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*:

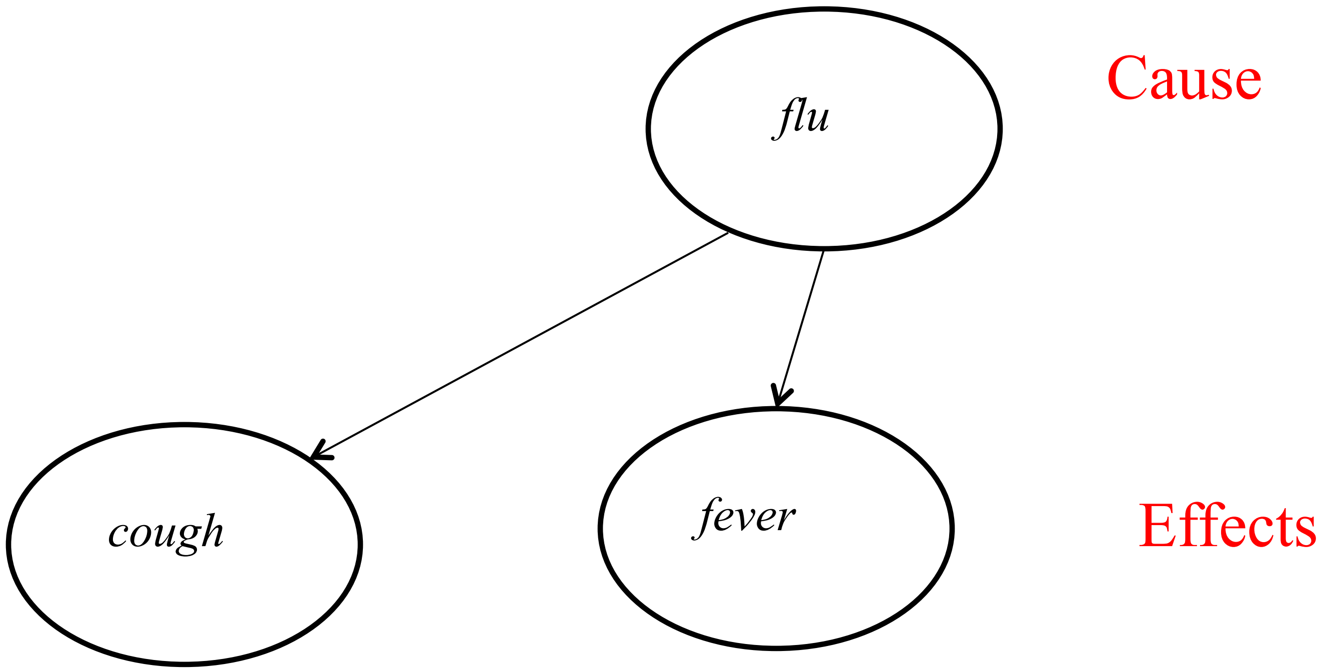
h_1: patient has flu

h_2: patient does not have flu

Data *D*:

$coughing$ = true, $fever$ = true, $smokes$ = true

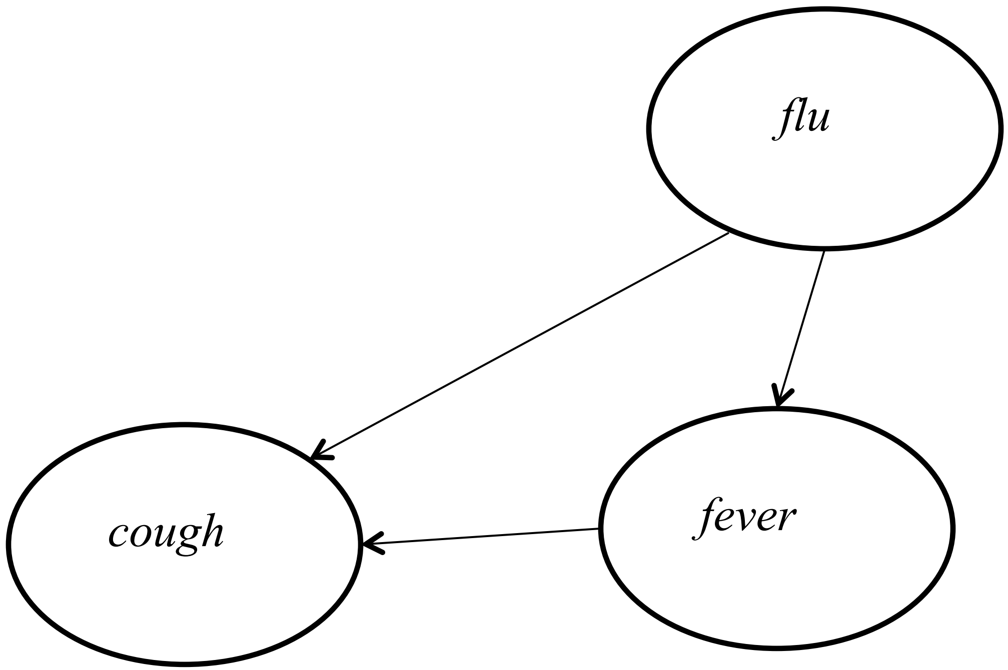
Naive Bayes



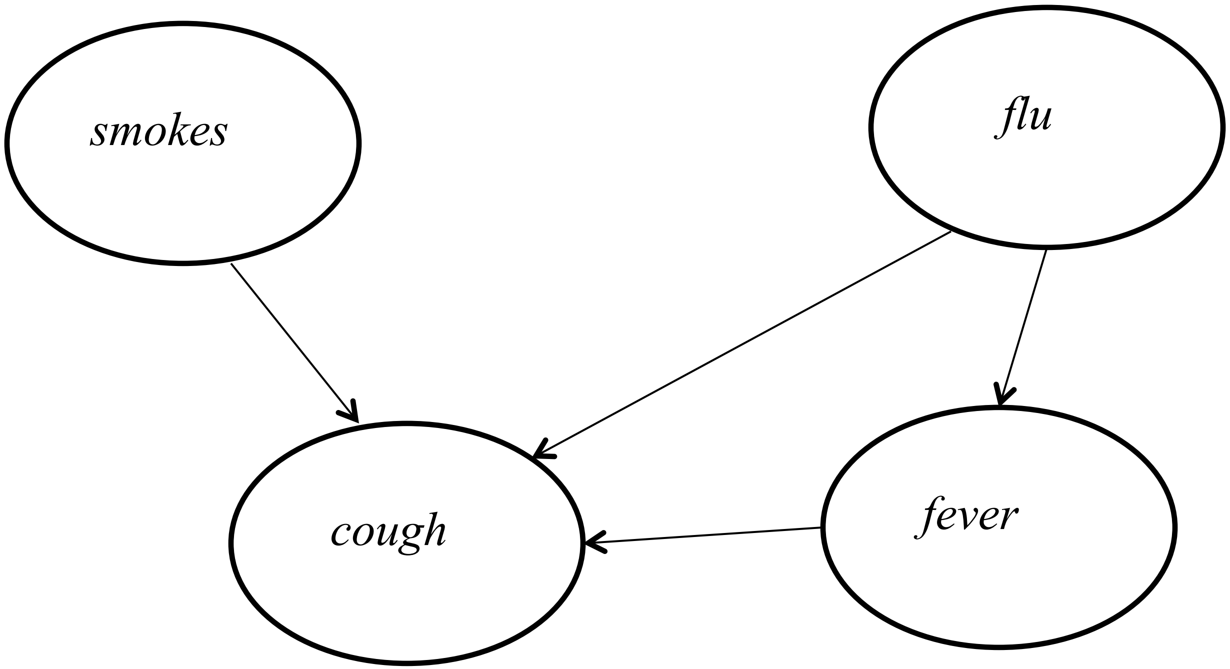
$$h1: P(flu \mid cough, fever) \approx P(flu)P(cough \mid flu)P(fever \mid flu)$$

$$h2: P(\neg flu \mid cough, fever) \approx P(\neg flu)P(cough \mid \neg flu)P(fever \mid \neg flu)$$

What if attributes are not independent?



What if more than one possible cause?



Full joint probability distribution

smokes				
	cough		*¬ cough*	
	Fever	*¬ Fever*	*Fever*	*¬ Fever*
flu	p_1	p_2	p_3	p_4
¬flu	p_5	p_6	p_7	p_8

¬ smokes				
	cough		*¬ cough*	
	fever	*¬ fever*	*fever*	*¬ fever*
flu	p_9	p_{10}	p_{11}	p_{12}
¬flu	p_{13}	p_{14}	p_{15}	p_{16}

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.

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

smokes				
	cough		*¬ cough*	
	Fever	*¬ Fever*	*Fever*	*¬ Fever*
flu	p_1	p_2	p_3	p_4
¬flu	p_5	p_6	p_7	p_8

¬ smokes				
	cough		*¬ cough*	
	fever	*¬ fever*	*fever*	*¬ fever*
flu	p_9	p_{10}	p_{11}	p_{12}
¬flu	p_{13}	p_{14}	p_{15}	p_{16}

For example, what if we had another attribute, "allergies"?

How many probabilities would we need to specify?

Allergy				
smokes				
	cough		*¬ cough*	
	Fever	*¬Fever*	*Fever*	*¬ Fever*
flu	p_1	p_2	p_3	p_4
¬flu	p_5	p_6	p_7	p_8

¬Allergy				
smokes				
	cough		*¬ cough*	
	Fever	*¬Fever*	*Fever*	*¬ Fever*
flu	p_{17}	p_{18}	p_{19}	p_{20}
¬flu	p_{21}	p_{22}	p_{23}	p_{24}

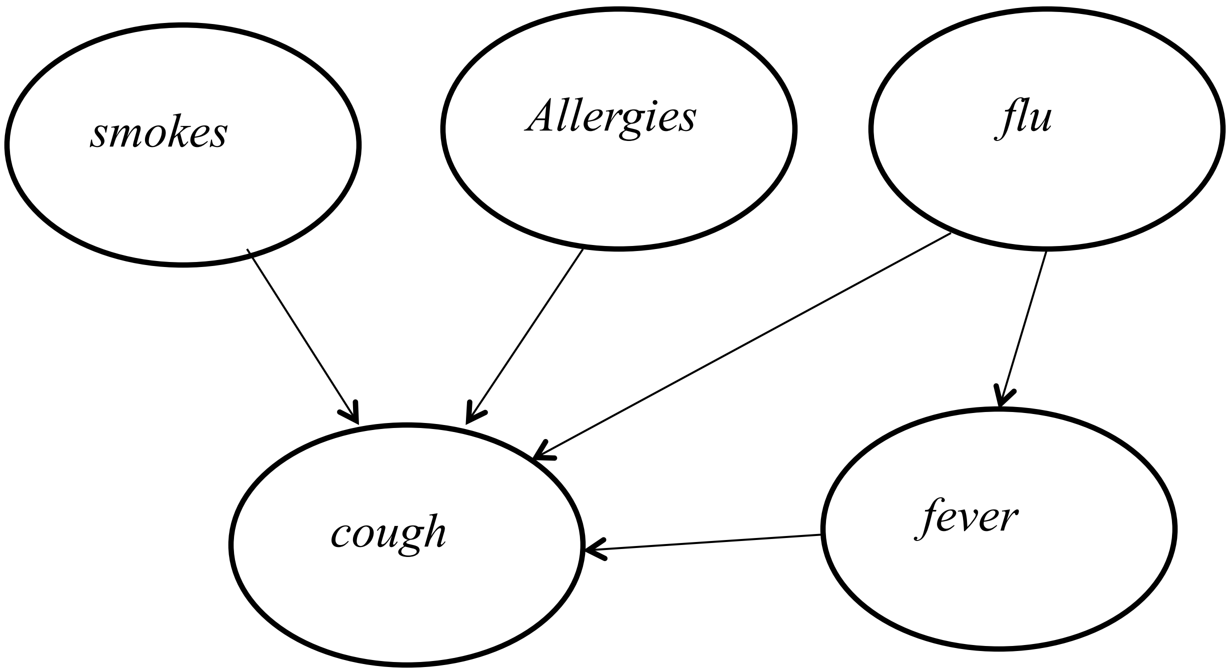
Allergy				
¬ smokes				
	cough		*¬ cough*	
	fever	*¬ fever*	*fever*	*¬ fever*
flu	p_9	p_{10}	p_{11}	p_{12}
¬flu	p_{13}	p_{14}	p_{15}	p_{16}

¬Allergy				
¬ smokes				
	cough		*¬ cough*	
	fever	*¬ fever*	*fever*	*¬ fever*
flu	p_{25}	p_{26}	p_{27}	p_{28}
¬flu	p_{29}	p_{30}	p_{31}	p_{32}

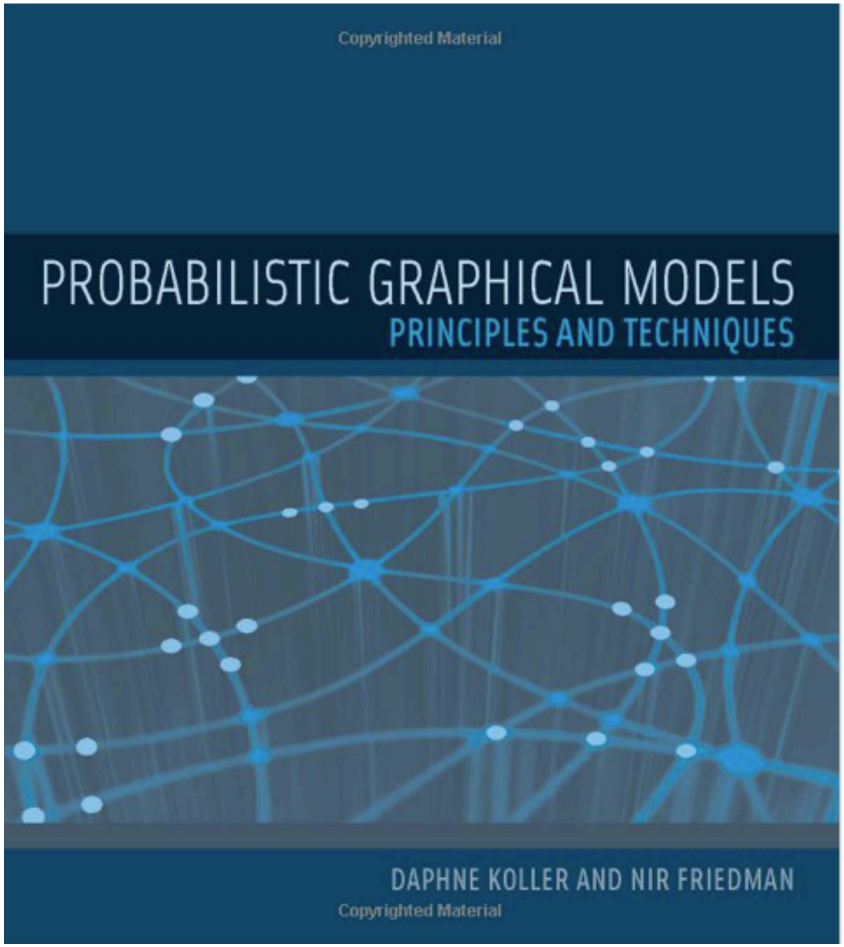
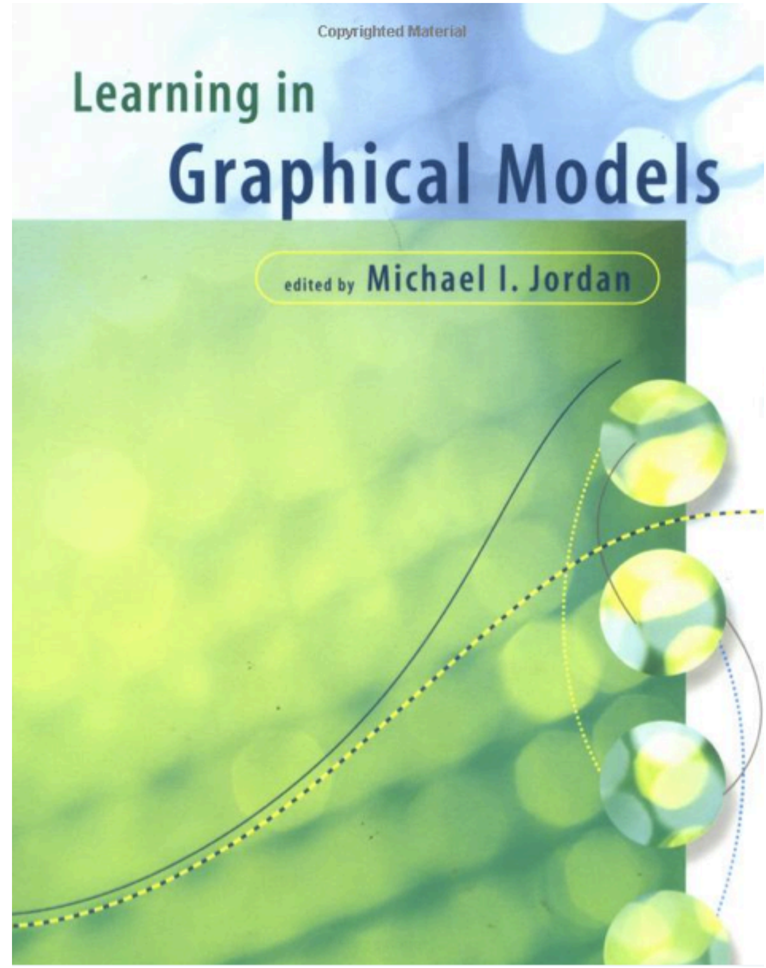
But can reduce this if we know which variables are conditionally independent

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

coursera | Global Partners

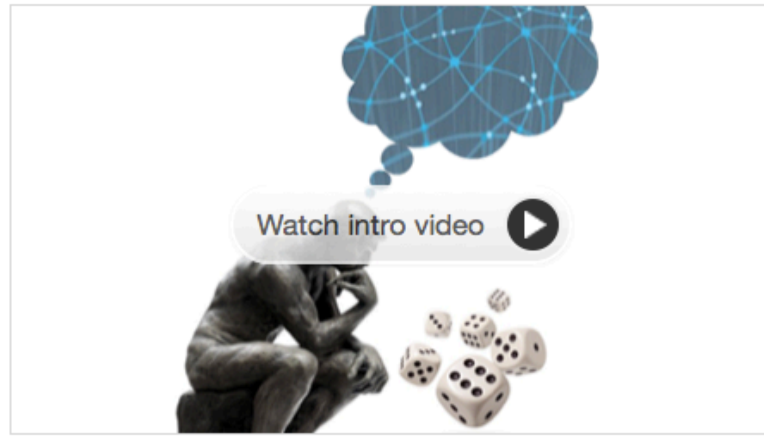
Courses Specializations New Institutions About | Sign In Sign Up

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

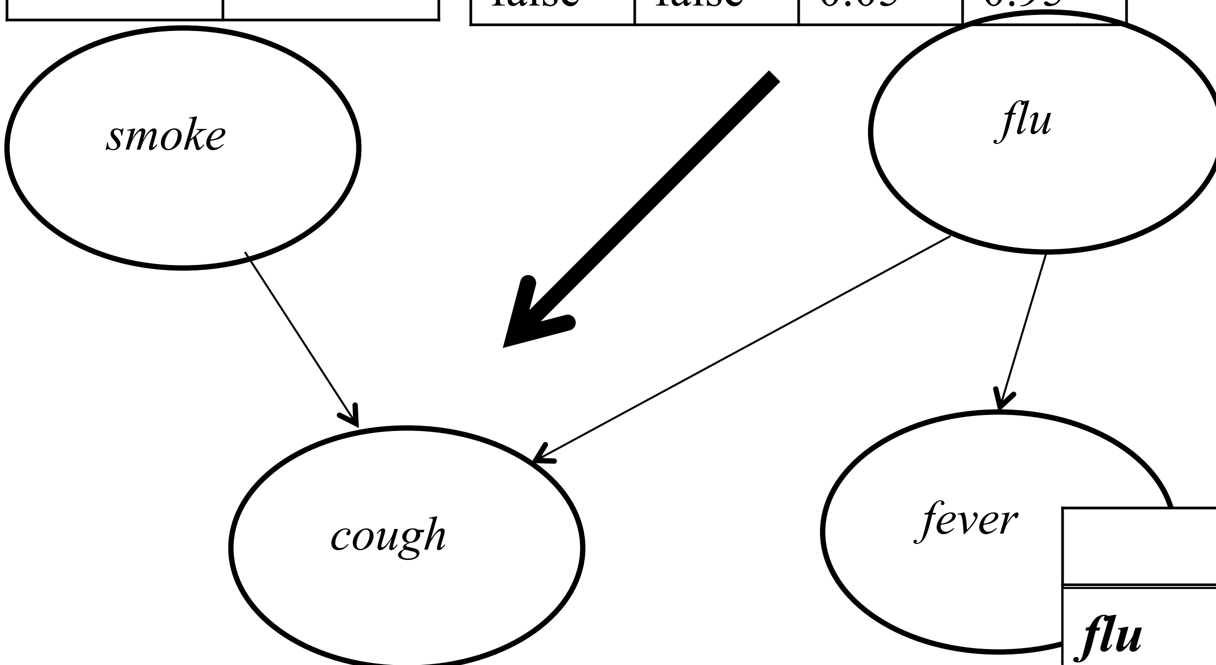
Example:

		cough	
flu	**smoke**	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

smoke	
true	0.2
false	0.8

flu	
true	0.01
false	0.99



	fever	
flu	true	false
true	0.9	0.1
false	0.2	0.8

Inference in Bayesian networks

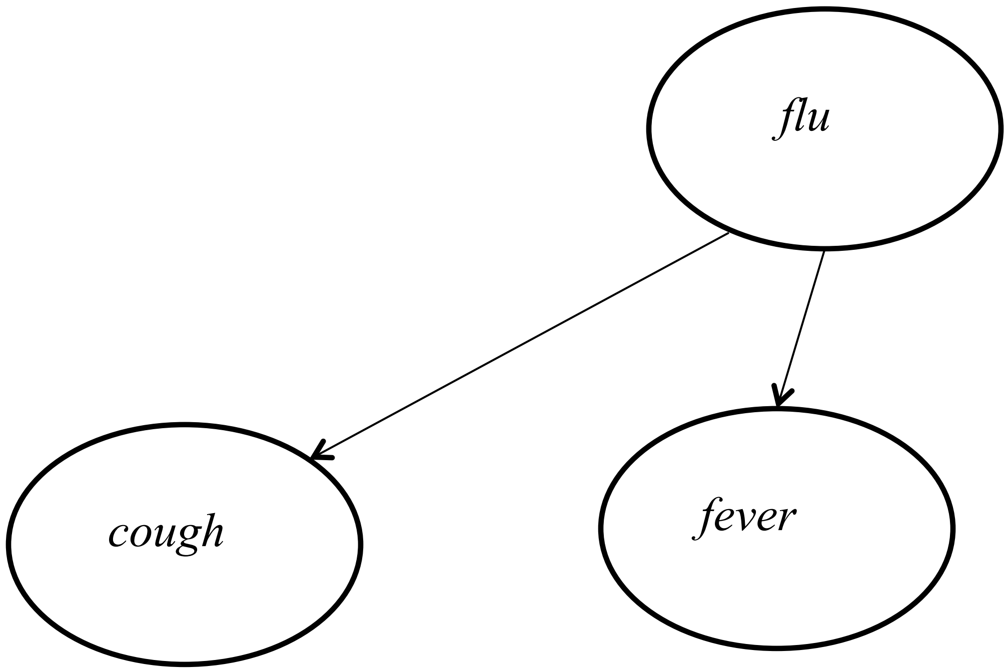
- If network is correct, can calculate full joint probability distribution from network.

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

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

where parents(X_i) denotes specific values of parents of X_i.

Naive Bayes Example



$$P(flu \mid cough, fever) \approx P(flu)P(cough \mid flu)P(fever \mid flu)$$

Example

- Calculate

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

$$= P\left(cough \wedge \neg fever \wedge \neg flu \wedge \neg smoke\right)$$

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

$$= P\left(cough \mid \neg flu \wedge \neg smoke\right) \times P\left(\neg fever \mid \neg flu\right) \times P(\neg flu) \times P(\neg smoke)$$

$$= .05 \times .8 \times .99 \times .8 = .032$$

In general...

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

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

where 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:

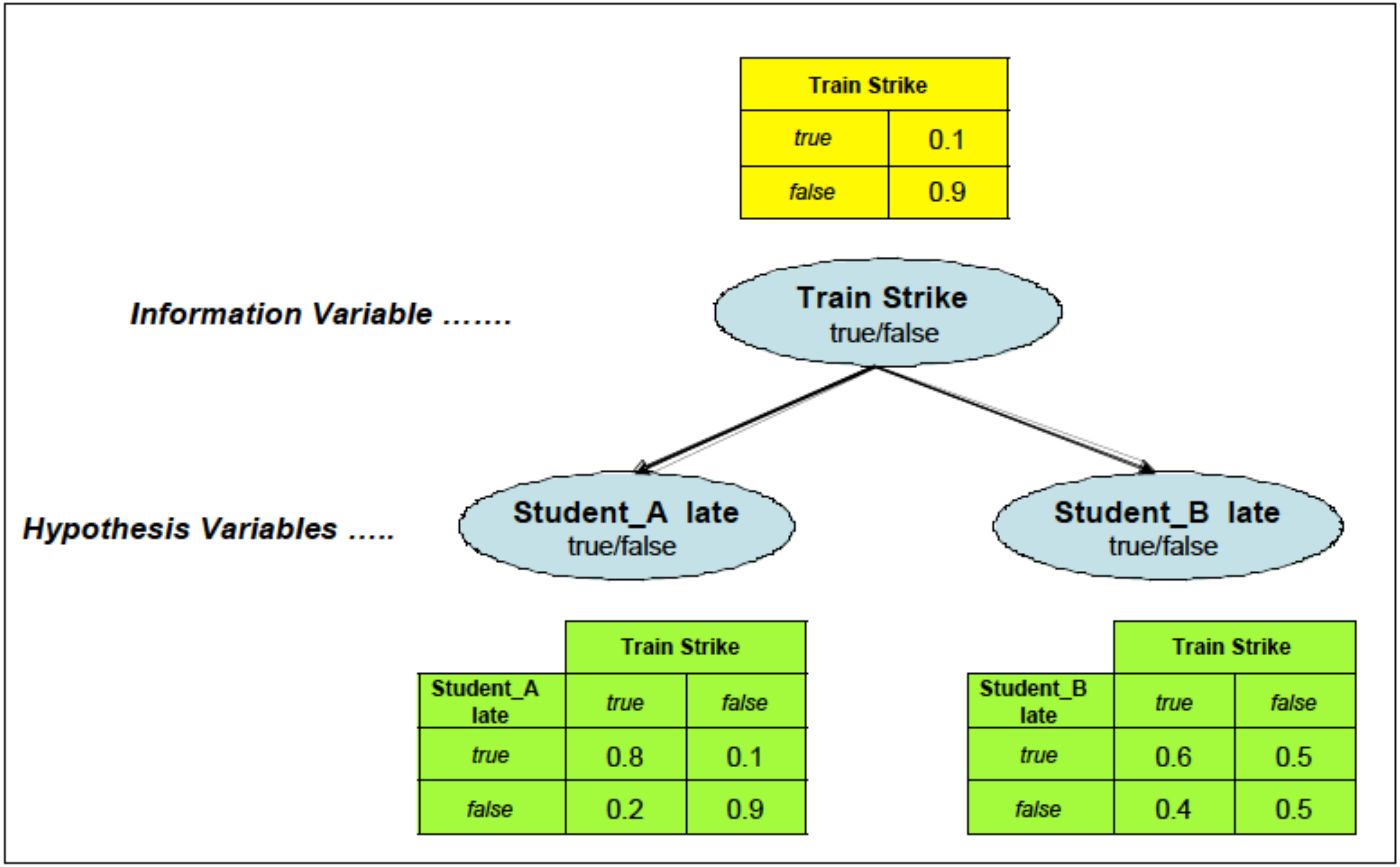


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

What is the probability that Student A is late?

What is the probability that Student B is late?

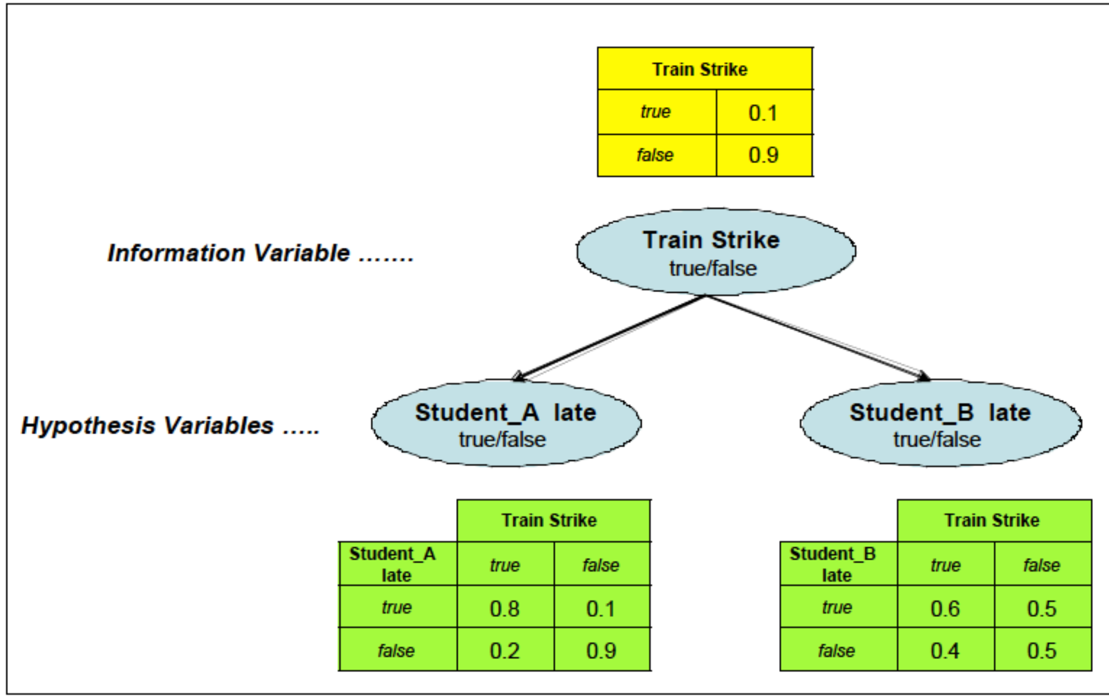


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Unconditional ("marginal") probability. We don't know if there is a train strike.

$$P(StudentALate) = P(StudentALate \mid TrainStrike)P(TrainStrike)$$
$$+P(StudentALate \mid \neg TrainStrike)P(\neg TrainStrike)$$
$$= 0.8 \times 0.1 + 0.8 \times 0.9 = 0.17$$

$$P(StudentBLate) = P(StudentBLate \mid TrainStrike)P(TrainStrike)$$
$$+P(StudentBLate \mid \neg TrainStrike)P(\neg TrainStrike)$$
$$= 0.6 \times 0.1 + 0.5 \times 0.9 = 0.51$$

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

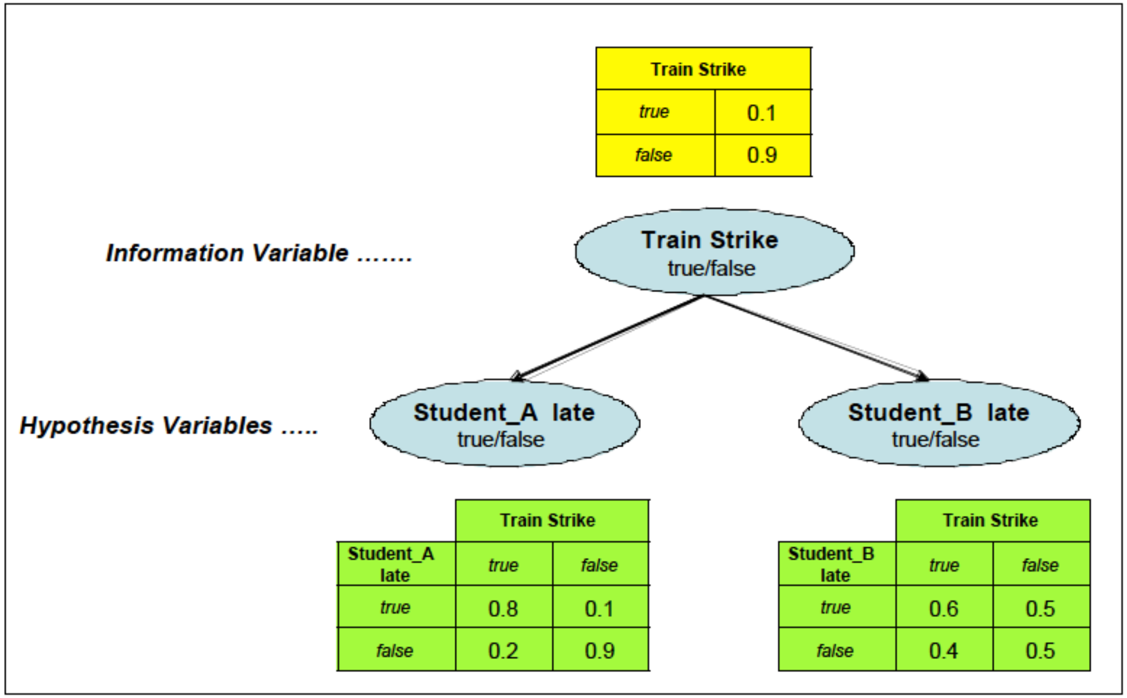


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Evidence: There is a train strike.

$$P(StudentALate) = 0.8$$

$$P(StudentBLate) = 0.6$$

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".

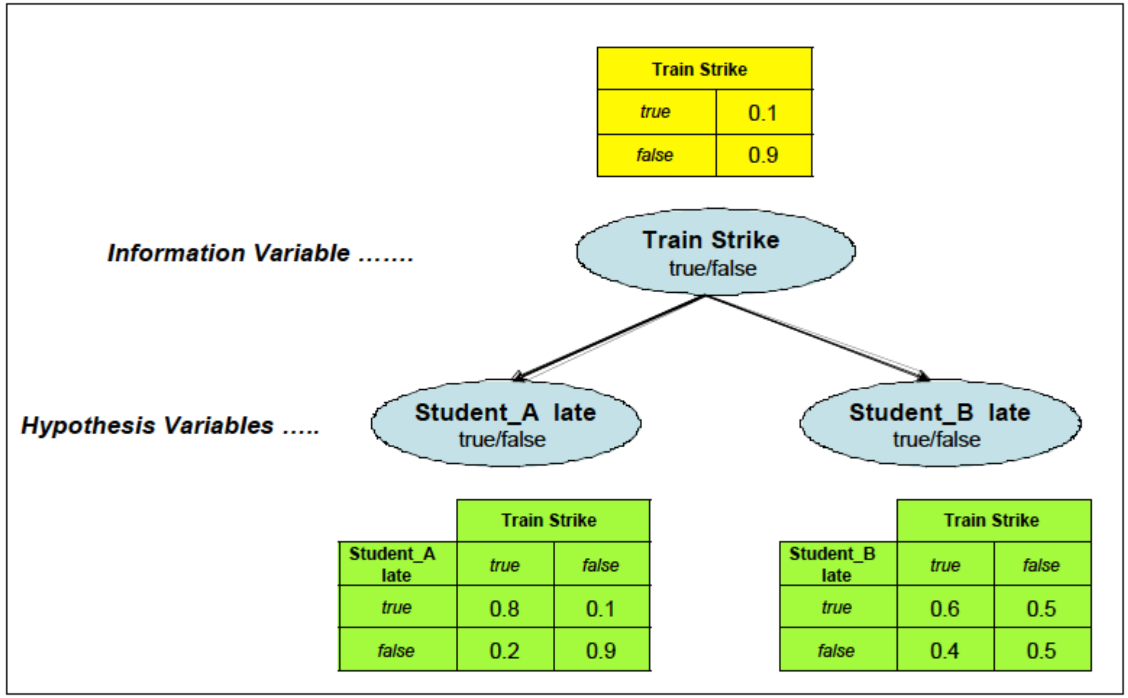


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

Evidence: Student A is late.

$$P(TrainStrike \mid StudentALate) = \frac{P(StudentALate \mid TrainStrike)P(TrainStrike)}{P(StudentALate)} \quad \text{by Bayes Theorem}$$

$$= \frac{0.8 \times 0.1}{0.17} = 0.47$$

$$P(StudentBLate) = P(StudentBLate \mid TrainStrike)P(TrainStrike)$$

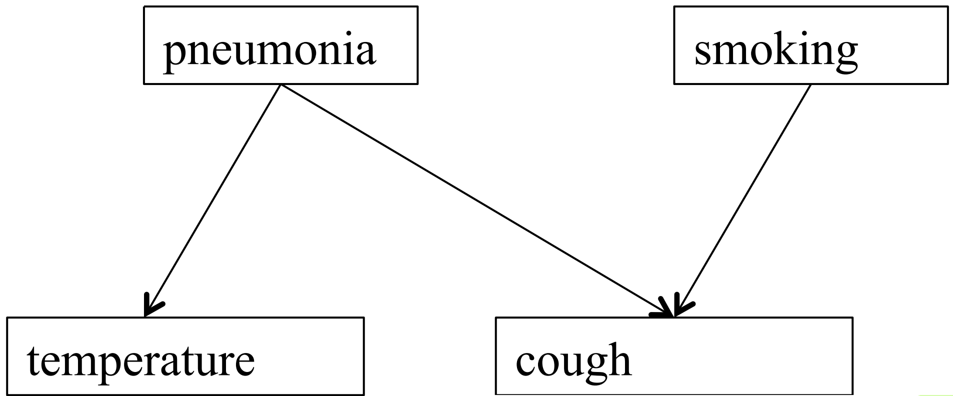
$$+P(StudentBLate \mid \neg TrainStrike)P(\neg TrainStrike)$$

$$= 0.6 \times 0.47 + 0.5 \times 0.53 = 0.55$$

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

Three types of inference

- **Diagnostic:** Use evidence of an effect to infer probability of a cause.
  - E.g., **Evidence:** *cough=true*. What is *P*(*pneumonia* | *cough*)?

- **Causal inference:** Use evidence of a cause to infer probability of an effect
  - E.g., **Evidence:** *pneumonia=true*. What is *P*(*cough* | *pneumonia*)?

- **Inter-causal inference:** "Explain away" potentially competing causes of a shared effect.
  - E.g., Evidence: *smoking=true.* What is *P*(*pneumonia* | *cough* and *smoking*)?

Math we used:

- **Definition of conditional probability:**

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

- **Bayes Theorem**

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

- **Unconditional (marginal) probability**

If X_1 depends on X_2 then

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

- **Probability inference in Bayesian networks:**

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

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".*

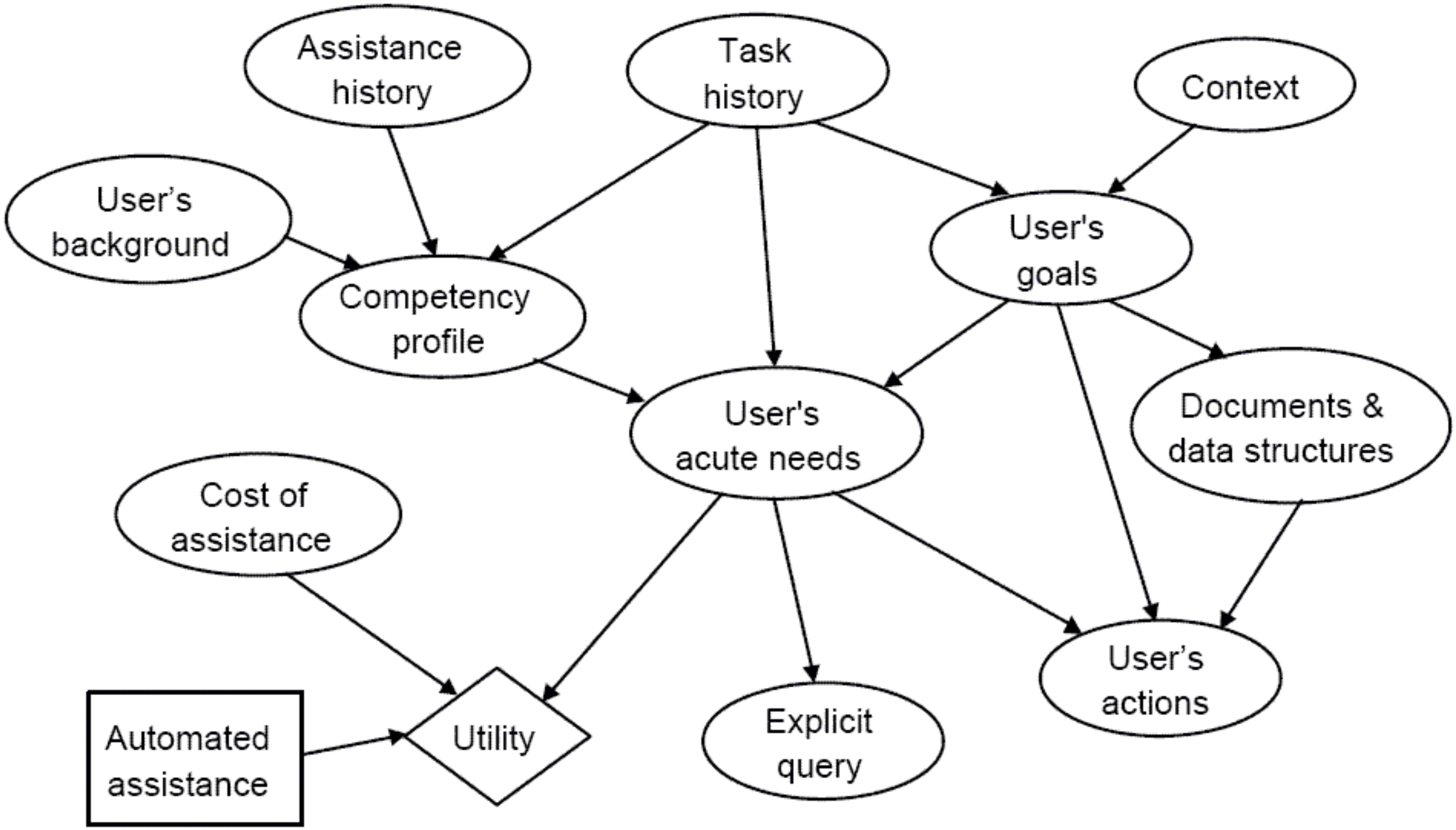


Figure 1: An influence diagram for providing intelligent assistance given uncertainty in a user's background, goals, and competency in working with a software application.

- Networks were developed by experts from user modeling studies.

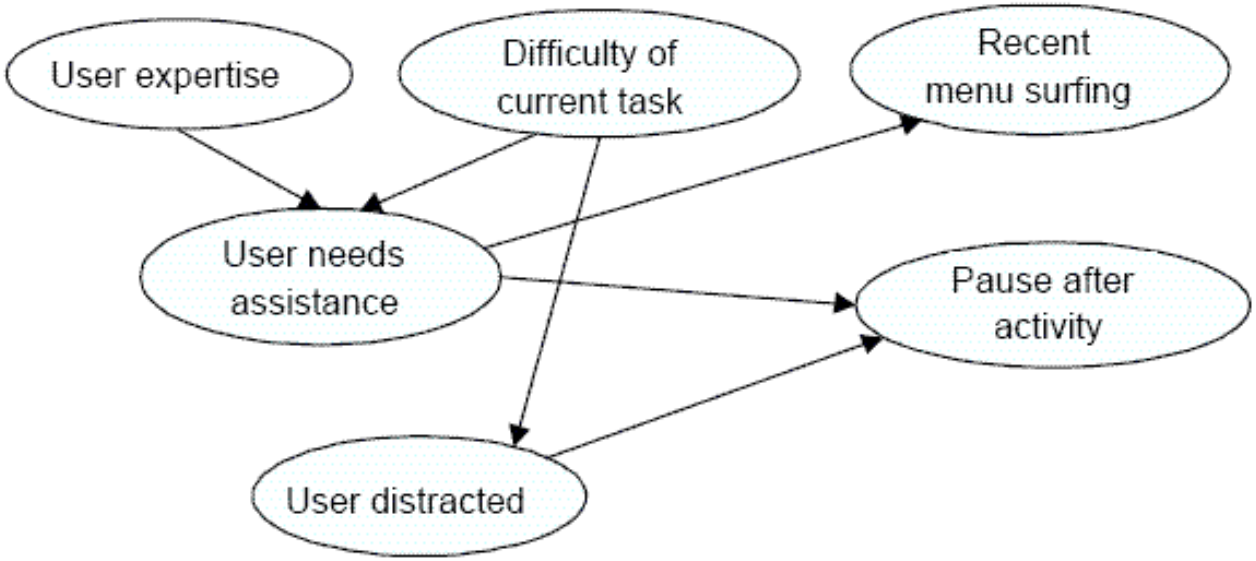


Figure 2: A portion of a Bayesian user model for inferring the likelihood that a user needs assistance, considering profile information as well as observations of recent activity.

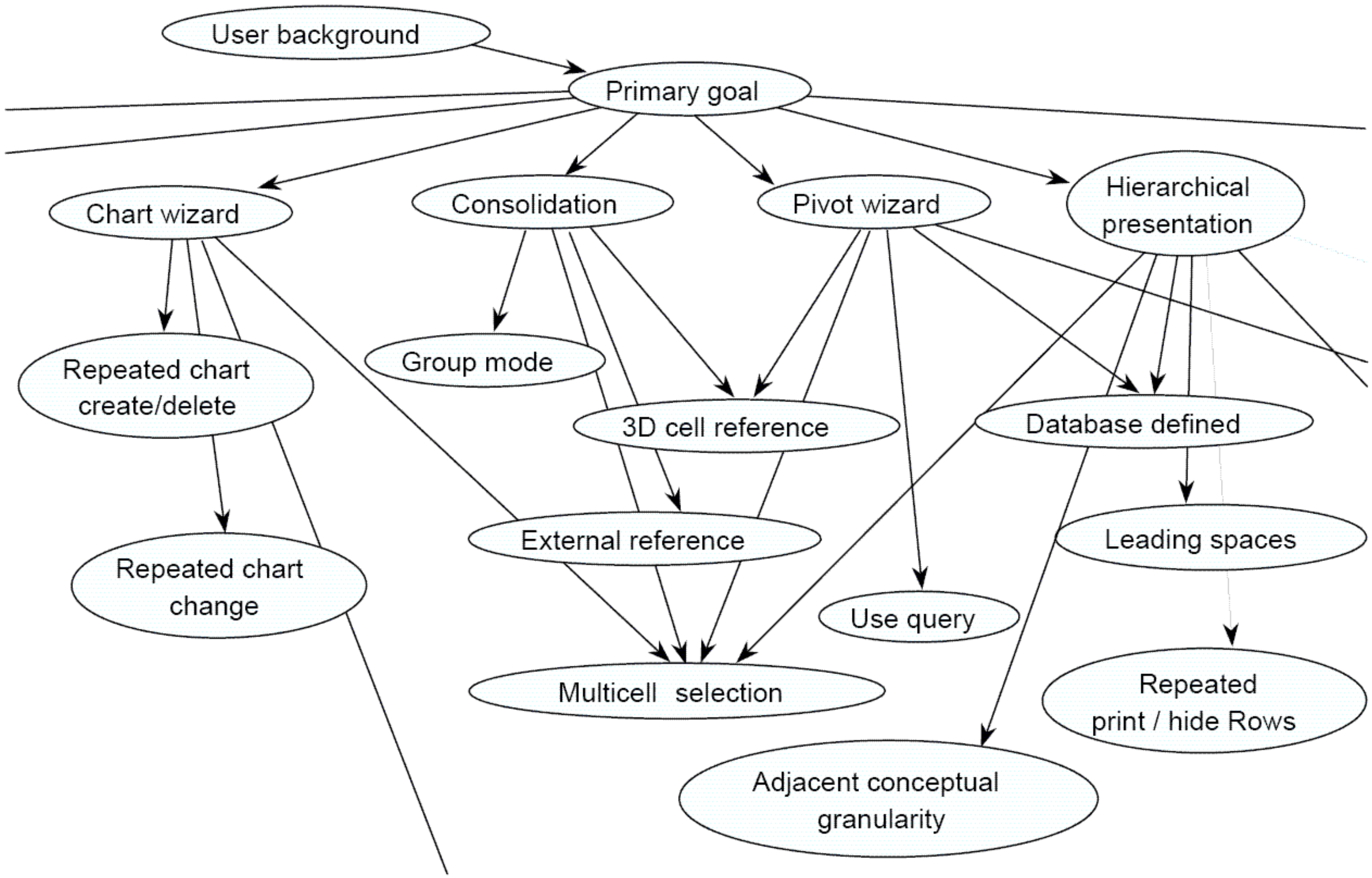
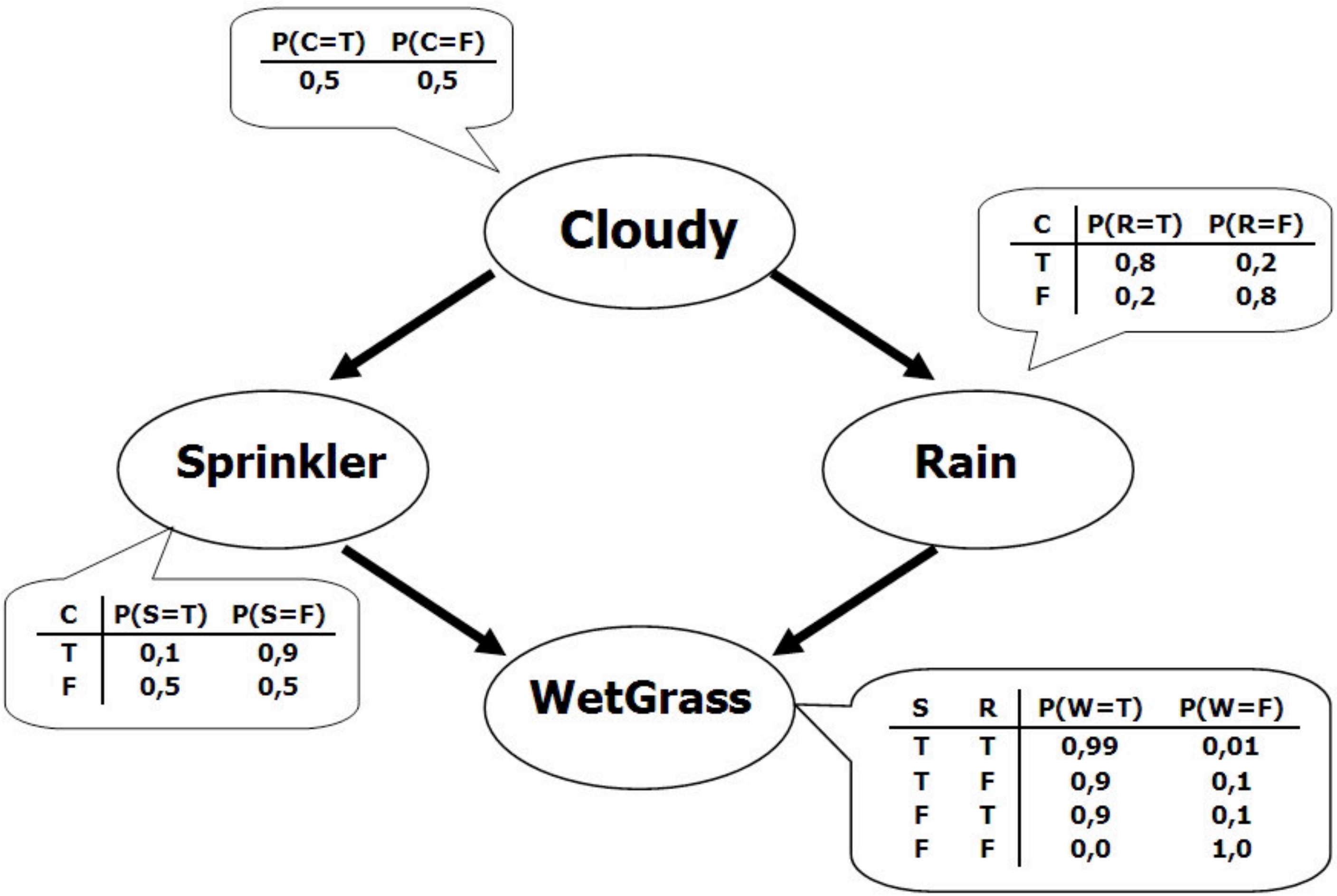


Figure 3: Partial structure of an early formulation of a Bayesian user model for inferring a user's needs for the Excel spreadsheet application.

- 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)



P(C=T)	P(C=F)
0,5	0,5

C	P(R=T)	P(R=F)
T	0,8	0,2
F	0,2	0,8

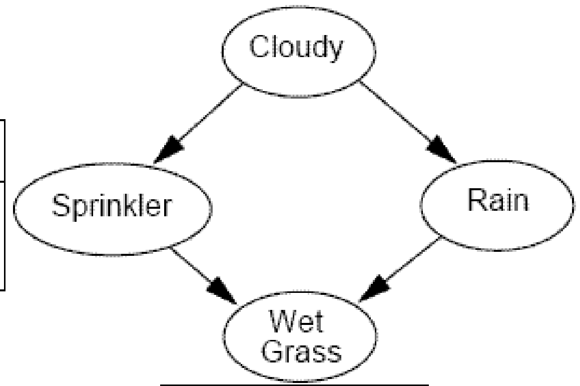
C	P(S=T)	P(S=F)
T	0,1	0,9
F	0,5	0,5

S	R	P(W=T)	P(W=F)
T	T	0,99	0,01
T	F	0,9	0,1
F	T	0,9	0,1
F	F	0,0	1,0

Exact Inference in Bayesian Networks

Example

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

General question: What is $P(X|\boldsymbol{e})$?

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

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

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

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

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

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

- Worst-case complexity is exponential in *n* (number of nodes)

- Problem is having to enumerate all possibilities for many variables.

$$\alpha \sum_{s} \left[P(c)P(r \mid c)P(s \mid c)P(w \mid s, r) \right]$$

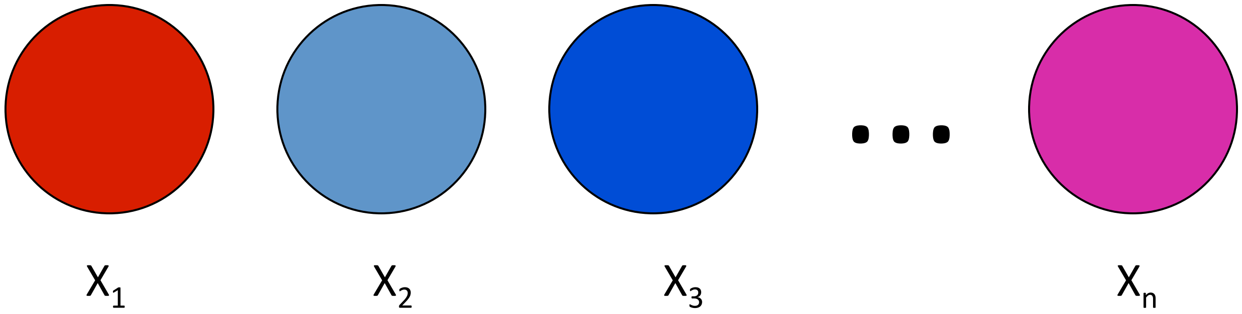
Can reduce computation by computing terms only once and storing for future use.

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.



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

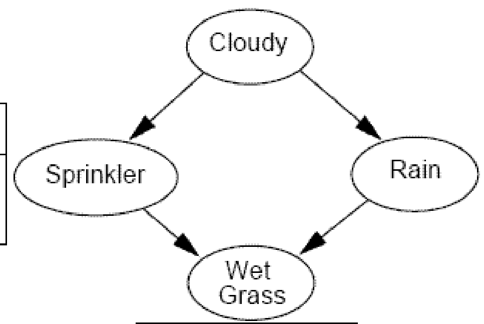
- **Fact:** By construction of Bayesian network, a node is conditionally independent of its non-descendants, given its parents.

What is the Markov Blanket of *Rain*?

What is the Markov blanket of *Wet Grass*?

Example

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

- **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: $(x_1, ..., x_n)$. 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(*Rain* | *Sprinkler* = *true*, *WetGrass* = *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

  $\mathrm{P}(Rain \mid Sprinkler = true, WetGrass = true)$

- Suppose we perform 100 such samples, 20 with $Rain = true$ and 80 with $Rain = false$.

- Then answer to the query is

  `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

Markov Chain Monte Carlo Algorithm in a nutshell

- **Markov blanket** of a variable X_i:
  - parents, children, children's other parents

- **MCMC algorithm:**

For a given set of evidence variables $\{X_j = x_k\}$

Repeat for NumSamples:

- Start with random sample from variables, with evidence variables fixed: $(x_1, \ldots, x_n)$. 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.

Finally, return the estimated distribution of each non-evidence variable X_i

Example

- Query: What is P(*Sprinkler* =*true* | *WetGrass* = *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 *Sprinkler*, given current values of its Markov blanket:

       Cloudy = *false, Rain*= *false, Wet* = *true*. Suppose

       result is *true*. New state: [false, true, false, true].

- Each sample contributes to estimate for query

  $P(Sprinkler = true \mid WetGrass = true)$

- Suppose we perform 50 such samples, 20 with *Sprinkler* = *true* and 30 with *Sprinkler*= *false.*

- Then answer to the query is

  `Normalize` $(\langle 20,30 \rangle) = \langle .4,.6 \rangle$

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 $\Rightarrow$ 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 **X** be the “state variables” (i.e., set of non-evidence variables) describing the environment (e.g., *Words* said during time step **t**)

  - Let **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 $P(Words \mid \mathbf{S})$.

- We know from Bayes rule:

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

- $P(\boldsymbol{S} \mid Words)$, 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).

- $\boldsymbol{P}(Words)$, 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 **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

$\Rightarrow$ acoustic model = pronunciation model + phone model

ARPAbet designed for American English

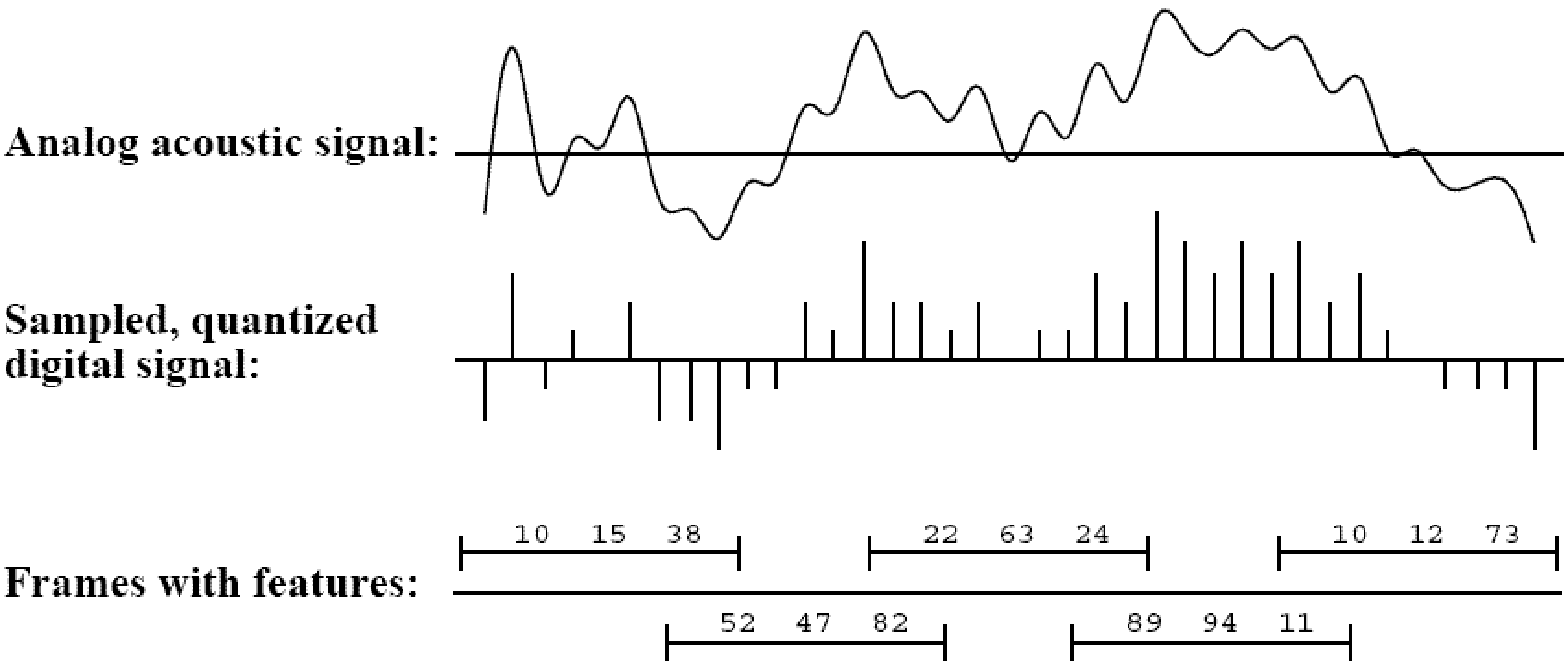
[iy]	beat	[b]	bet	[p]	pet
[ih]	bit	[ch]	Chet	[r]	rat
[ey]	bet	[d]	debt	[s]	set
[ao]	bought	[hh]	hat	[th]	thick
[ow]	boat	[hv]	high	[dh]	that
[er]	Bert	[l]	let	[w]	wet
[ix]	roses	[ng]	sing	[en]	button
⋮	⋮	⋮	⋮	⋮	⋮

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

From http://www.cs.berkeley.edu/~russell/slides/

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

From http://www.cs.berkeley.edu/~russell/slides/

Hidden Markov Models

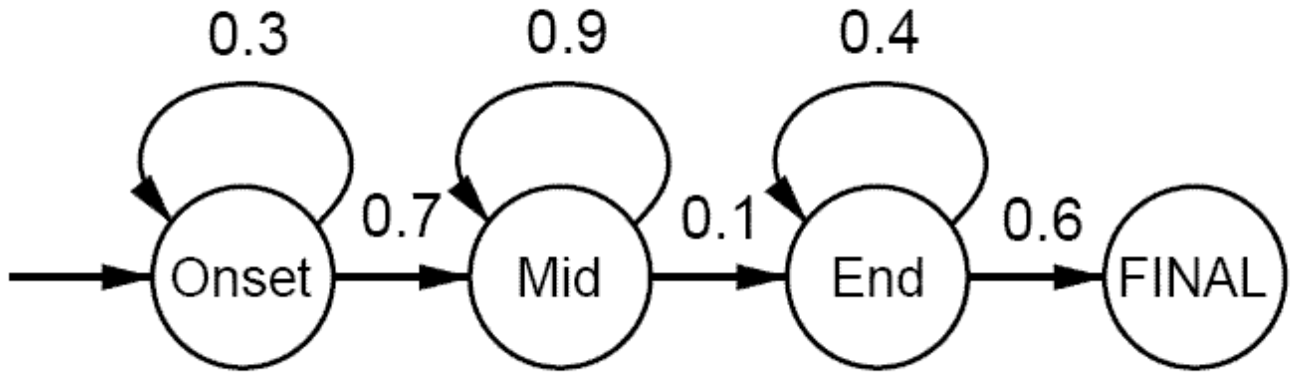
- **Markov model**: Given state $\boldsymbol{X}_t$, what is probability of transitioning to next state $\boldsymbol{X}_{t+1}$?
  - E.g., word bigram probabilities give

    $P\,(word_{t+1} \mid word_t)$

- **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:

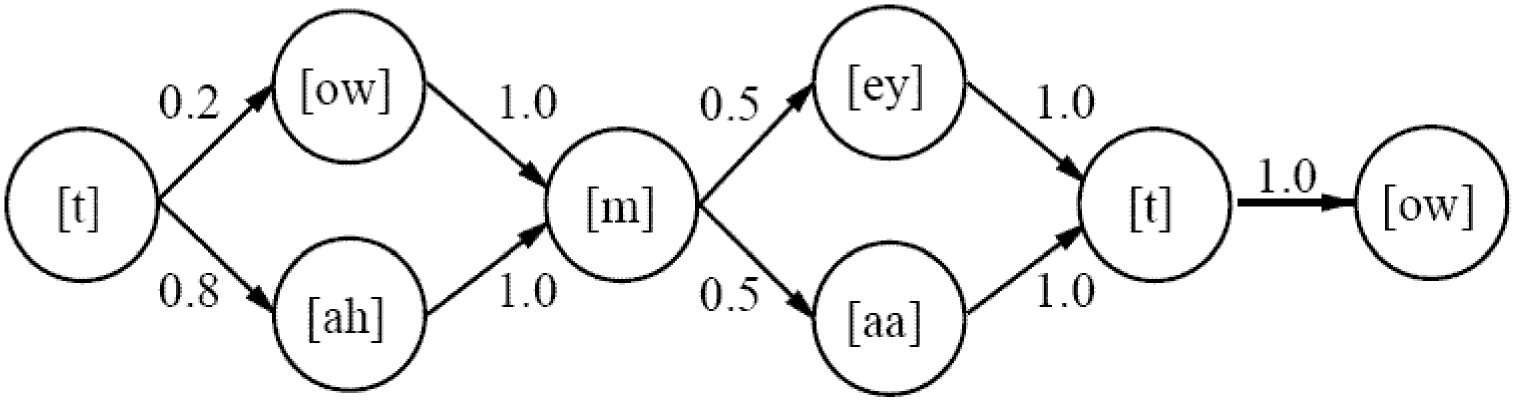
Onset:	Mid:	End:
C1: 0.5	C3: 0.2	C4: 0.1
C2: 0.2	C4: 0.7	C6: 0.5
C3: 0.3	C5: 0.1	C7: 0.4

From http://www.cs.berkeley.edu/~russell/slides/

Word pronunciation models

Each word is described as a distribution over phone sequences

Distribution represented as an HMM transition model



$$P([towmeytow]|\text{“tomato”}) = P([towmaatow]|\text{“tomato”}) = 0.1$$
$$P([tahmeytow]|\text{“tomato”}) = P([tahmaatow]|\text{“tomato”}) = 0.4$$

Structure is created manually, transition probabilities learned from data

From http://www.cs.berkeley.edu/~russell/slides/

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?"

From http://www.cs.berkeley.edu/~russell/slides/

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

From http://www.cs.berkeley.edu/~russell/slides/