

Homework 3: Naive Bayes Classification

Bayesian Networks

Reading assignment:

S. Wooldridge, *Bayesian Belief Networks*

(linked from course webpage)

Copyrighted Material

PROBABILISTIC REASONING IN INTELLIGENT SYSTEMS:

Networks of Plausible Inference



Judea Pearl

REVISED SECOND PRINTING

A patient comes into a doctor's office with a fever and a bad cough.

Hypothesis space H :

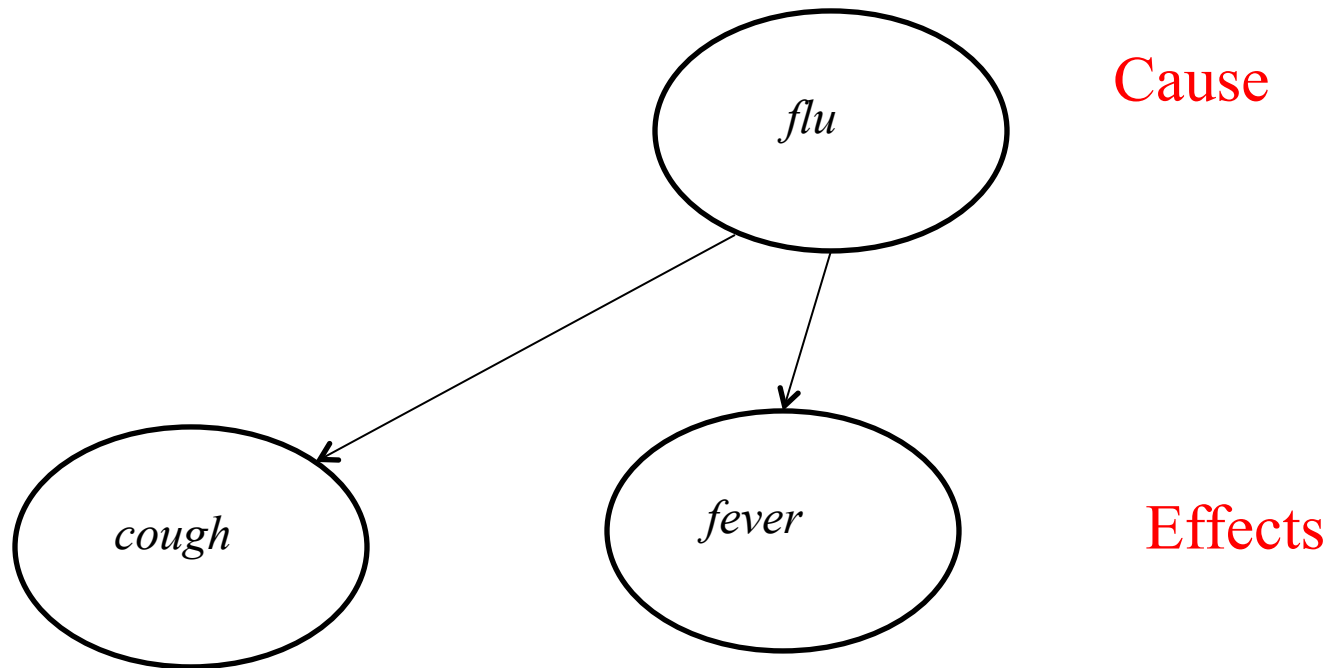
h_1 : patient has flu

h_2 : patient does not have flu

Data D :

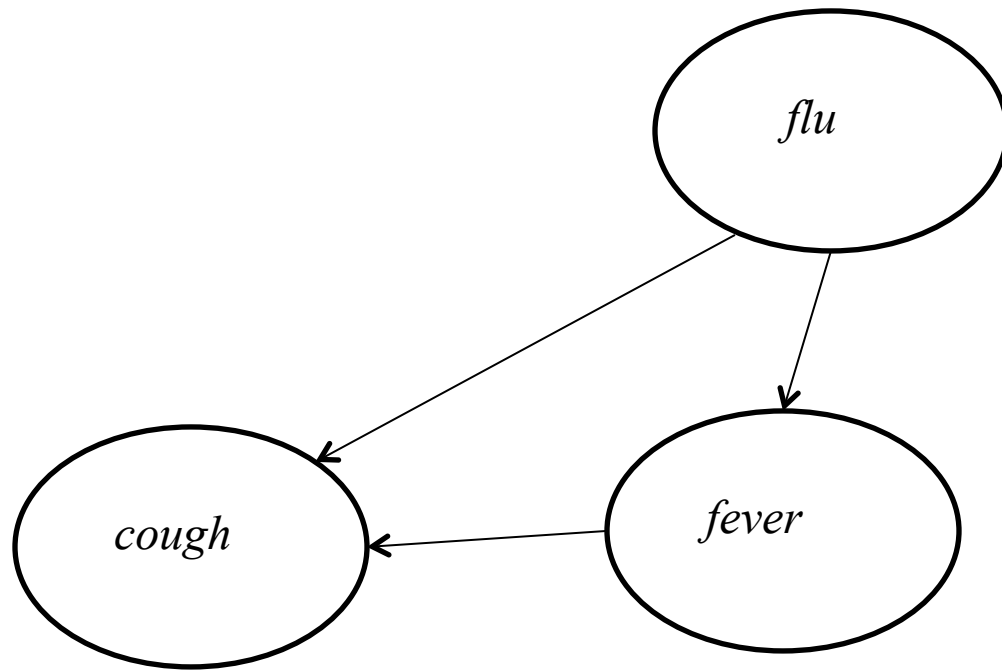
coughing = true, *fever* = true, *smokes* = true

Naive Bayes

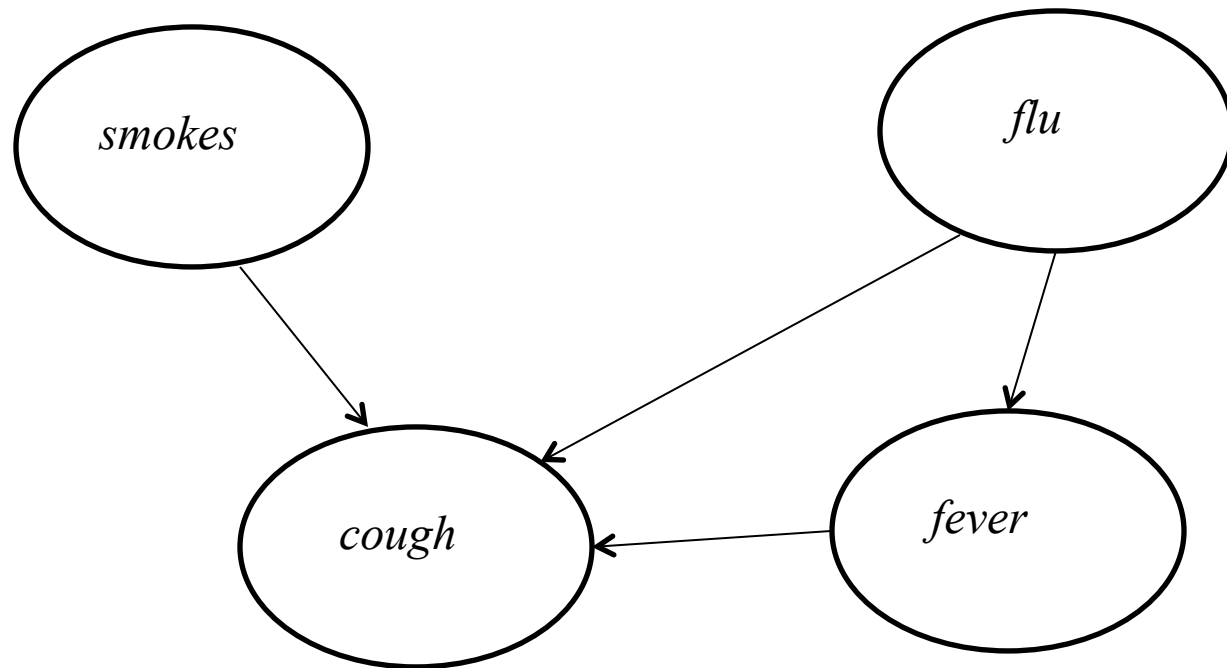


$$P(\text{flu} \mid \text{cough}, \text{fever}) \approx P(\text{flu})P(\text{cough} \mid \text{flu})P(\text{fever} \mid \text{flu})$$

What if attributes are not independent?



What if more than one possible cause?



Full joint probability distribution

<i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>Fever</i>	\neg <i>Fever</i>	<i>Fever</i>	\neg <i>Fever</i>
<i>flu</i>	p_1	p_2	p_3	p_4
\neg <i>flu</i>	p_5	p_6	p_7	p_8

Sum of all boxes
is 1.

In principle, the full joint distribution can be used to answer any question about probabilities of these combined parameters.

\neg <i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>fever</i>	\neg <i>fever</i>	<i>fever</i>	\neg <i>fever</i>
<i>flu</i>	p_9	p_{10}	p_{11}	p_{12}
\neg <i>flu</i>	p_{13}	p_{14}	p_{15}	p_{16}

However, size of full joint distribution scales exponentially with number of parameters so is expensive to store and to compute with.

Full joint probability distribution

<i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>Fever</i>	\neg <i>Fever</i>	<i>Fever</i>	\neg <i>Fever</i>
<i>flu</i>	p_1	p_2	p_3	p_4
\neg <i>flu</i>	p_5	p_6	p_7	p_8

For example, what if we had another attribute, “allergies”?

How many probabilities would we need to specify?

\neg <i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>fever</i>	\neg <i>fever</i>	<i>fever</i>	\neg <i>fever</i>
<i>flu</i>	p_9	p_{10}	p_{11}	p_{12}
\neg <i>flu</i>	p_{13}	p_{14}	p_{15}	p_{16}

<i>Allergy</i>				
<i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>Fever</i>	\neg <i>Fever</i>	<i>Fever</i>	\neg <i>Fever</i>
<i>flu</i>	p ₁	p ₂	p ₃	p ₄
\neg <i>flu</i>	p ₅	p ₆	p ₇	p ₈

\neg <i>Allergy</i>				
<i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>Fever</i>	\neg <i>Fever</i>	<i>Fever</i>	\neg <i>Fever</i>
<i>flu</i>	p ₁₇	p ₁₈	p ₁₉	p ₂₀
\neg <i>flu</i>	p ₂₁	p ₂₂	p ₂₃	p ₂₄

<i>Allergy</i>				
\neg <i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>fever</i>	\neg <i>fever</i>	<i>fever</i>	\neg <i>fever</i>
<i>flu</i>	p ₉	p ₁₀	p ₁₁	p ₁₂
\neg <i>flu</i>	p ₁₃	p ₁₄	p ₁₅	p ₁₆

\neg <i>Allergy</i>				
\neg <i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>fever</i>	\neg <i>fever</i>	<i>fever</i>	\neg <i>fever</i>
<i>flu</i>	p ₂₅	p ₂₆	p ₂₇	p ₂₈
\neg <i>flu</i>	p ₂₉	p ₃₀	p ₃₁	p ₃₂

<i>Allergy</i>				
<i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>Fever</i>	\neg <i>Fever</i>	<i>Fever</i>	\neg <i>Fever</i>
<i>flu</i>	p ₁	p ₂	p ₃	p ₄
\neg <i>flu</i>	p ₅	p ₆	p ₇	p ₈

\neg <i>Allergy</i>				
<i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>Fever</i>	\neg <i>Fever</i>	<i>Fever</i>	\neg <i>Fever</i>
<i>flu</i>	p ₁₇	p ₁₈	p ₁₉	p ₂₀
\neg <i>flu</i>	p ₂₁	p ₂₂	p ₂₃	p ₂₄

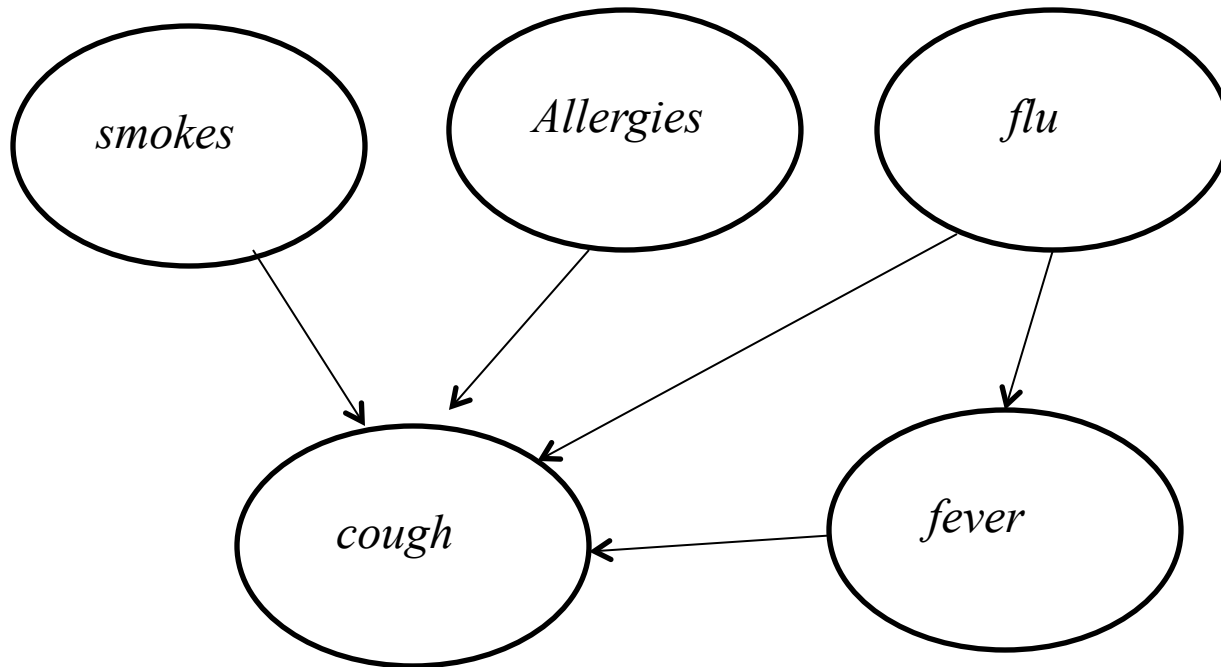
<i>Allergy</i>				
\neg <i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>fever</i>	\neg <i>fever</i>	<i>fever</i>	\neg <i>fever</i>
<i>flu</i>	p ₉	p ₁₀	p ₁₁	p ₁₂
\neg <i>flu</i>	p ₁₃	p ₁₄	p ₁₅	p ₁₆

\neg <i>Allergy</i>				
\neg <i>smokes</i>				
	<i>cough</i>		\neg <i>cough</i>	
	<i>fever</i>	\neg <i>fever</i>	<i>fever</i>	\neg <i>fever</i>
<i>flu</i>	p ₂₅	p ₂₆	p ₂₇	p ₂₈
\neg <i>flu</i>	p ₂₉	p ₃₀	p ₃₁	p ₃₂

But can reduce this if we know which variables are conditionally independent

Bayesian networks

- Idea is to represent dependencies (or causal relations) for all the variables so that space and computation-time requirements are minimized.



“Graphical Models”

Copyrighted Material

PROBABILISTIC GRAPHICAL MODELS

PRINCIPLES AND TECHNIQUES



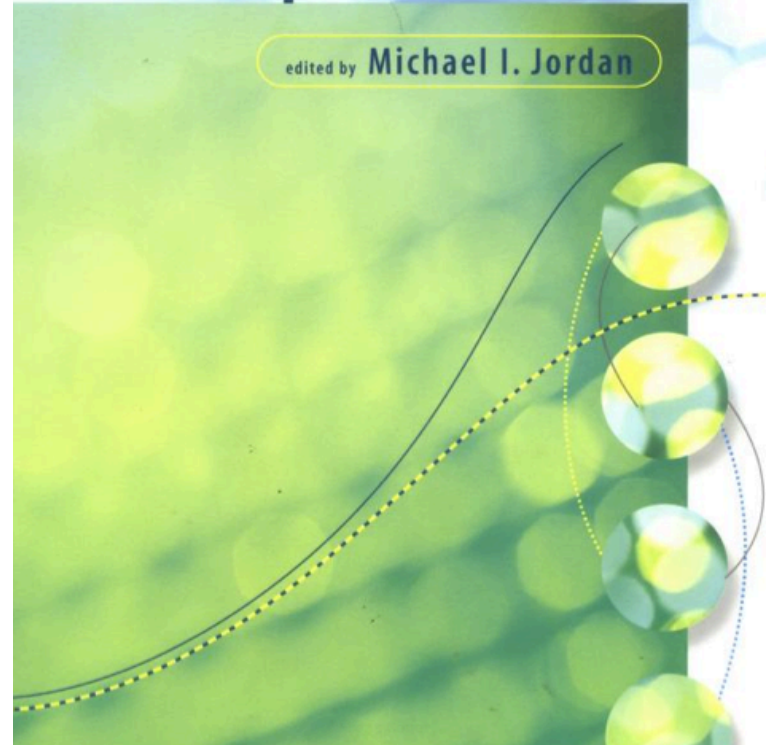
DAPHNE KOLLER AND NIR FRIEDMAN

Copyrighted Material

Copyrighted Material

Learning in Graphical Models

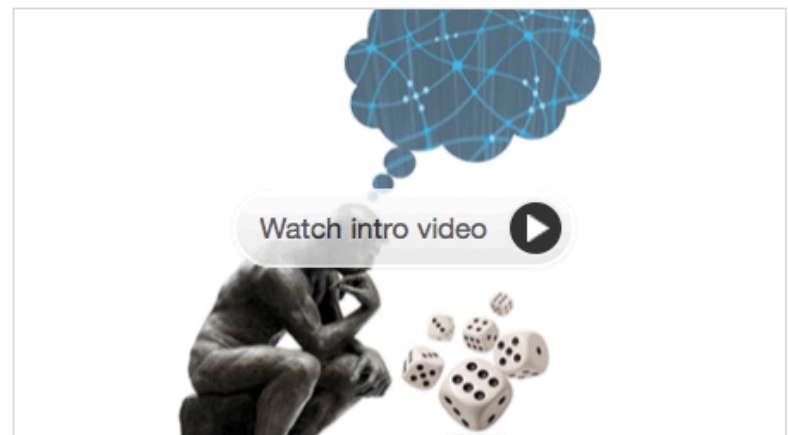
edited by **Michael I. Jordan**



Stanford

Probabilistic Graphical Models

In this class, you will learn the basics of the PGM representation and how to construct them, using both human knowledge and machine learning techniques.

[Preview Lectures](#)

Bayesian Networks = Bayesian Belief Networks = Bayes Nets

Bayesian Network: Alternative representation for complete joint probability distribution

“Useful for making probabilistic inference about models domains characterized by inherent complexity and uncertainty”

Uncertainty can come from:

- incomplete knowledge of domain
- inherent randomness in behavior in domain

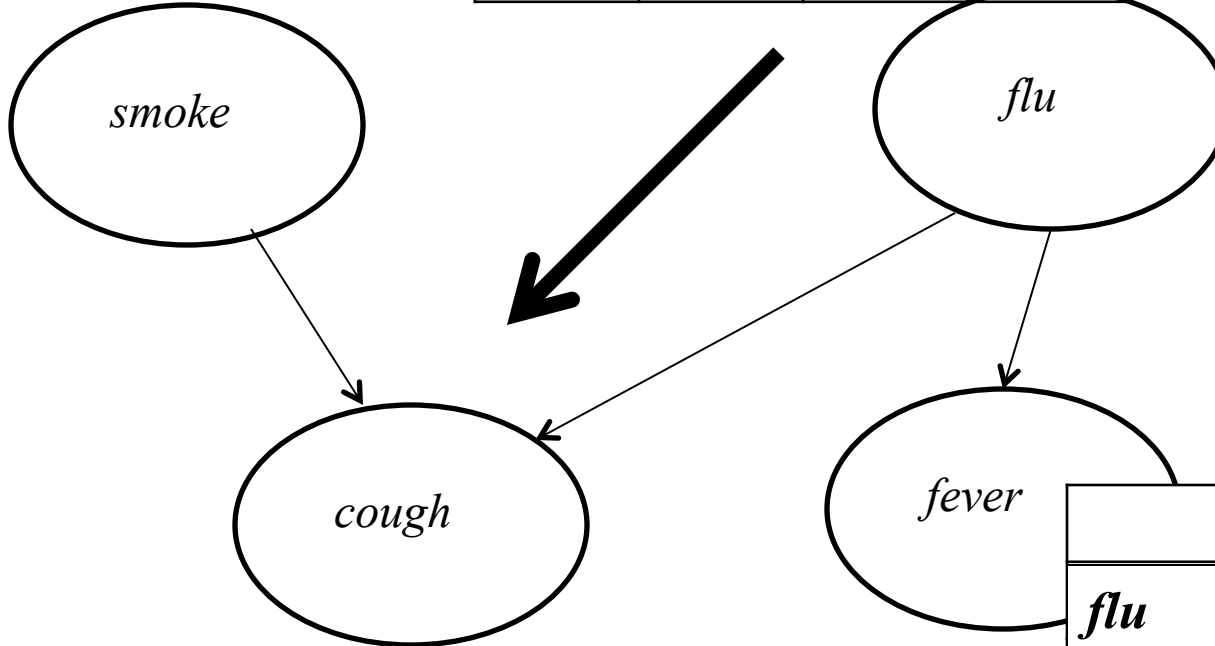
Example:

<i>smoke</i>	
true	0.2
false	0.8

<i>flu</i>	<i>smoke</i>	<i>cough</i>	
		true	false
True	True	0.95	0.05
True	False	0.8	0.2
False	True	0.6	0.4
false	false	0.05	0.95

Conditional probability tables for each node

<i>flu</i>	
true	0.01
false	0.99



<i>flu</i>	<i>fever</i>	
	true	false
true	0.9	0.1
false	0.2	0.8

Inference in Bayesian networks

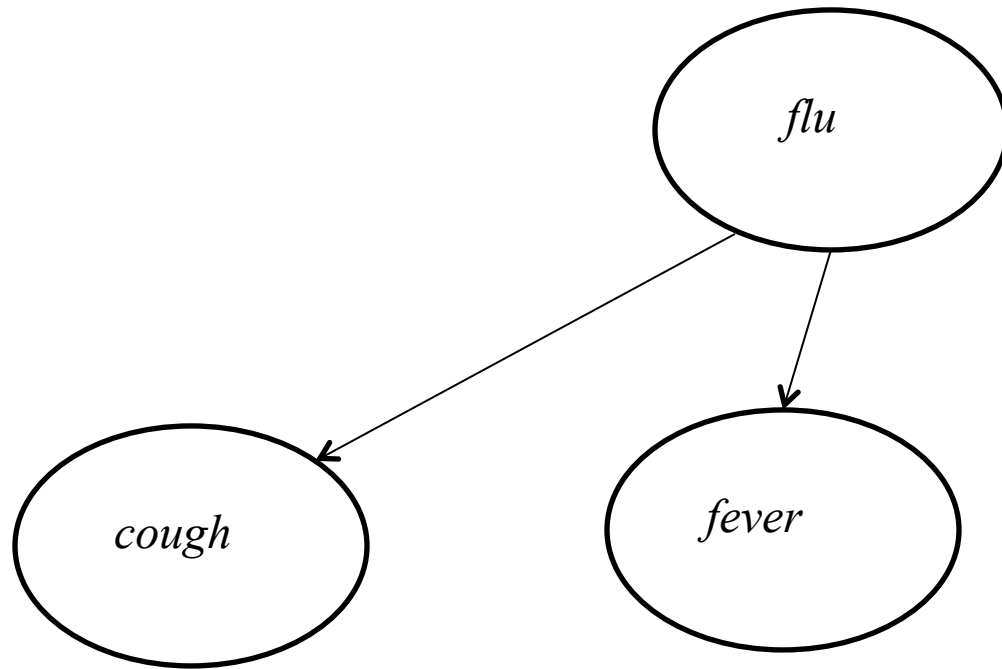
- If network is correct, can calculate full joint probability distribution from network.

$$P((X_1 = x_1) \wedge \dots \wedge (X_n = x_n))$$

$$= \prod_{i=1}^n P((X_i = x_i) \mid \text{parents}(X_i))$$

where $\text{parents}(X_i)$ denotes specific values of parents of X_i .

Naive Bayes Example



$$P(\text{flu} \mid \text{cough}, \text{fever}) \approx P(\text{flu})P(\text{cough} \mid \text{flu})P(\text{fever} \mid \text{flu})$$

Example

- Calculate

$$P(cough = t \wedge fever = f \wedge flu = f \wedge smoke = f)$$

Example

- Calculate

$$P(\text{cough} = t \wedge \text{fever} = f \wedge \text{flu} = f \wedge \text{smoke} = f)$$

$$= \prod_{i=1}^n P(X_i = x_i \mid \text{parents}(X_i))$$

$$= P(\text{cough} = t \mid \text{flu} = f \wedge \text{smoke} = f)$$

$$\times P(\text{fever} = f \mid \text{flu} = f)$$

$$\times P(\text{flu} = f)$$

$$\times P(\text{smoke} = f)$$

$$= .05 \times .8 \times .99 \times .8$$

$$= .032$$

In general...

- If network is correct, can calculate full joint probability distribution from network.

$$P(X_1, \dots, X_d) = \prod_{i=1}^d P(X_i \mid \text{parents}(X_i))$$

where $\text{parents}(X_i)$ denotes specific values of parents of X_i .

But need efficient algorithms to do this (e.g., “belief propagation”, “Markov Chain Monte Carlo”).

Example from the reading:

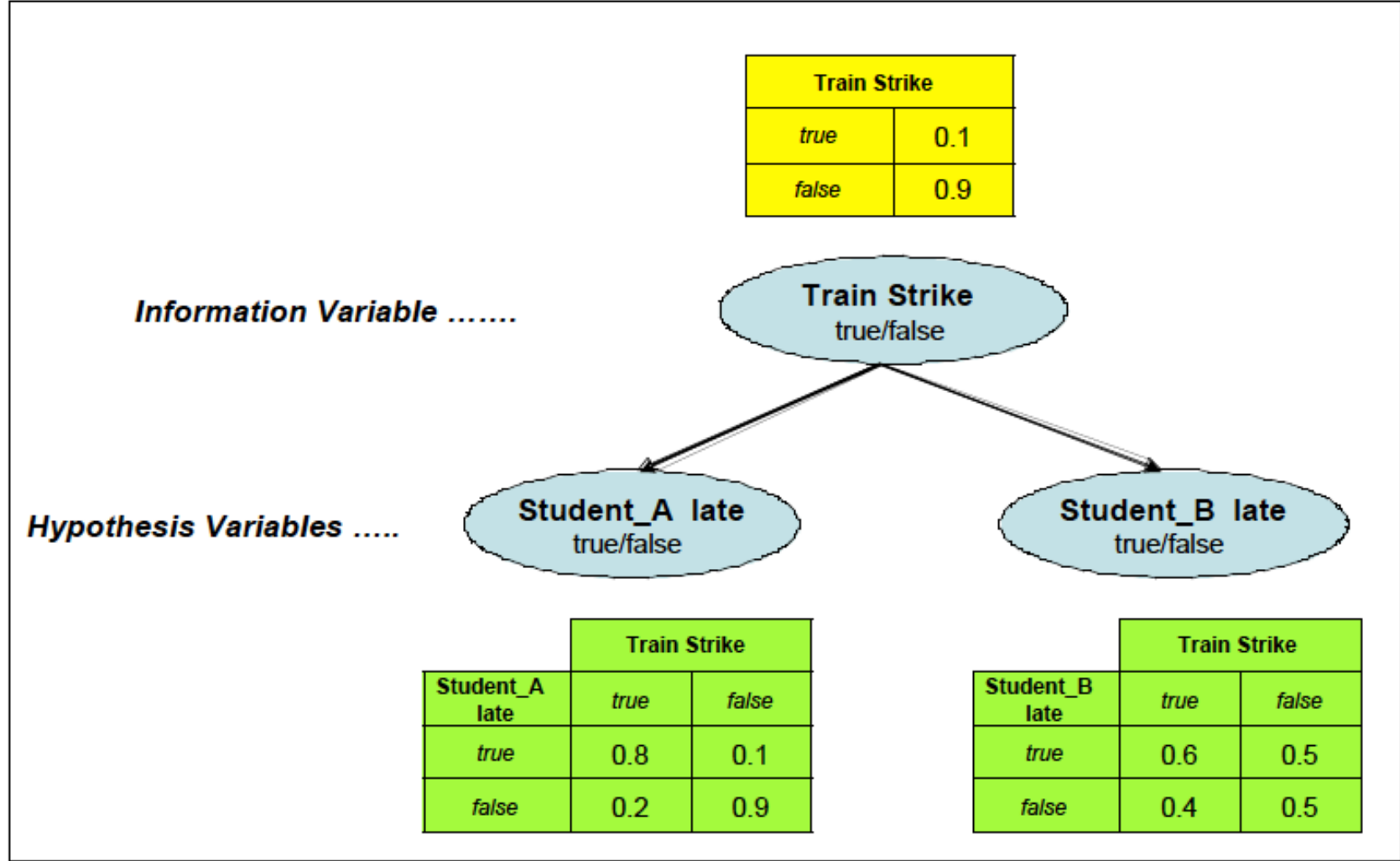


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

What is the probability that Student A is late?

What is the probability that Student B is late?

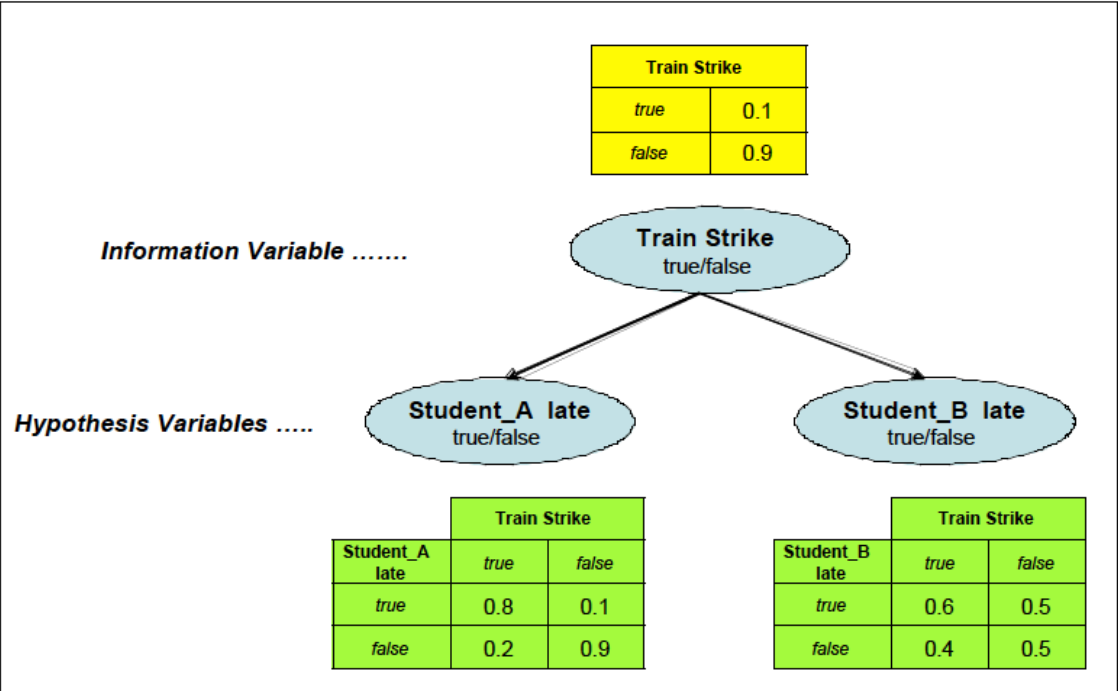


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

What is the probability that Student A is late?

What is the probability that Student B is late?

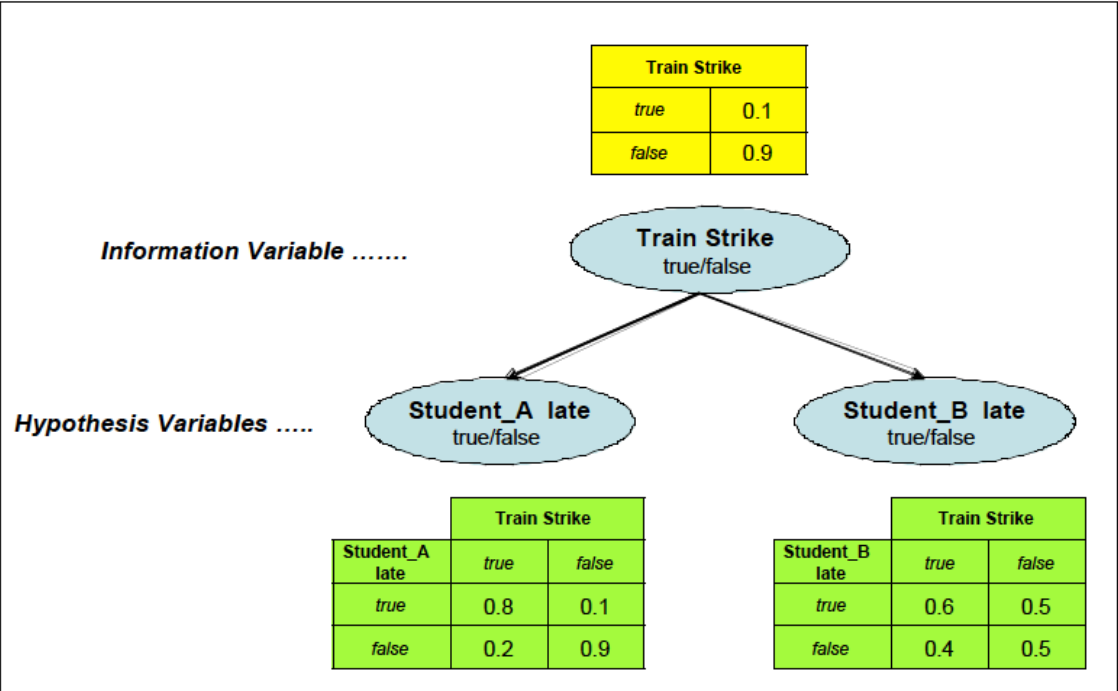


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Unconditional (“marginal”) probability. We don’t know if there is a train strike.

What is the probability that Student A is late?

What is the probability that Student B is late?

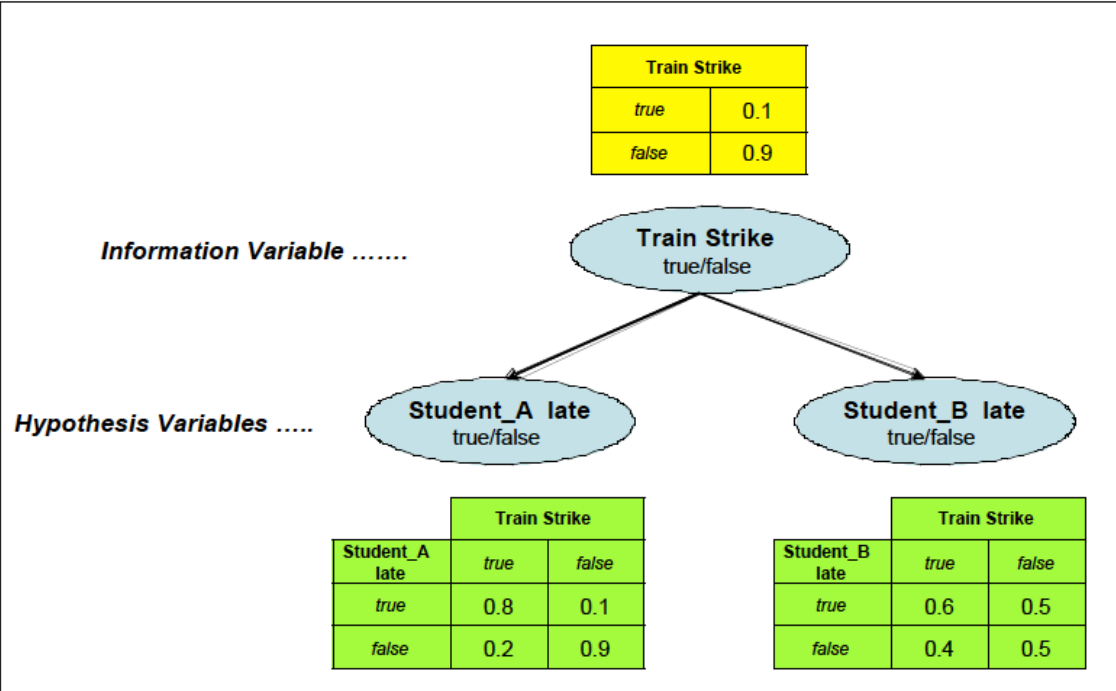


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Unconditional (“marginal”) probability. We don’t know if there is a train strike.

$$\begin{aligned} P(\text{StudentALate}) &= P(\text{StudentALate} \mid \text{TrainStrike})P(\text{TrainStrike}) \\ &+ P(\text{StudentALate} \mid \neg \text{TrainStrike})P(\neg \text{TrainStrike}) \\ &= 0.8 \times 0.1 + 0.2 \times 0.9 = 0.17 \end{aligned}$$

$$\begin{aligned} P(\text{StudentBLate}) &= P(\text{StudentBLate} \mid \text{TrainStrike})P(\text{TrainStrike}) \\ &+ P(\text{StudentBLate} \mid \neg \text{TrainStrike})P(\neg \text{TrainStrike}) \\ &= 0.6 \times 0.1 + 0.5 \times 0.9 = 0.51 \end{aligned}$$

Now, suppose we know that there is a train strike. How does this revise the probability that the students are late?

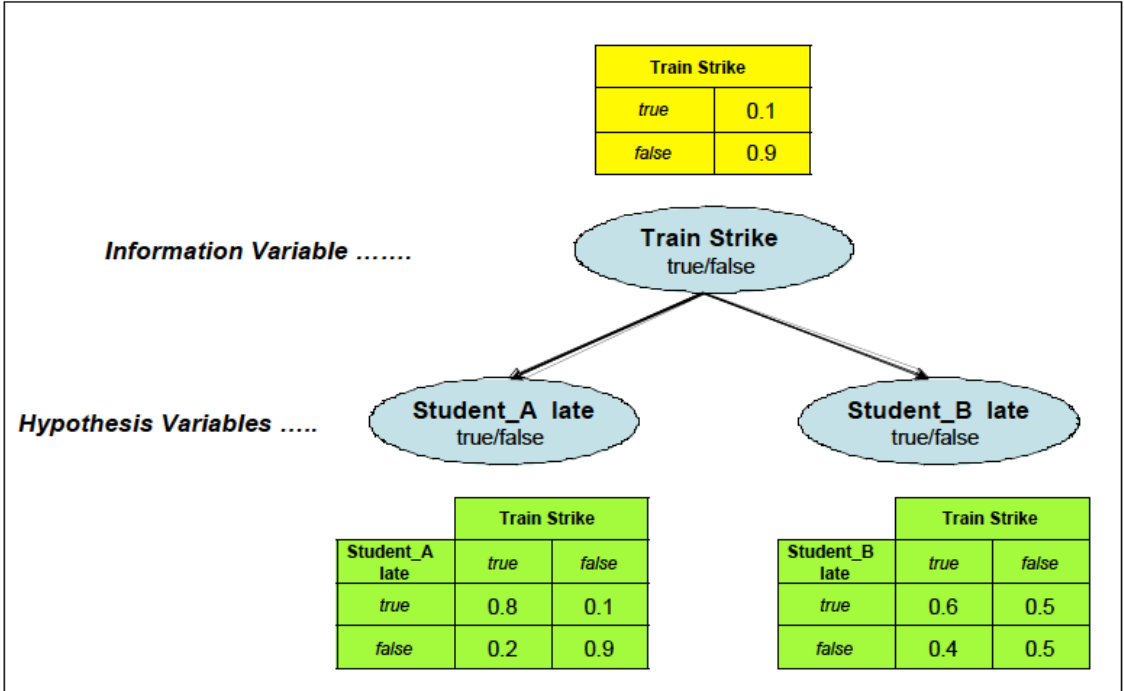


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Now, suppose we know that there is a train strike. How does this revise the probability that the students are late?

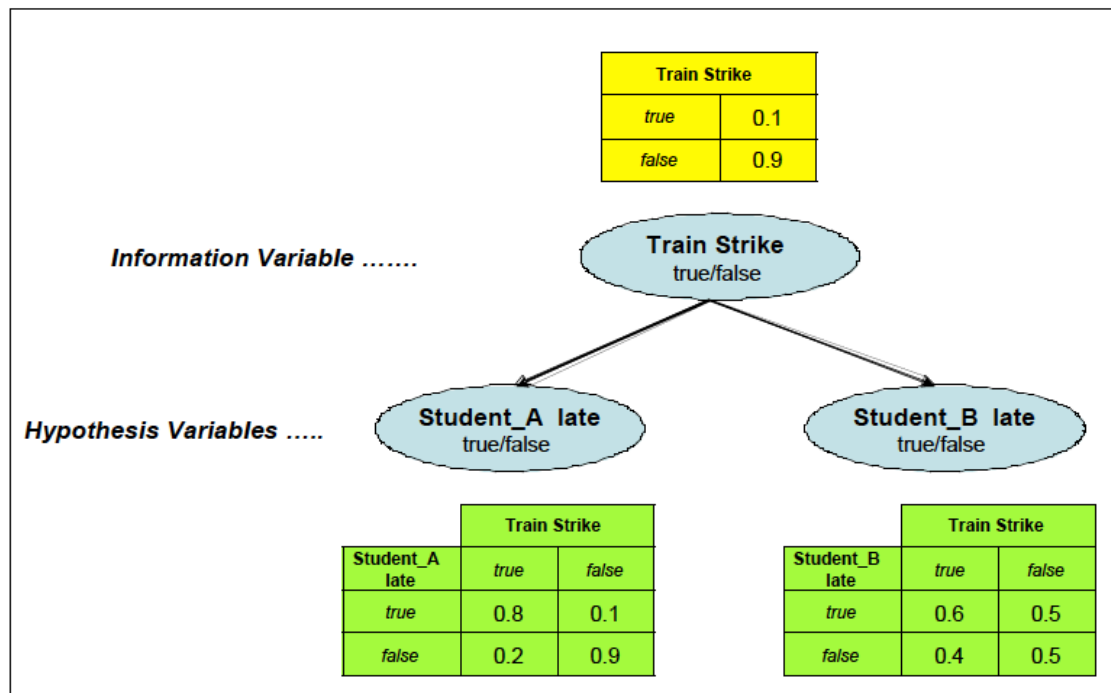


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Evidence: There is a train strike.

$$P(\text{StudentALate}) = 0.8$$

$$P(\text{StudentBLate}) = 0.6$$

Now, suppose we know that Student A is late.

How does this revise the probability that there is a train strike?

How does this revise the probability that Student B is late?

Notion of “belief propagation”.

Evidence: Student A is late.

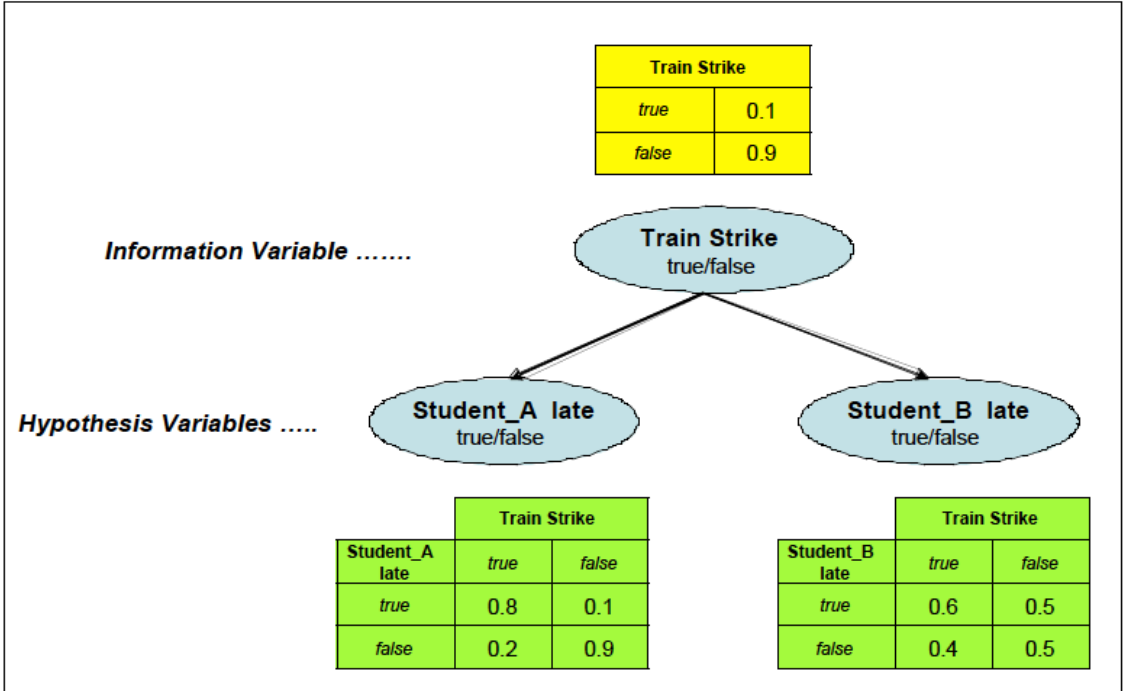


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Now, suppose we know that Student A is late.

How does this revise the probability that there is a train strike?

How does this revise the probability that Student B is late?

Notion of “belief propagation”.

Evidence: Student A is late.

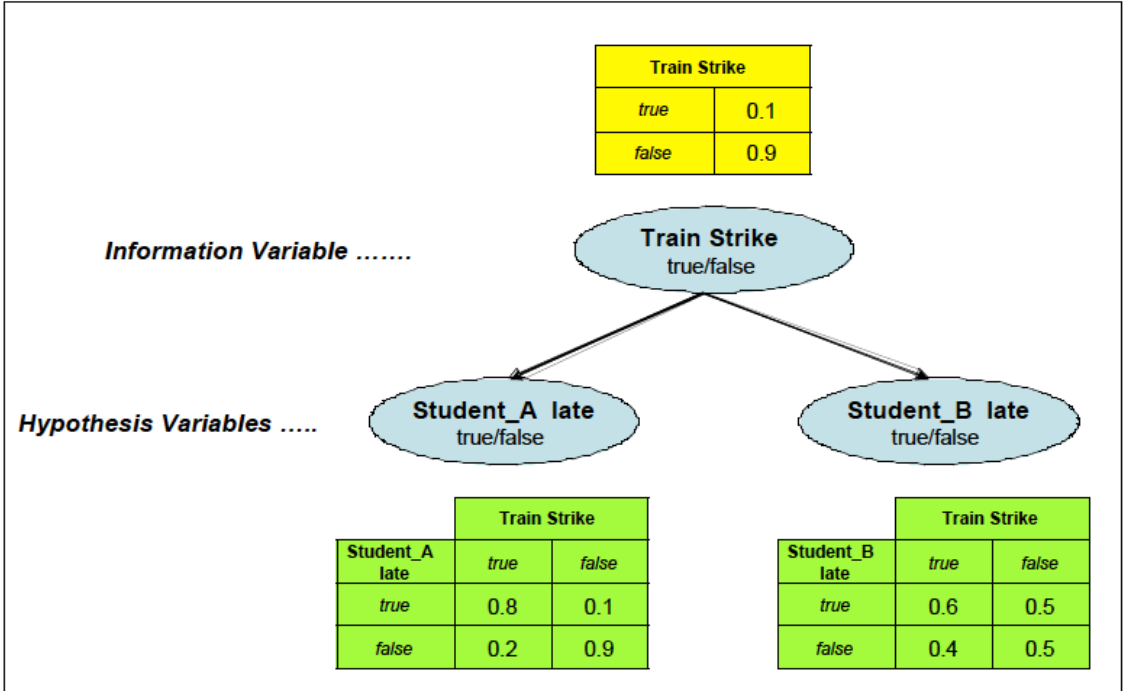


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Now, suppose we know that Student A is late.

How does this revise the probability that there is a train strike?

How does this revise the probability that Student B is late?

Notion of “belief propagation”.

Evidence: Student A is late.

$$\begin{aligned}
 P(\text{TrainStrike} \mid \text{StudentALate}) &= \frac{P(\text{StudentALate} \mid \text{TrainStrike})P(\text{TrainStrike})}{P(\text{StudentALate})} \quad \text{by Bayes Theorem} \\
 &= \frac{0.8 \times 0.1}{0.17} = 0.47
 \end{aligned}$$

$$\begin{aligned}
 P(\text{StudentBLate}) &= P(\text{StudentBLate} \mid \text{TrainStrike})P(\text{TrainStrike}) \\
 &\quad + P(\text{StudentBLate} \mid \neg \text{TrainStrike})P(\neg \text{TrainStrike}) \\
 &= 0.6 \times 0.47 + 0.5 \times 0.53 = 0.55
 \end{aligned}$$

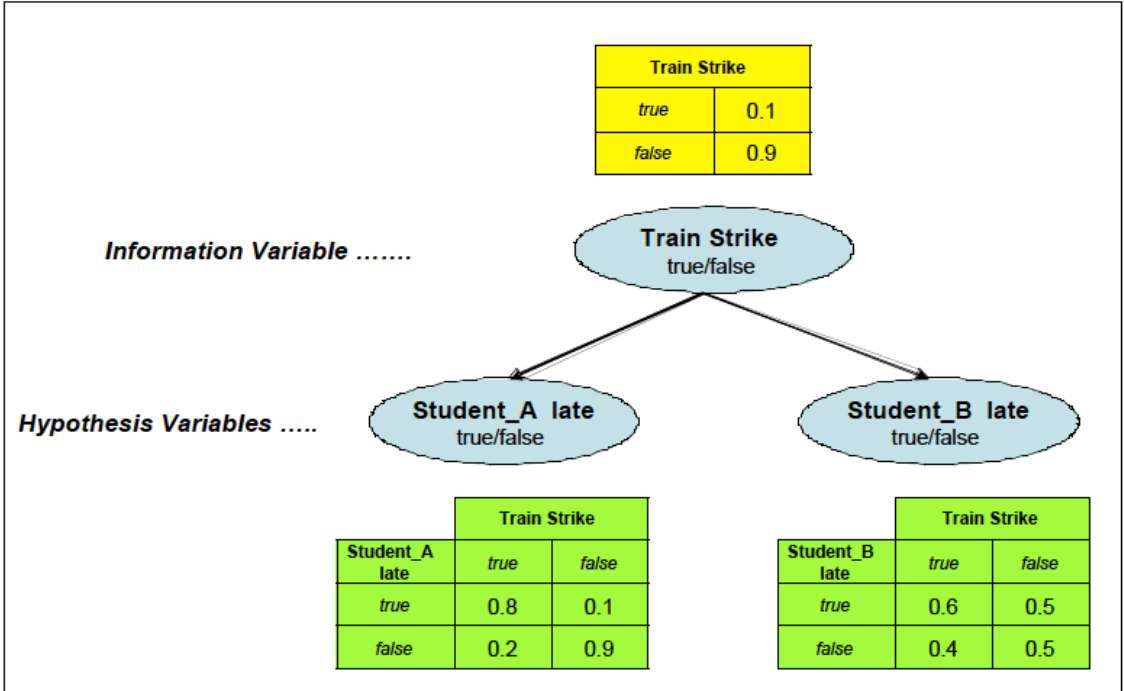
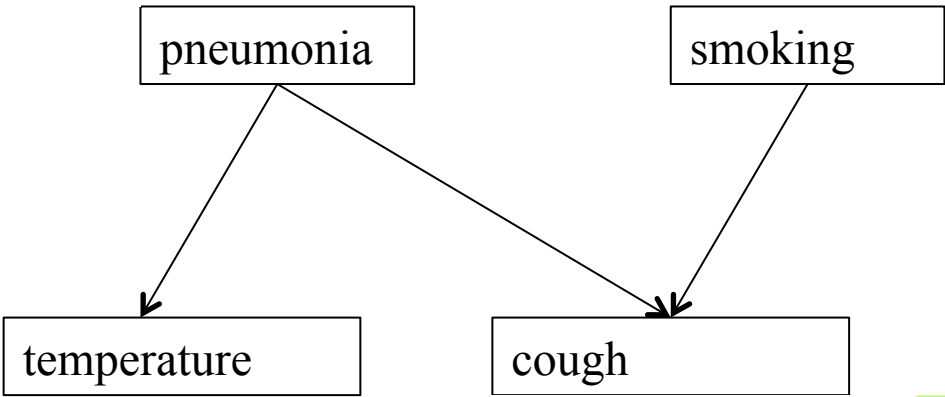


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Another example from the reading:

pneumonia	
true	0.1
false	0.9

smoking	
yes	0.2
no	0.8



	temperature	
pneumonia	yes	no
yes	0.9	0.1
no	0.2	0.8

		cough	
pneumonia	smoking	true	false
true	yes	0.95	0.05
true	no	0.8	0.2
false	yes	0.6	0.4
false	no	0.05	0.95

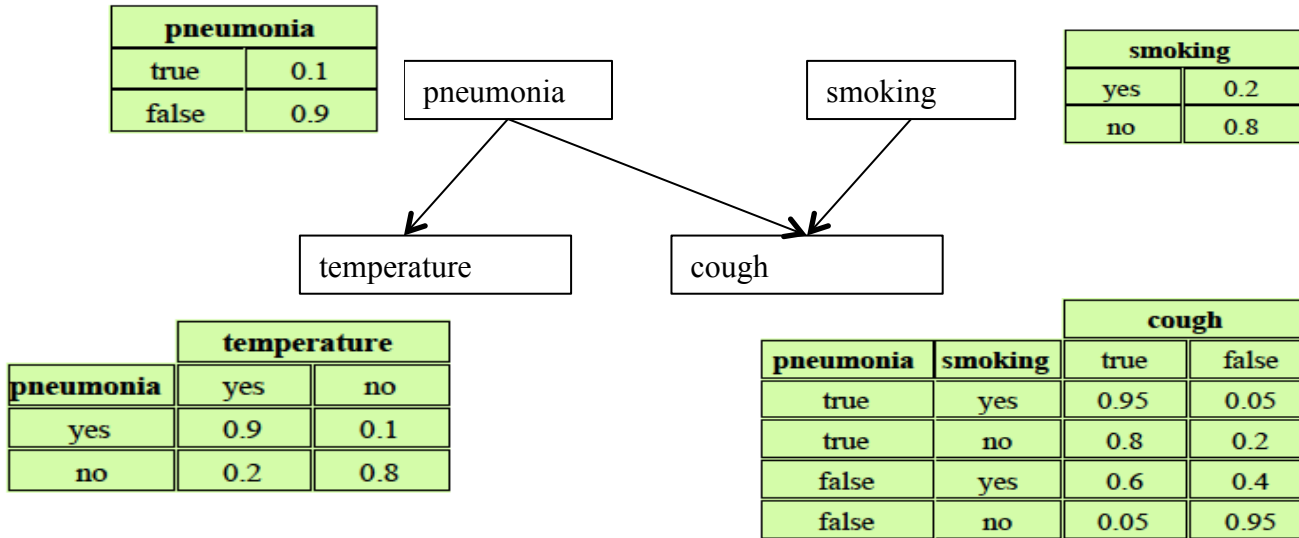
What is $P(\textit{cough})$?

In-class exercises

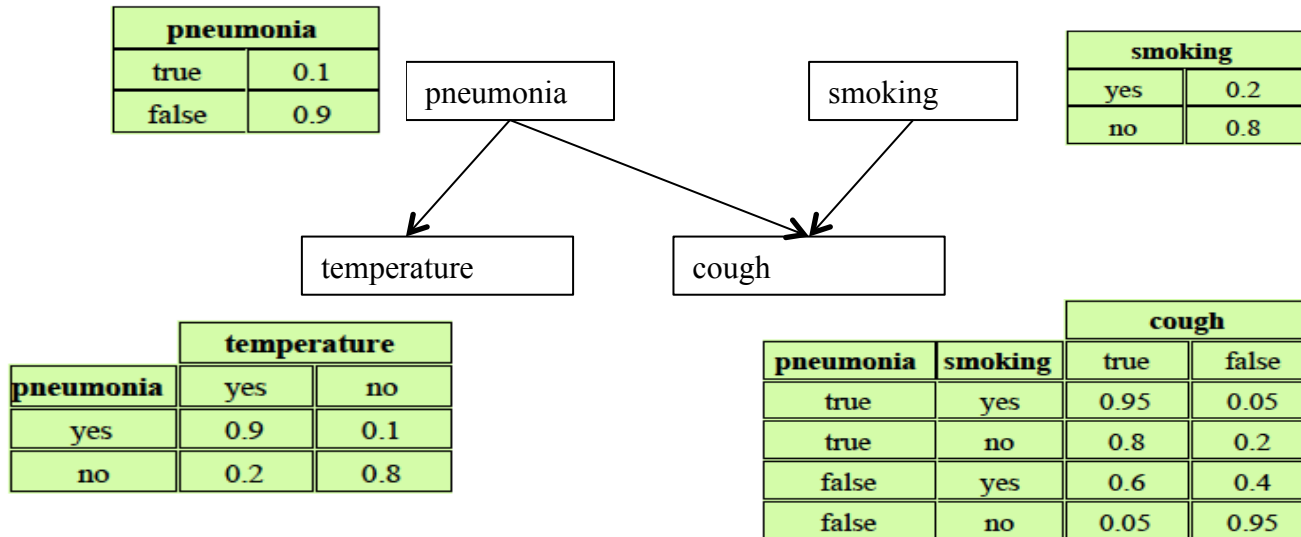
Three types of inference

- **Diagnostic:** Use evidence of an effect to infer probability of a cause.
 - E.g., **Evidence:** *cough=true*. What is $P(\text{pneumonia} \mid \text{cough})$?
- **Causal inference:** Use evidence of a cause to infer probability of an effect
 - E.g., **Evidence:** *pneumonia=true*. What is $P(\text{cough} \mid \text{pneumonia})$?
- **Inter-causal inference:** “Explain away” potentially competing causes of a shared effect.
 - E.g., **Evidence:** *smoking=true*. What is $P(\text{pneumonia} \mid \text{cough and smoking})$?

- Diagnostic: **Evidence:** $cough=true$. What is $P(pneumonia \mid cough)$?

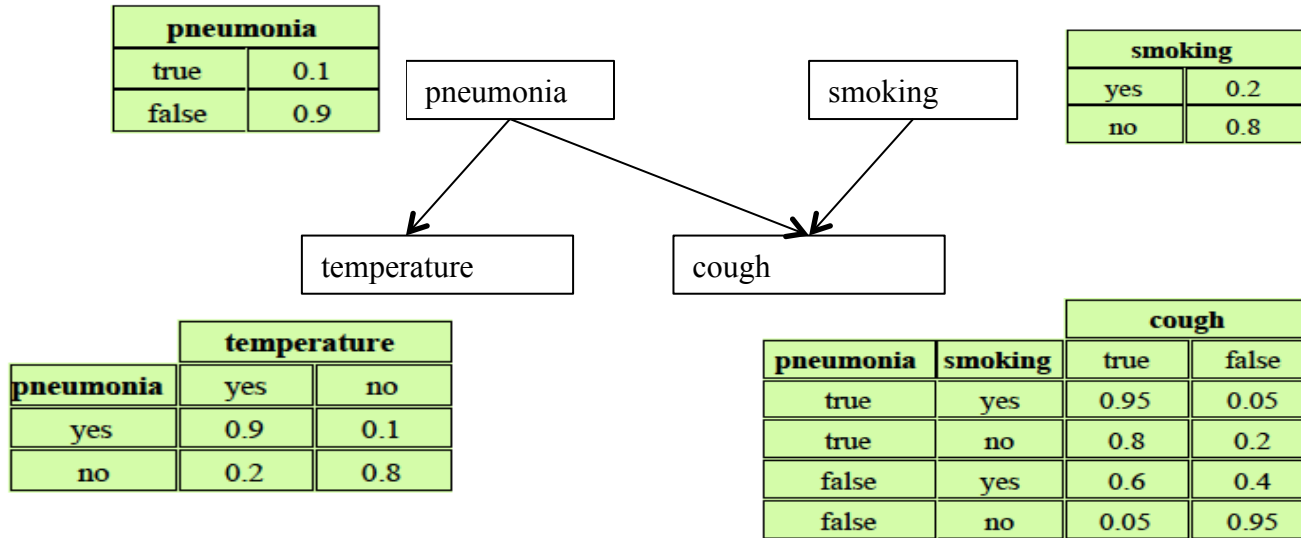


- Diagnostic: **Evidence:** *cough=true*. What is $P(\text{pneumonia} \mid \text{cough})$?

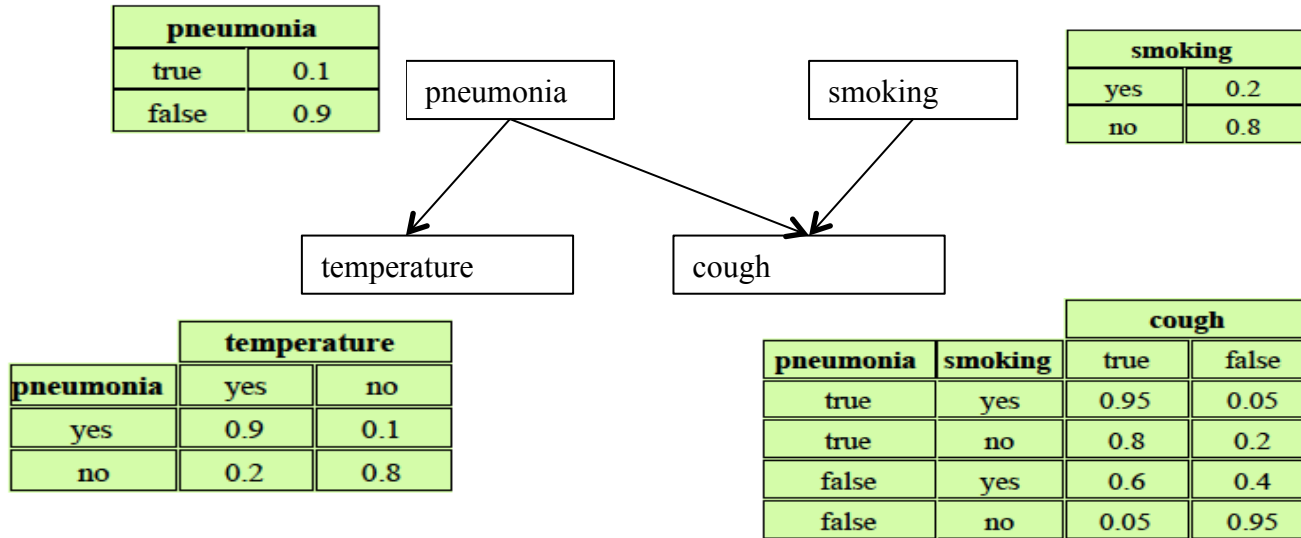


$$\begin{aligned}
 P(\text{pneumonia} \mid \text{cough}) &= \frac{P(\text{cough} \mid \text{pneumonia})P(\text{pneumonia})}{P(\text{cough})} \\
 &= \frac{[P(\text{cough} \mid \text{pneumonia}, \text{smoking})P(\text{smoking}) + P(\text{cough} \mid \text{pneumonia}, \neg \text{smoking})P(\neg \text{smoking})]P(\text{pneumonia})}{P(\text{cough})} \\
 &= \frac{[(.95)(.2) + (.8)(.8)](.1)}{P(\text{cough})} = \frac{.083}{P(\text{cough})} \\
 &= \frac{.083}{P(\text{cough})} = \frac{.083}{.227} = .366
 \end{aligned}$$

- **Causal: Evidence:** *pneumonia=true*. What is $P(\text{cough} \mid \text{pneumonia})$?

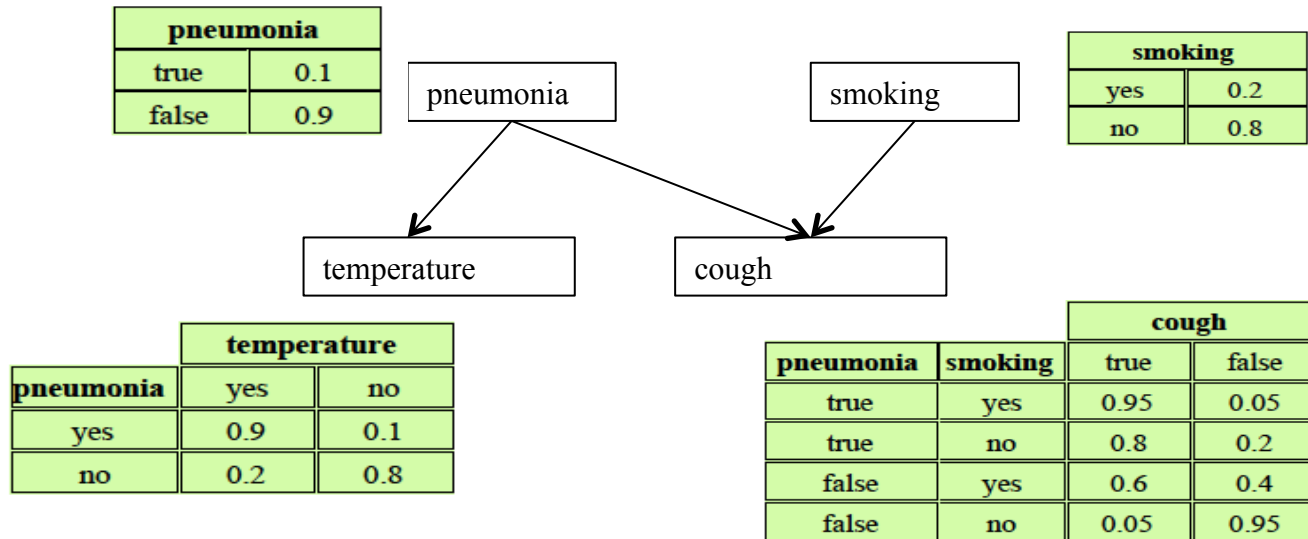


- Causal: Evidence:** *pneumonia=true*. What is $P(\text{cough} \mid \text{pneumonia})$?



$$\begin{aligned}
 P(\text{cough} \mid \text{pneumonia}) &= P(\text{cough} \mid \text{pneumonia}, \text{smoking})P(\text{smoking}) \\
 &+ P(\text{cough} \mid \text{pneumonia}, \neg \text{smoking})P(\neg \text{smoking}) \\
 &= [(.95)(.2) + (.8)(.8)] = .83
 \end{aligned}$$

- **Inter-causal:** Evidence: *smoking=true*. What is $P(\text{pneumonia} \mid \text{cough and smoking})$?



$$P(pneumonia | cough \wedge smoking) = \frac{P(cough \wedge smoking | pneumonia)P(pneumonia)}{P(cough \wedge smoking)}$$

$$= \frac{P(cough \wedge smoking \wedge pneumonia)}{P(pneumonia)} \frac{P(pneumonia)}{P(cough \wedge smoking)}$$

$$= \frac{P(cough \wedge smoking \wedge pneumonia)}{P(cough \wedge smoking)} = \frac{P(cough | pneumonia, smoking)P(smoking)P(pneumonia)}{P(cough | smoking)P(smoking)}$$

$$= \frac{(.95)(.2)(.1)}{[P(cough | smoking, pneumonia)P(pneumonia) + P(cough | smoking, \neg pneumonia)P(\neg pneumonia)]P(smoking)}$$

$$= \frac{.019}{[(.95)(.1) + (.6)(.9)](.2)} = .15$$

“Explaining away”

Math we used:

- **Definition of conditional probability:**

$$P(X_1 | X_2) = \frac{P(X_1 \wedge X_2)}{P(X_2)}$$

- **Bayes Theorem**

$$P(X_1 | X_2) = \frac{P(X_2 | X_1)P(X_1)}{P(X_2)}$$

- **Unconditional (marginal) probability**

If X_1 depends on X_2 then

$$P(X_1) = P(X_1 | X_2)P(X_2) + P(X_1 | \neg X_2)P(\neg X_2)$$

- **Probability inference in Bayesian networks:**

$$P(X_1, \dots, X_d) = P(X_1 \wedge \dots \wedge X_d) = \prod_{i=1}^d P(X_i | \text{parents}(X_i))$$

Complexity of Bayesian Networks

For n random Boolean variables:

- Full joint probability distribution: 2^n entries
- Bayesian network with at most k parents per node:
 - Each conditional probability table: at most 2^k entries
 - Entire network: $n 2^k$ entries

What are the advantages of Bayesian networks?

- Intuitive, concise representation of joint probability distribution (i.e., conditional dependencies) of a set of random variables.
- Represents “beliefs and knowledge” about a particular class of situations.
- Efficient (approximate) inference algorithms
- Efficient, effective learning algorithms

Issues in Bayesian Networks

- Building / learning network topology
- Assigning / learning conditional probability tables
- Approximate inference via sampling

Real-World Example:

The Lumière Project at Microsoft Research

- Bayesian network approach to answering user queries about Microsoft Office.
- *“At the time we initiated our project in Bayesian information retrieval, managers in the Office division were finding that users were having difficulty finding assistance efficiently.”*
- *“As an example, users working with the Excel spreadsheet might have required assistance with formatting “a graph”. Unfortunately, Excel has no knowledge about the common term, “graph,” and only considered in its keyword indexing the term “chart”.*

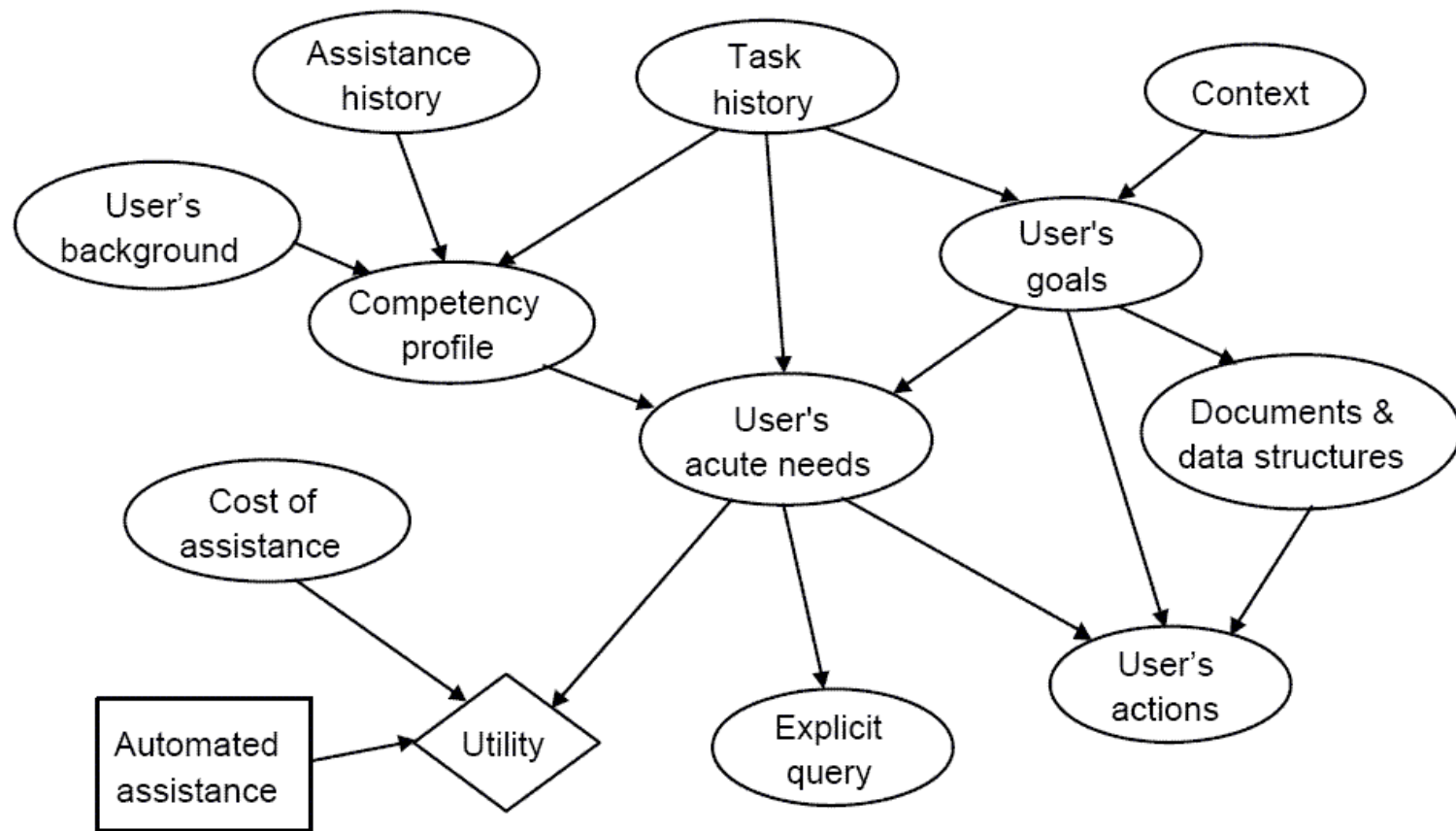


Figure 1: An influence diagram for providing intelligent assistance given uncertainty in a user's background, goals, and competency in working with a software application.

- Networks were developed by experts from user modeling studies.

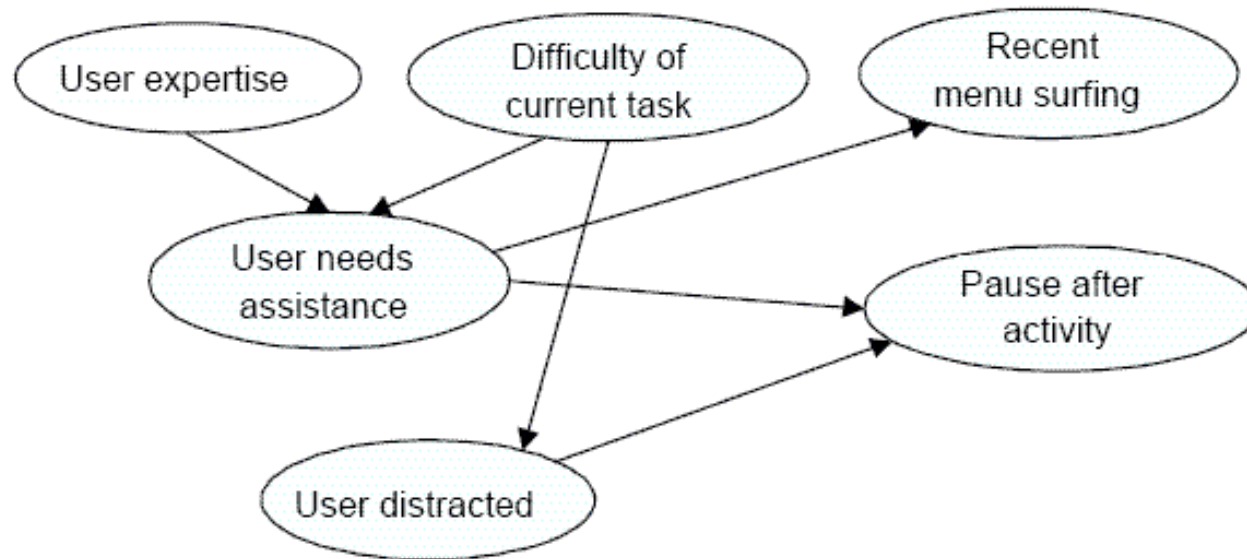


Figure 2: A portion of a Bayesian user model for inferring the likelihood that a user needs assistance, considering profile information as well as observations of recent activity.

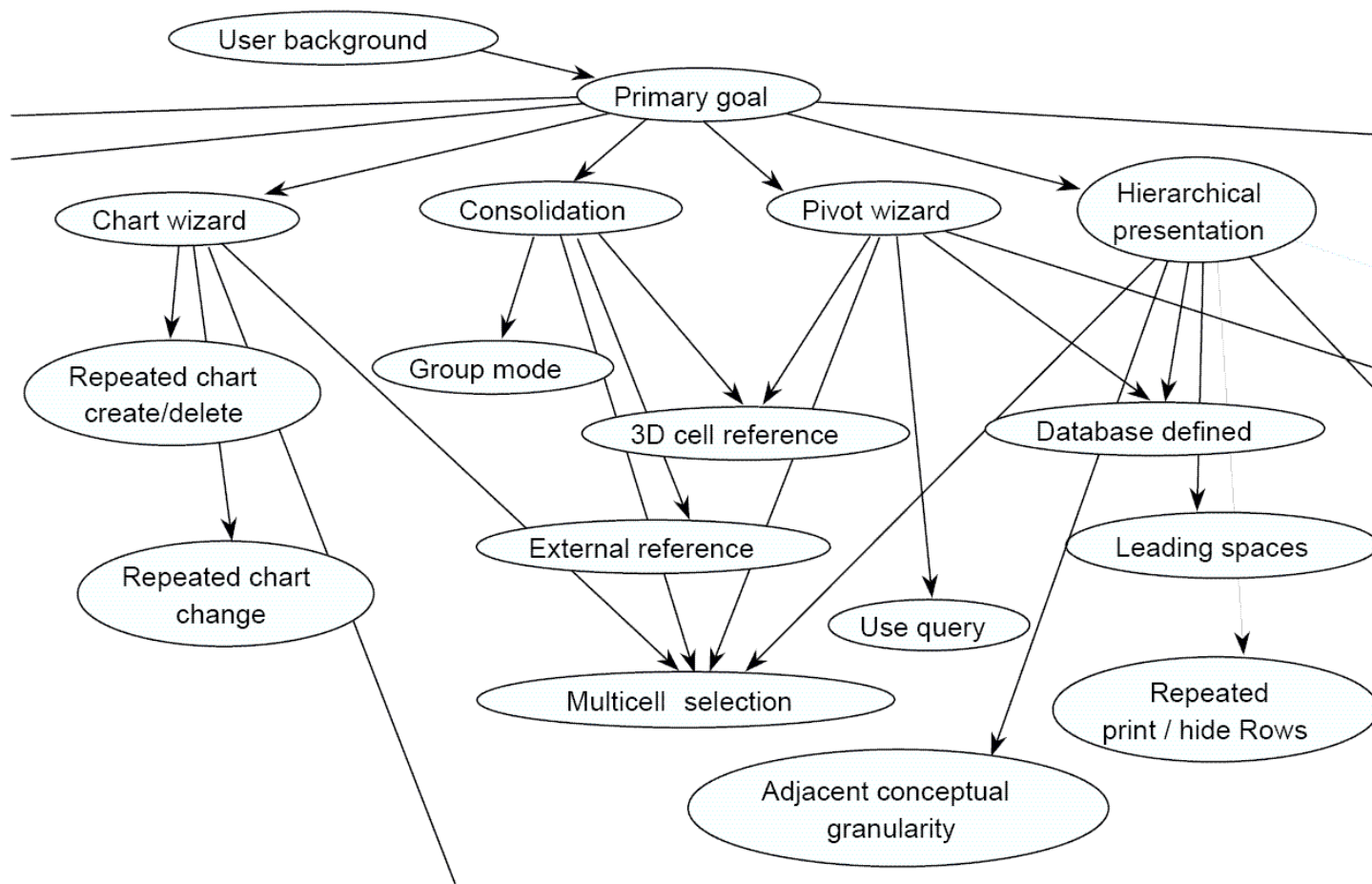


Figure 3: Partial structure of an early formulation of a Bayesian user model for inferring a user's needs for the Excel spreadsheet application.

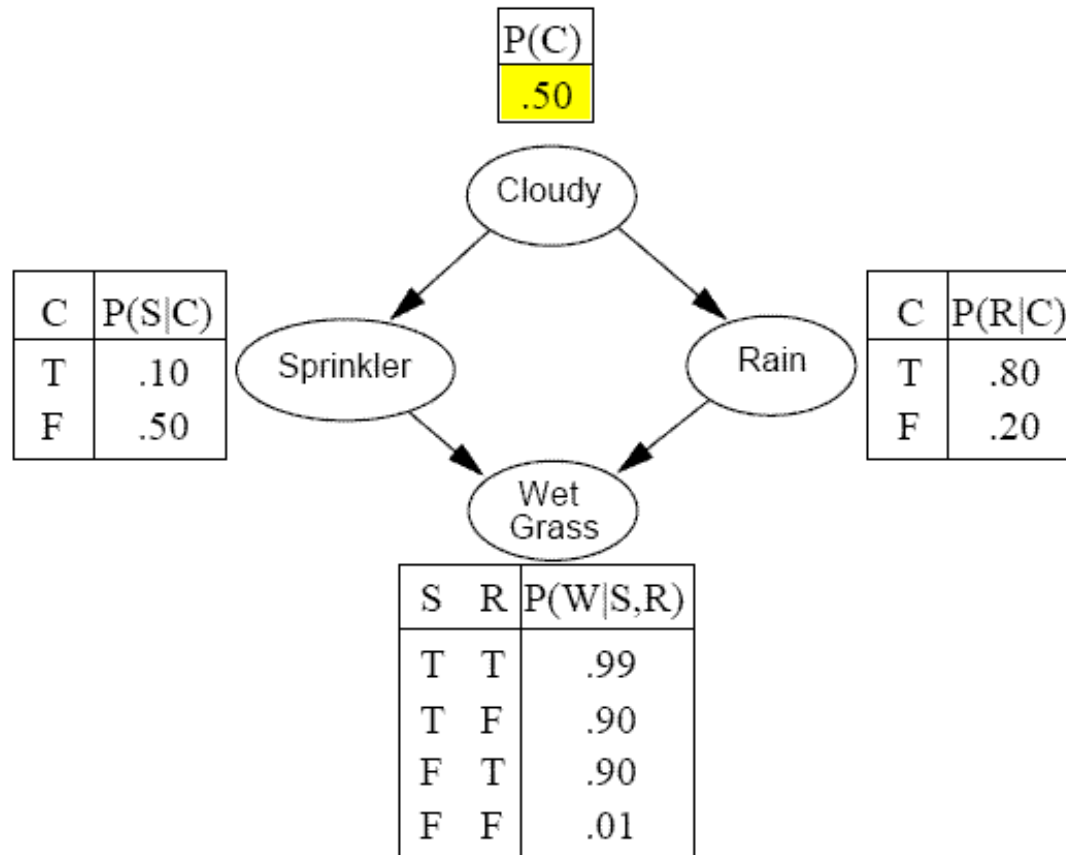
- Offspring of project was *Office Assistant* in Office 97, otherwise known as “clippie”.

<http://www.youtube.com/watch?v=bt-JXQS0zYc>

The famous “sprinkler” example

(J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, 1988)

Example



Recall rule for inference in Bayesian networks:

$$P((X_1 = x_1) \wedge \dots \wedge (X_n = x_n))$$

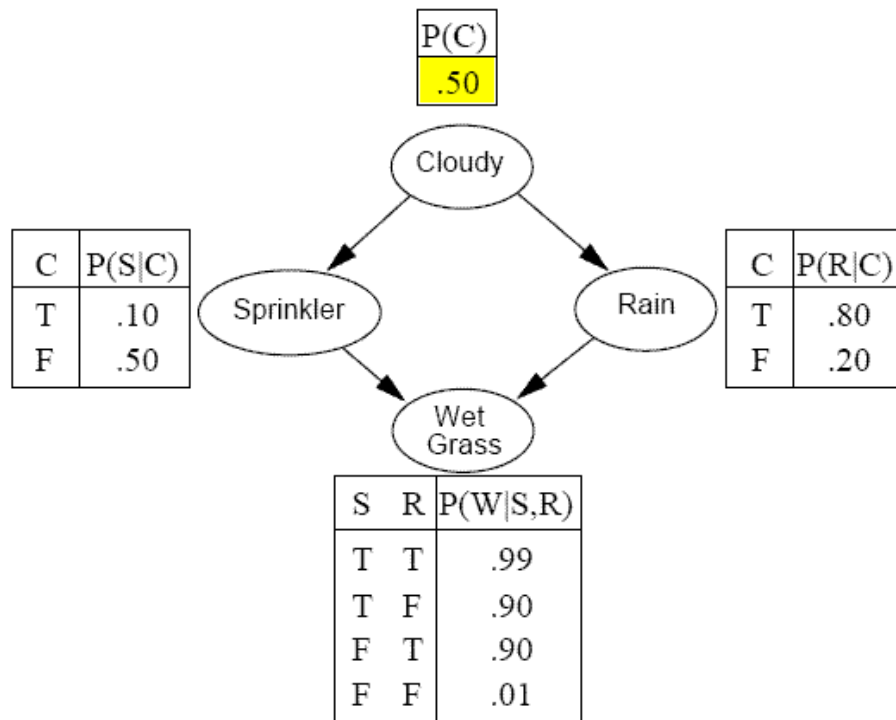
$$= \prod_{i=1}^n P((X_i = x_i) \mid \text{parents}(X_i))$$

Example: What is $P(C, R, \neg S, W)$?

In-Class Exercise

More Exercises

Example



What is $P(\text{Cloudy} | \text{Sprinkler})$?

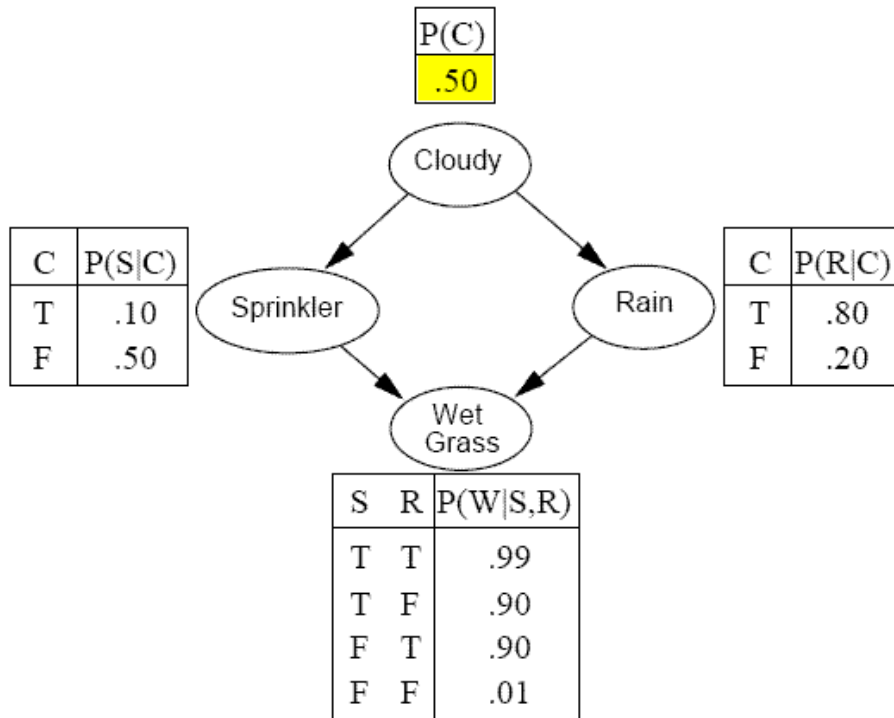
What is $P(\text{Cloudy} | \text{Rain})$?

What is $P(\text{Cloudy} | \text{Wet Grass})$?

More Exercises

Example

What is $P(\text{Cloudy} | \text{Sprinkler})$?

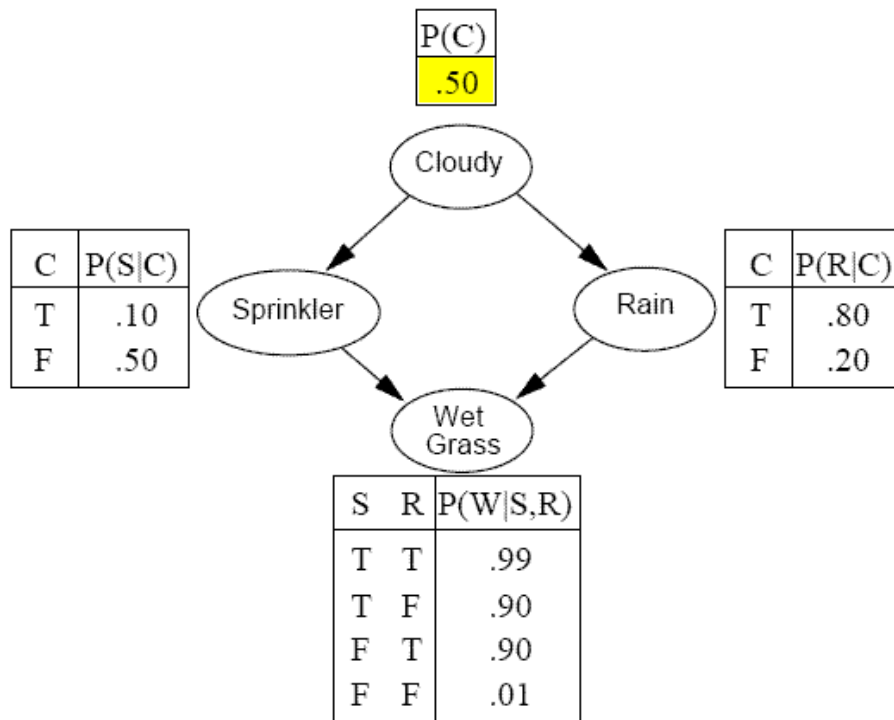


$$\begin{aligned}
 P(C|S) &= \frac{P(S|C)P(C)}{P(S)} \\
 &= \frac{P(S|C)P(C)}{P(S|C)P(C) + P(S|\neg C)P(\neg C)} \\
 &= \frac{(.1)(.5)}{(.1)(.5) + (.5)(.5)} \\
 &= \frac{(.05)}{.3} = .17
 \end{aligned}$$

More Exercises

Example

What is $P(\text{Cloudy} | \text{Rain})$?

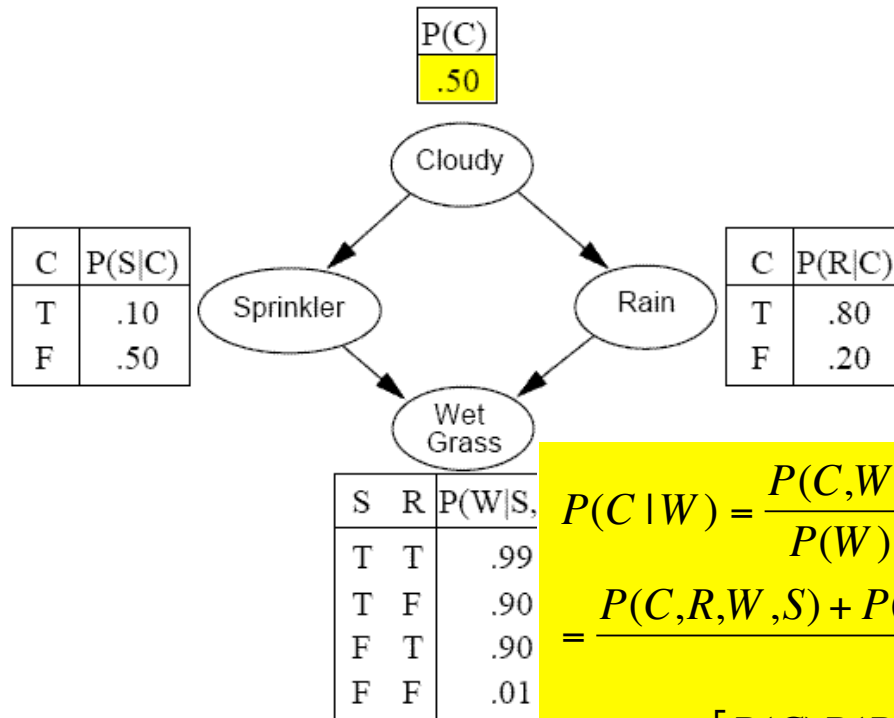


$$\begin{aligned}
 P(C | R) &= \frac{P(R | C)P(C)}{P(R)} \\
 &= \frac{P(R | C)P(C)}{P(R | C)P(C) + P(R | \neg C)P(\neg C)} \\
 &= \frac{(.8)(.5)}{(.8)(.5) + (.2)(.5)} \\
 &= \frac{.4}{.5} = .8
 \end{aligned}$$

More Exercises

Example

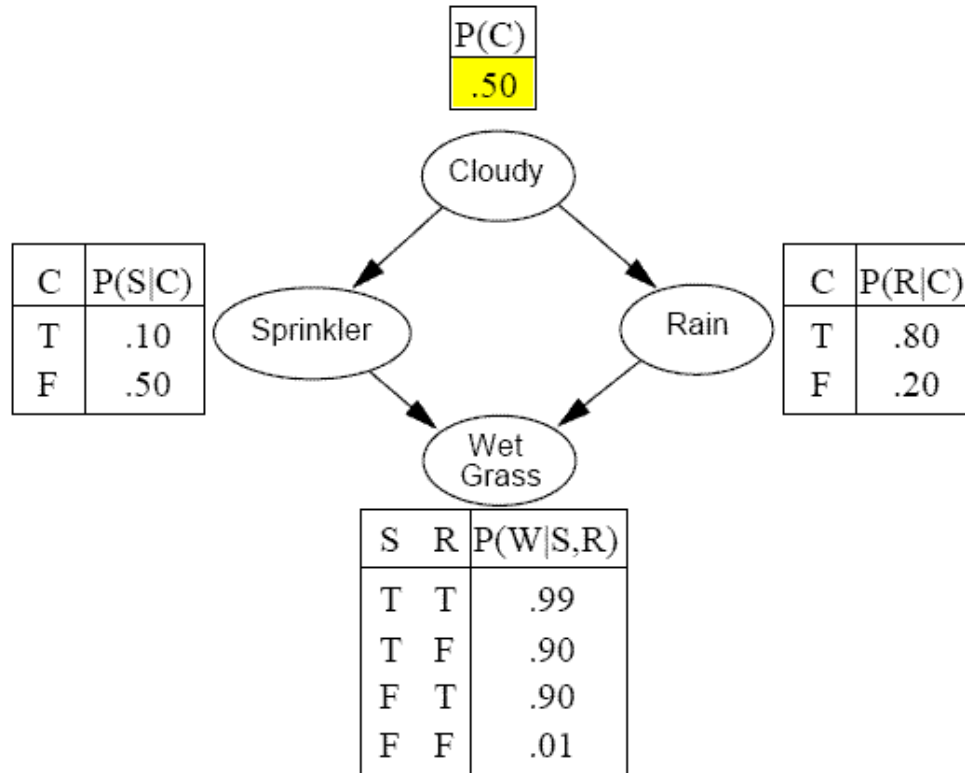
What is $P(\text{Cloudy} | \text{Wet Grass})$?



$$\begin{aligned}
 P(C | W) &= \frac{P(C, W)}{P(W)} \\
 &= \frac{P(C, R, W, S) + P(C, R, W, \neg S) + P(C, \neg R, W, S) + P(C, \neg R, W, \neg S)}{P(W)} \\
 &= \frac{1}{P(W)} \left[\begin{aligned} &P(C)P(R|C)P(W | R, S)P(S|C) \\ &+ P(C)P(R|C)P(W | R, \neg S)P(\neg S|C) \\ &+ P(C)P(\neg R|C)P(W | R, S)P(S|C) \\ &+ P(C)P(\neg R|C)P(W | \neg R, \neg S)P(\neg S|C) \end{aligned} \right]
 \end{aligned}$$

Exact Inference in Bayesian Networks

Example



General question: What is $P(X|e)$?

Notation convention: upper-case letters refer to random variables;
lower-case letters refer to specific values of those variables

More Exercises

1. Suppose you observe it is cloudy and raining. What is the probability that the grass is wet?
2. Suppose you observe the sprinkler to be on and the grass is wet. What is the probability that it is raining?
3. Suppose you observe that the grass is wet and it is raining. What is the probability that it is cloudy?

General question: Given query variable X and observed evidence variable values \mathbf{e} , what is $P(X \mid \mathbf{e})$?

$$P(X \mid \mathbf{e}) = \frac{P(X, \mathbf{e})}{P(\mathbf{e})} \quad (\text{definition of conditional probability})$$

$$= \alpha P(X, \mathbf{e}) \quad \left(\alpha = \frac{1}{P(\mathbf{e})} \right)$$

$$= \alpha \sum_{\mathbf{y} \in \mathbf{Y}} P(X, \mathbf{e}, \mathbf{y}) \quad (\text{where } \mathbf{Y} \text{ are the non-evidence variables other than } X)$$

$$= \alpha \sum_{\mathbf{y}} \prod_{z \in \{X, \mathbf{e}, \mathbf{y}\}} P(z \mid \text{parents}(z)) \quad (\text{semantics of Bayesian networks})$$

Example: What is $P(C | W, R)$?

$$P(C \mid W, R) = \alpha(P(C, R, W, S) + P(C, R, W, \neg S))$$

$$= \alpha \left(\prod_{z \in \{C, R, W, S\}} P(z \mid \text{parents}(Z)) + \prod_{z \in \{C, R, W, \neg S\}} P(z \mid \text{parents}(Z)) \right)$$

$$= \alpha [P(C)P(R \mid C)P(S \mid C)P(W \mid S, R) + P(C)P(R \mid C)P(\neg S \mid C)P(W \mid \neg S, R)]$$

$$= \alpha [(.5 \times .8 \times .1 \times .99) + (.5 \times .8 \times .9 \times .9)]$$

$$= \alpha(.3636)$$

$$\langle P(C \mid R, W), P(\neg C \mid R, W) \rangle = \alpha \langle .3636, .0945 \rangle$$

$$= \left\langle \frac{.3636}{.3636 + .0945}, \frac{.0945}{.3636 + .0945} \right\rangle$$

$$= \langle .794, .206 \rangle$$

- Worst-case complexity is exponential in n (number of nodes)
- Problem is having to enumerate all possibilities for many variables.

$$\alpha \sum_s [P(c)P(r | c)P(s | c)P(w | s, r)]$$

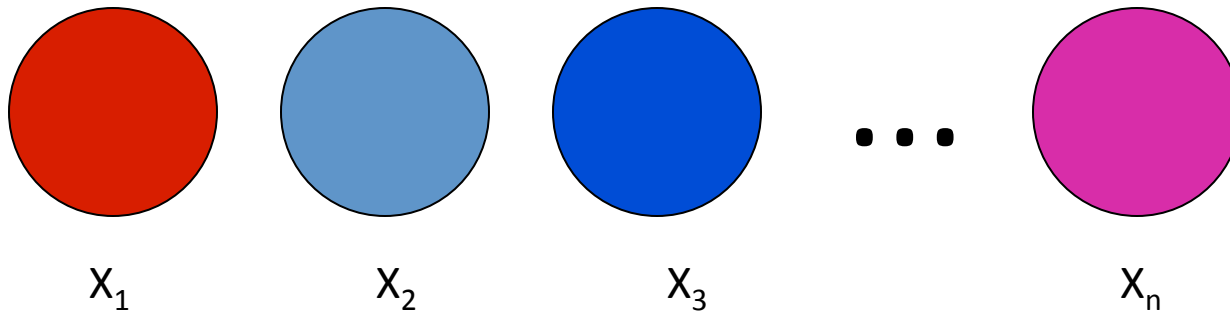
Can reduce computation by computing terms only once and storing for future use.

E.g., “variable elimination algorithm”. (We won’t cover this.)

- In general, however, exact inference in Bayesian networks is too expensive.

Approximate inference in Bayesian networks

Instead of enumerating all possibilities, sample to estimate probabilities.



Direct Sampling

- Suppose we have no evidence, but we want to determine $P(C,S,R,W)$ for all C,S,R,W .
- Direct sampling:
 - Sample each variable in topological order, conditioned on values of parents.
 - I.e., always sample from $P(X_i \mid \text{parents}(X_i))$

Example

1. Sample from $P(\textit{Cloudy})$. Suppose returns *true*.
2. Sample from $P(\textit{Sprinkler} \mid \textit{Cloudy} = \textit{true})$. Suppose returns *false*.
3. Sample from $P(\textit{Rain} \mid \textit{Cloudy} = \textit{true})$. Suppose returns *true*.
4. Sample from $P(\textit{WetGrass} \mid \textit{Sprinkler} = \textit{false}, \textit{Rain} = \textit{true})$. Suppose returns *true*.

Here is the sampled event: $[\textit{true}, \textit{false}, \textit{true}, \textit{true}]$

- Suppose there are N total samples, and let $N_S(x_1, \dots, x_n)$ be the observed frequency of the specific event x_1, \dots, x_n .

$$\lim_{N \rightarrow \infty} \frac{N_S(x_1, \dots, x_n)}{N} = P(x_1, \dots, x_n)$$

$$\frac{N_S(x_1, \dots, x_n)}{N} \approx P(x_1, \dots, x_n)$$

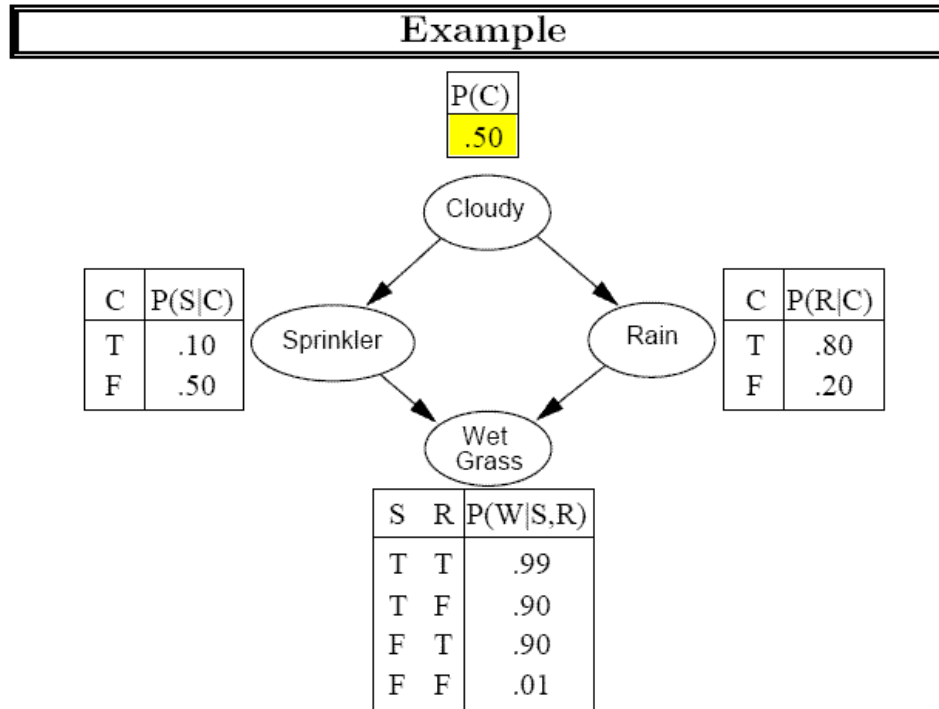
- Suppose N samples, n nodes. Complexity $O(Nn)$.
- Problem 1: Need *lots* of samples to get good probability estimates.
- Problem 2: Many samples are not realistic; low likelihood.

Markov Chain Monte Carlo Sampling

- One of most common methods used in real applications.
- Uses idea of **Markov blanket** of a variable X_i :
 - parents, children, children's other parents
- **Fact:** By construction of Bayesian network, a node is conditionally independent of its non-descendants, given its parents.

What is the Markov Blanket of *Rain*?

What is the Markov blanket of *Wet Grass*?



- **Proposition:** A node X_i is conditionally independent of all other nodes in the network, given its Markov blanket.

Markov Chain Monte Carlo (MCMC) Sampling Algorithm

- Start with random sample from variables, with evidence variables fixed: (x_1, \dots, x_n) . This is the current “state” of the algorithm.
- Next state: Randomly sample value for one non-evidence variable X_i , conditioned on current values in “Markov Blanket” of X_i .

Example

- Query: What is $P(\text{Rain} \mid \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$?
- MCMC:
 - Random sample, with evidence variables fixed:
[Cloudy, Sprinkler, Rain, WetGrass]
= [true, true, false, true]
 - Repeat:
 1. Sample *Cloudy*, given current values of its Markov blanket:
Sprinkler = true, Rain = false. Suppose result is *false*. New state: *[false, true, false, true]*
 2. Sample *Rain*, given current values of its Markov blanket:
Cloudy = false, Sprinkler = true, WetGrass = true. Suppose result is *true*. New state: *[false, true, true, true]*.

- Each sample contributes to estimate for query
 $P(\text{Rain} \mid \text{Sprinkler} = \text{true}, \text{WetGrass} = \text{true})$
- Suppose we perform 100 such samples, 20 with $\text{Rain} = \text{true}$ and 80 with $\text{Rain} = \text{false}$.
- Then answer to the query is
 $\text{Normalize}(\langle 20, 80 \rangle) = \langle .20, .80 \rangle$
- **Claim:** “The sampling process settles into a dynamic equilibrium in which the long-run fraction of time spent in each state is exactly proportional to its posterior probability, given the evidence.”
 - That is: for all variables X_i , the probability of the value x_i of X_i appearing in a sample is equal to $P(x_i \mid \mathbf{e})$.
- **Proof of claim:** Reference on request


Issues in Bayesian Networks

- Building / learning network topology
- Assigning / learning conditional probability tables
- Approximate inference via sampling
- Incorporating temporal aspects (e.g., evidence changes from one time step to the next).

Learning network topology

- Many different approaches, including:
 - Heuristic search, with evaluation based on information theory measures
 - Genetic algorithms
 - Using “meta” Bayesian networks!

Learning conditional probabilities

- In general, random variables are not binary, but real-valued
- Conditional probability tables  conditional probability distributions
- Estimate parameters of these distributions from data
- If data is missing on one or more variables, use “expectation maximization” algorithm

Speech Recognition

- Task: Identify sequence of words uttered by speaker, given acoustic signal.
- Uncertainty introduced by noise, speaker error, variation in pronunciation, homonyms, etc.
- Thus speech recognition is viewed as problem of probabilistic inference.

Speech Recognition

- So far, we've looked at probabilistic reasoning in static environments.
- Speech: Time sequence of “static environments”.
 - Let \mathbf{X} be the “state variables” (i.e., set of non-evidence variables) describing the environment (e.g., *Words* said during time step t)
 - Let \mathbf{E} be the set of evidence variables (e.g., features of acoustic signal).

- The **E** values and **X** joint probability distribution changes over time.

$t_1: \mathbf{X}_1, \mathbf{e}_1$

$t_2: \mathbf{X}_2, \mathbf{e}_2$

etc.

- At each t , we want to compute $P(\text{Words} \mid \mathbf{S})$.
- We know from Bayes rule:

$$P(\text{Words} \mid \mathbf{S}) = \alpha P(\mathbf{S} \mid \text{Words})P(\text{Words})$$

- $P(\mathbf{S} \mid \text{Words})$, for all words, is a previously learned “acoustic model”.
 - E.g. For each word, probability distribution over phones, and for each phone, probability distribution over acoustic signals (which can vary in pitch, speed, volume).
- $P(\text{Words})$, for all words, is the “language model”, which specifies prior probability of each utterance.
 - E.g. “**bigram model**”: probability of each word following each other word.

- Speech recognition typically makes three assumptions:
 1. Process underlying change is itself “stationary”
i.e., state transition probabilities don’t change
 2. Current state \mathbf{X} depends on only a finite history of previous states (“**Markov assumption**”).
 - Markov process of order n : Current state depends only on n previous states.
 3. Values \mathbf{e}_t of evidence variables depend only on current state \mathbf{X}_t . (“**Sensor model**”)

Phones

All human speech is composed from 40-50 **phones**, determined by the configuration of **articulators** (lips, teeth, tongue, vocal cords, air flow)

Form an intermediate level of hidden states between words and signal

⇒ acoustic model = pronunciation model + phone model

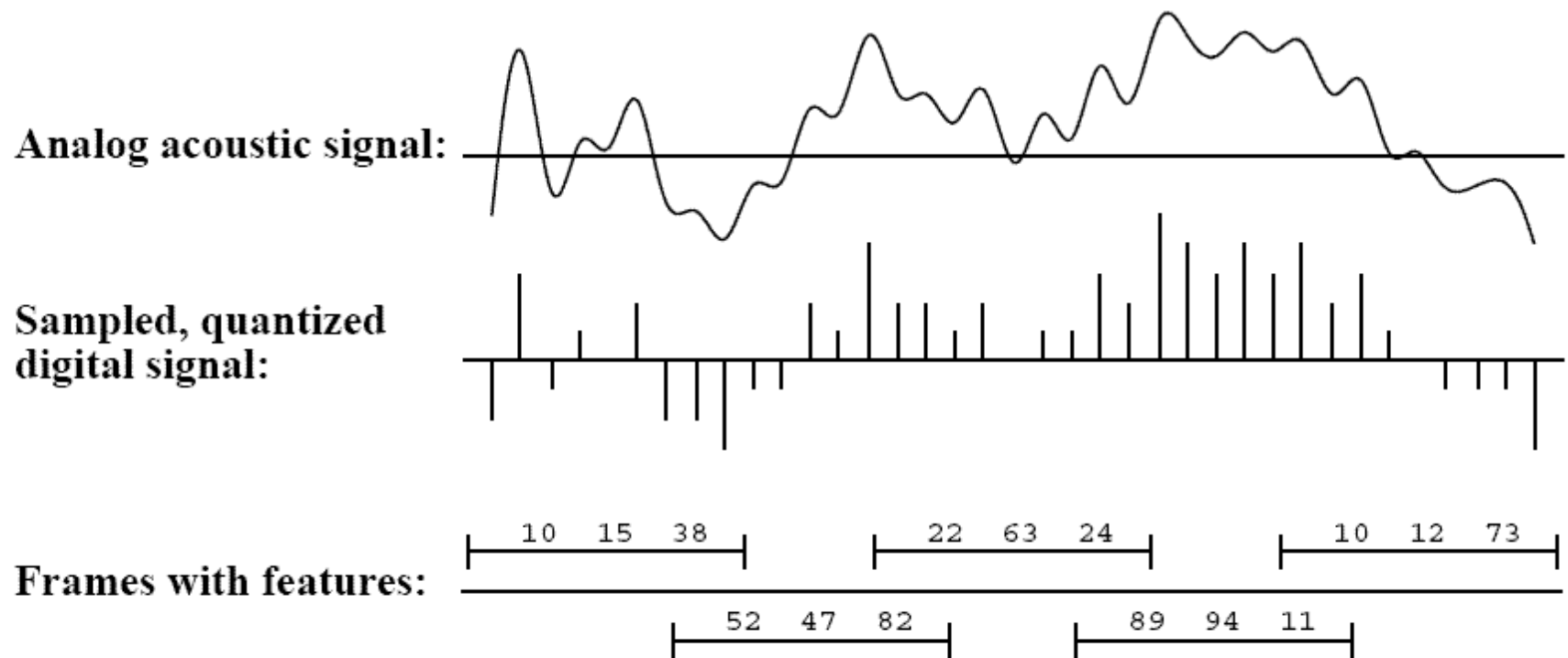
ARPAbet designed for American English

[iy]	<u>b</u> eat	[b]	<u>b</u> et	[p]	<u>p</u> et
[ih]	b <u>i</u> t	[ch]	<u>C</u> het	[r]	<u>r</u> at
[ey]	b <u>e</u> t	[d]	<u>d</u> ebt	[s]	<u>s</u> et
[ao]	<u>b</u> ought	[hh]	<u>h</u> at	[th]	<u>t</u> hick
[ow]	<u>b</u> oat	[hv]	<u>h</u> igh	[dh]	<u>t</u> hat
[er]	B <u>e</u> r	[l]	<u>l</u> et	[w]	<u>w</u> et
[ix]	ros <u>e</u> s	[ng]	s <u>i</u> ng	[en]	butt <u>o</u> n
⋮	⋮	⋮	⋮	⋮	⋮

E.g., “ceiling” is [s iy l ih ng] / [s iy l ix ng] / [s iy l en]

Speech sounds

Raw signal is the microphone displacement as a function of time;
processed into overlapping 30ms **frames**, each described by **features**



Frame features are typically **formants**—peaks in the power spectrum

Hidden Markov Models

- **Markov model:** Given state X_t , what is probability of transitioning to next state X_{t+1} ?

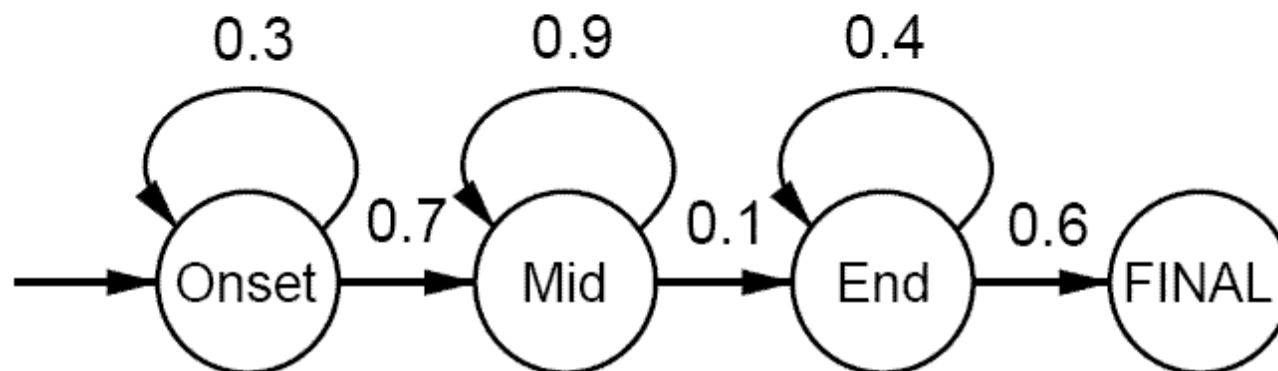
- E.g., word bigram probabilities give

$$P(\text{word}_{t+1} \mid \text{word}_t)$$

- **Hidden Markov model:** There are observable states (e.g., signal S) and “hidden” states (e.g., *Words*). **HMM** represents probabilities of hidden states given observable states.

Phone model example

Phone HMM for [m]:



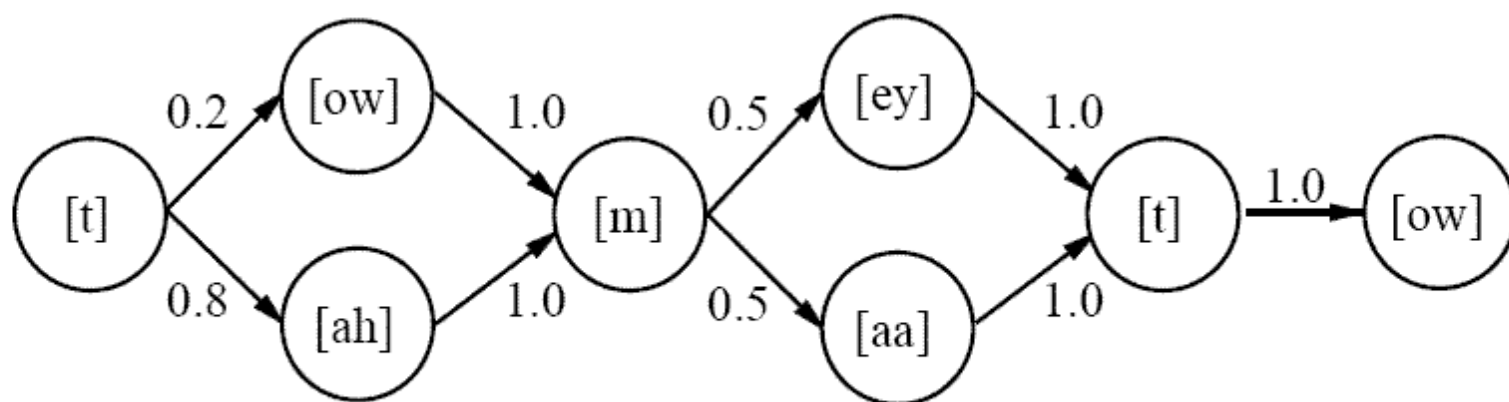
Output probabilities for the phone HMM:

Onset:	Mid:	End:
C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4

Word pronunciation models

Each word is described as a distribution over phone sequences

Distribution represented as an HMM transition model



$$P([touwmeytow] | \text{"tomato"}) = P([touwmaatow] | \text{"tomato"}) = 0.1$$

$$P([tahmeytow] | \text{"tomato"}) = P([tahmaatow] | \text{"tomato"}) = 0.4$$

Structure is created manually, transition probabilities learned from data

Continuous speech

Not just a sequence of isolated-word recognition problems!

- Adjacent words highly correlated
- Sequence of most likely words \neq most likely sequence of words
- Segmentation: there are few gaps in speech
- Cross-word coarticulation—e.g., “next thing”

Continuous speech systems manage 60–80% accuracy on a good day

Example: “I’m firsty, um, can I have something to dwink?”

Language model

Prior probability of a word sequence is given by chain rule:

$$P(w_1 \cdots w_n) = \prod_{i=1}^n P(w_i | w_1 \cdots w_{i-1})$$

Bigram model:

$$P(w_i | w_1 \cdots w_{i-1}) \approx P(w_i | w_{i-1})$$

Train by counting all word pairs in a large text corpus

More sophisticated models (trigrams, grammars, etc.) help a little bit