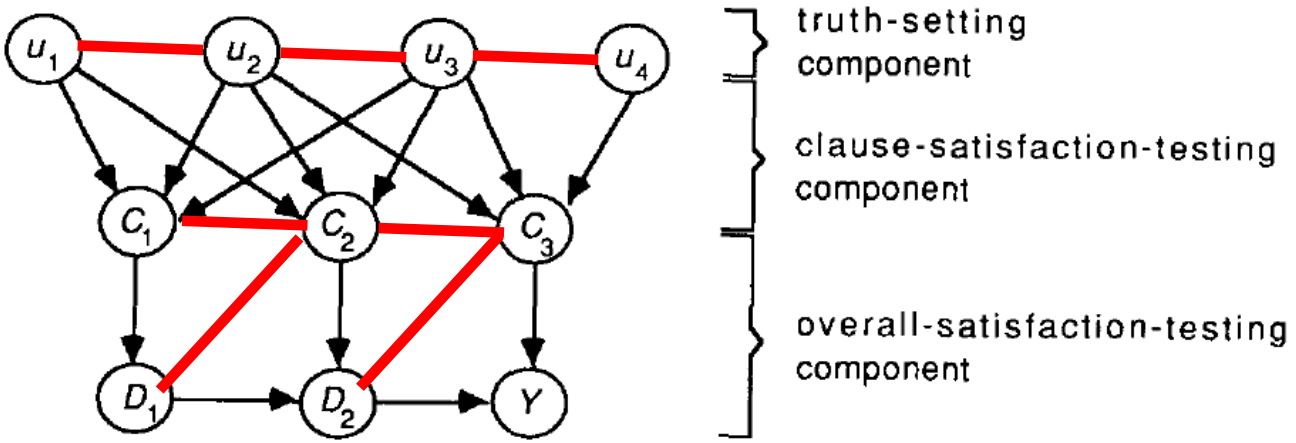
$$P(U_1,U_2,U_3,U_4,C_1,C_2,C_3,D_1,D_2,Y) =$$
$$P(U_1)P(U_2)P(U_3)P(U_4)$$
$$P(C_1 \mid U_1,U_2,U_3)P(C_2 \mid U_1,U_2,U_3)P(C_3 \mid U_2,U_3,U_4)$$
$$P(D_1 \mid C_1)P(D_2 \mid D_1,C_2)P(Y \mid D_2,C_3)$$

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

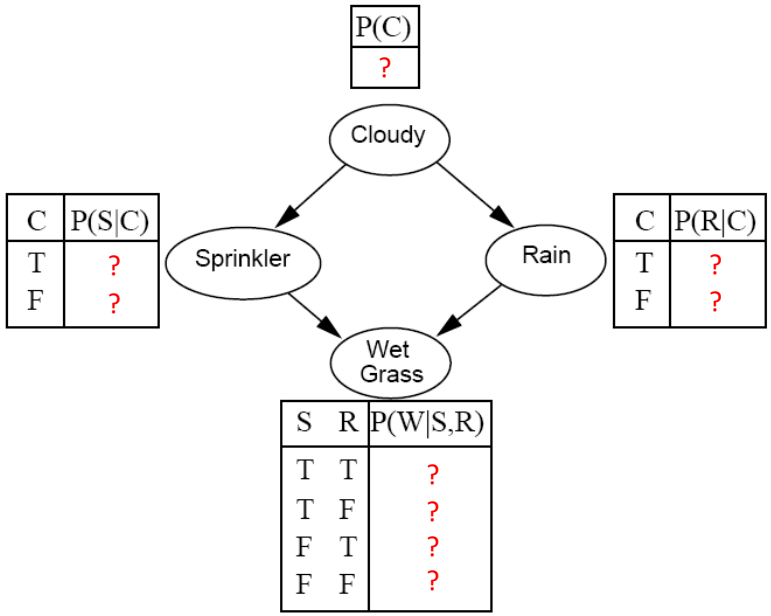
- Tree-structured Bayes nets: the sum-product algorithm
    - Quadratic complexity, $O\{NK^3\}$
- Polytrees: the junction tree algorithm
    - Pseudo-polynomial complexity, $O\{NK^M\}$, for M<N
- Arbitrary Bayes nets: #P complete, $O\{K^N\}$
    - The SAT problem is a Bayes net!

Parameter learning

- **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

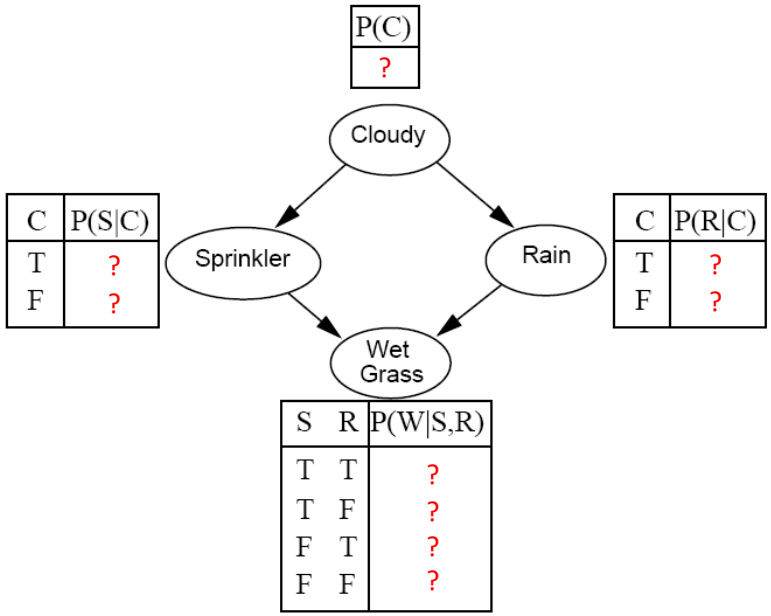
Sample	C	S	R	W
1	T	F	T	T
2	F	T	F	T
3	T	F	F	F
4	T	T	T	T
5	F	T	F	T
6	T	F	T	F
…	…	…	….	…

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***.

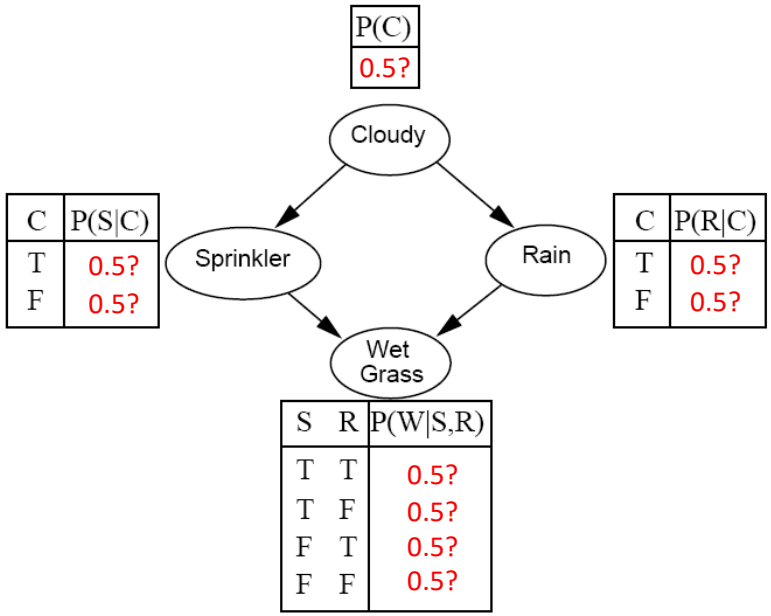


Training set

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



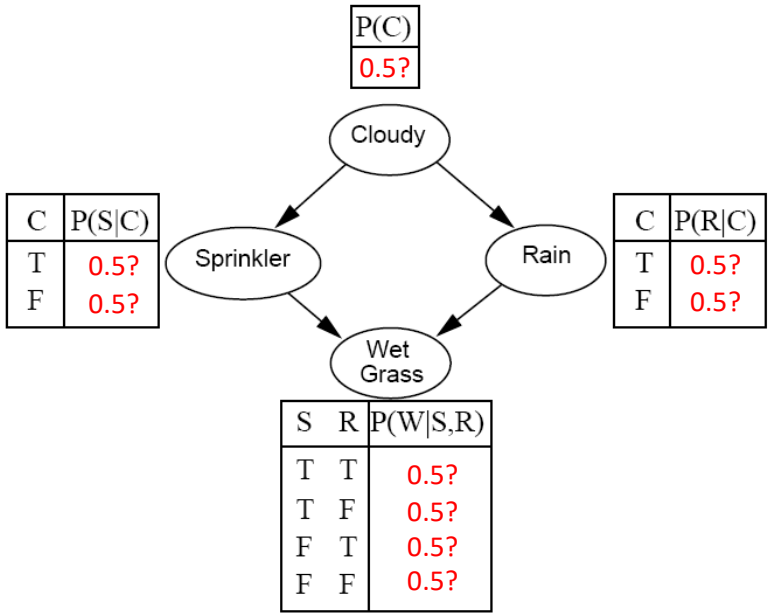
Training set

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

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



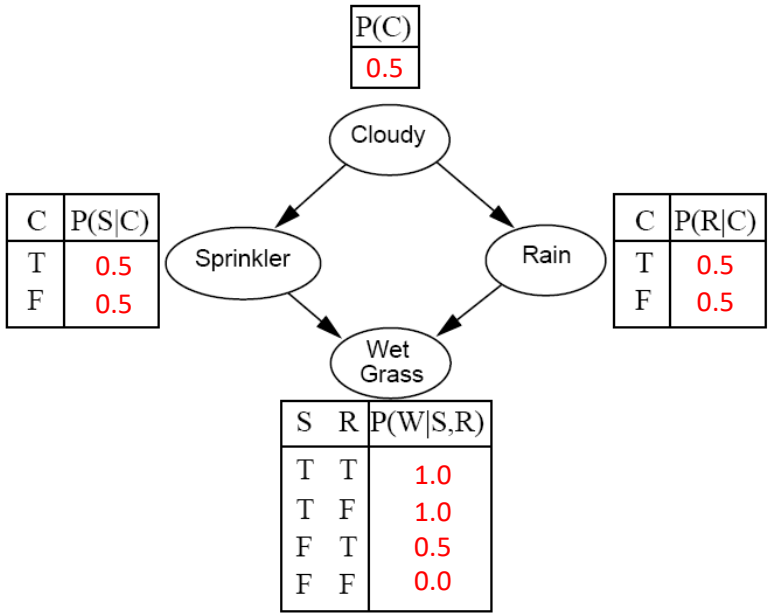
Training set

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

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

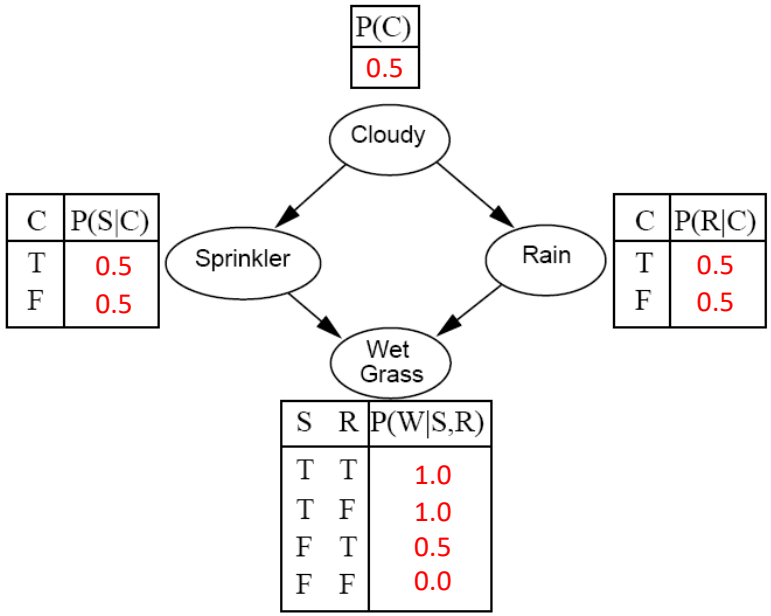


Training set

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