

Reasoning Under Uncertainty

Reading assignment:
“Bayesian Belief Networks”
by Scott Wooldridge

(Download from link on class website)

What is “Reasoning under Uncertainty”?

- Using probability theory to do automatic reasoning when not all aspects of the problem are known with certainty.
- E.g.:
 - Medical diagnosis
 - Speech recognition
 - Robot planning
 - Just about every AI task!

A famous example: The Monty Hall Problem

You are a contestant on a game show.

There are 3 doors, A, B, and C. There is a new car behind one of them and goats behind the other two.

Monty Hall, the host, asks you to pick a door, any door. You pick door A.

Monty tells you he will open a door, different from A, that has a goat behind it. He opens door B: behind it there is a goat.

Monty now gives you a choice: Stick with your original choice A or switch to C.

Should you switch?

<http://math.ucsd.edu/~crypto/Monty/monty.html>

Bayesian probability formulation

Hypothesis space H :

h_1 = Car is behind door A

h_2 = Car is behind door B

h_3 = Car is behind door C

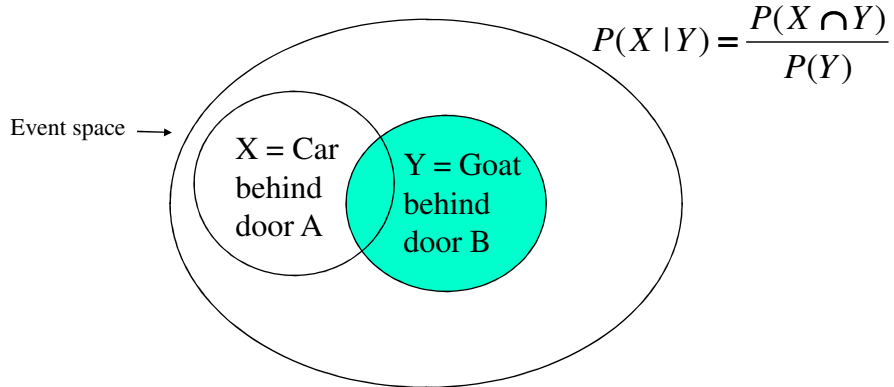
Data D = Monty opened B

What is $P(h_1 | D)$?

What is $P(h_2 | D)$?

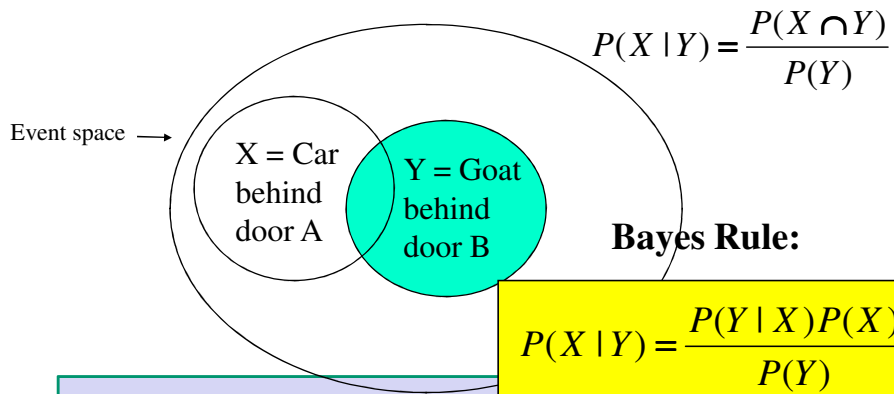
What is $P(h_3 | D)$?

Recall definition of conditional probability:



Event space = All possible configurations of cars and goats behind doors A, B, C

Recall definition of conditional probability:



Proof :

$$P(X | Y)P(Y) = P(X \cap Y) = P(Y | X)P(X)$$

Terminology

– **Prior probability of h :**

- $P(h)$: Probability that hypothesis h is true given our prior knowledge
- If no prior knowledge, all $h \in H$ are equally probable

– **Posterior probability of h :**

- $P(h|D)$: Probability that hypothesis h is true, given the data D .

– **Likelihood of D :**

- $P(D|h)$: Probability that we will see data D , given hypothesis h is true.

Bayes rule says:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Another example (Adapted from the reading)

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

Hypothesis space H :

h_1 : patient has flu
 h_2 : patient does not have flu

Data D :

$coughing = \text{true}, fever = \text{true}$

Prior probabilities:

$p(h_1) = .1$
 $p(h_2) = .9$

Likelihoods

$p(D|h_1) = .8$
 $p(D|h_2) = .4$

Prob. of data

$P(D) =$

Posterior probabilities:

$P(h_1|D) =$

$P(h_2|D) =$

- Let's say we have the following random variables:

cough

fever

flu

smokes

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.

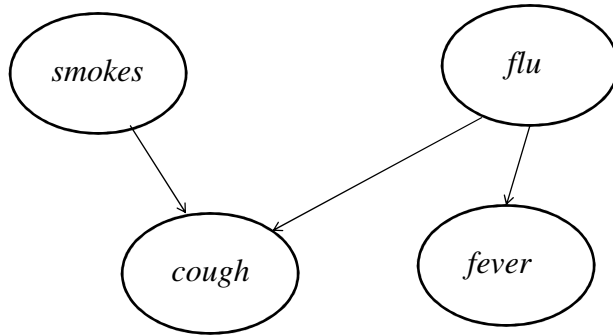
In principle, the full joint distribution can be used to answer any question about probabilities of these combined parameters.

\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.

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”

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

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

```

    graph TD
      smoke((smoke)) --> cough((cough))
      flu((flu)) --> cough
      flu --> fever((fever))
  
```

Semantics of Bayesian networks

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

$$P((X_1 = x_1) \wedge (X_2 = x_2) \dots \wedge (X_n = x_n))$$

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

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

Example

- Calculate

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

Examples

What is unconditional (marginal) probability that *fever* is true?

What is the unconditional (marginal) probability that cough is true?

Different types of inference in Bayesian Networks

Causal inference

Evidence is cause, inference is probability of effect

Example:

Instantiate evidence $flu = true$. What is $P(fever | flu)$?

$$P(fever | flu) = .9 \text{ (up from .207)}$$

Diagnostic inference

Evidence is effect, inference is probability of cause

Example: Instantiate evidence $fever = true$. What is $P(flu | fever)$?

$$P(flu | fever) = \frac{P(fever | flu) P(flu)}{P(fever)} = \frac{(.9)(.01)}{.207} = .043 \text{ (up from .01)}$$

Example: What is $P(\text{flu}|\text{cough})$?

$$\begin{aligned}
 P(\text{flu}|\text{cough}) &= \frac{P(\text{cough}|\text{flu})P(\text{flu})}{P(\text{cough})} = \\
 &= \frac{[P(\text{cough}|\text{flu}, \text{smoke})p(\text{smoke}) \\
 &+ P(\text{cough}|\text{flu}, \neg\text{smoke})p(\neg\text{smoke})]P(\text{flu})}{p(\text{cough})} \\
 &= \frac{[(.95)(.2) + (.8)(.8)](.01)}{.167} = .0497
 \end{aligned}$$

Inter-causal inference

Explain away different possible causes of effect

Example: What is $P(\text{flu}|\text{cough}, \text{smoke})$?

$$\begin{aligned}
 P(\text{flu}|\text{cough}, \text{smoke}) &= \\
 &= \frac{p(\text{flu} \wedge \text{cough} \wedge \text{smoke})}{p(\text{cough} \wedge \text{smoke})} \\
 &= \frac{p(\text{cough}|\text{flu}, \text{smoke})p(\text{flu})p(\text{smoke})}{p(\text{cough}|\text{flu}, \text{smoke})p(\text{flu})p(\text{smoke}) + p(\text{cough}|\text{smoke}, \neg\text{flu})p(\text{smoke})p(\neg\text{flu})} \\
 &= (.95)(.01)(.2) / [(.95)(.01)(.2) + (.6)(.2)(.99)] \\
 &= 0.016
 \end{aligned}$$

Why is $P(\text{flu}|\text{cough}, \text{smoke}) < P(\text{flu}|\text{cough})$?

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

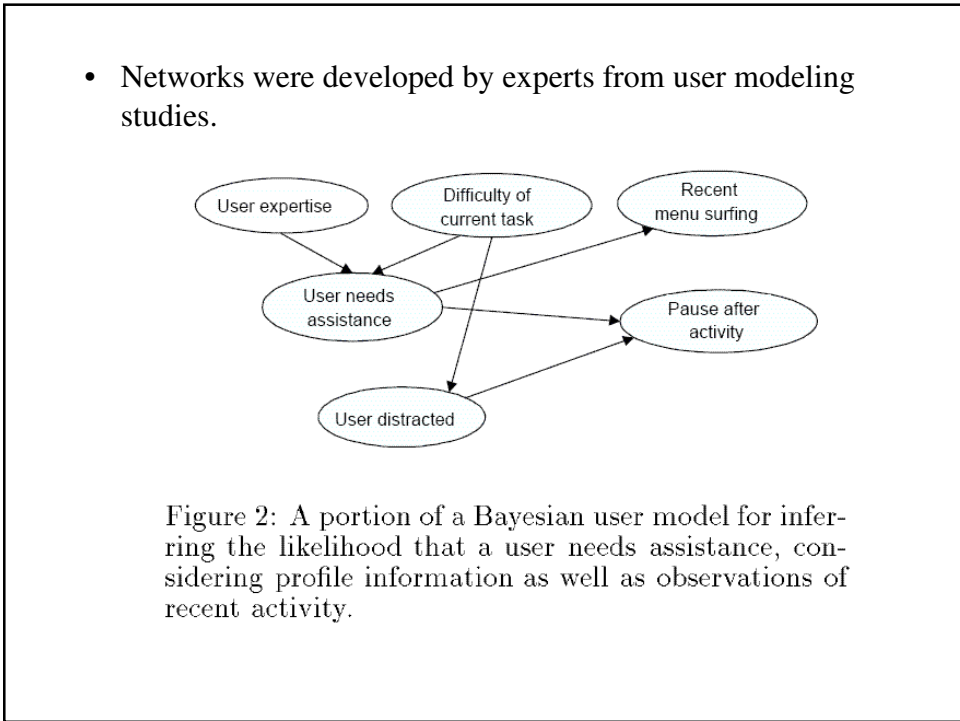
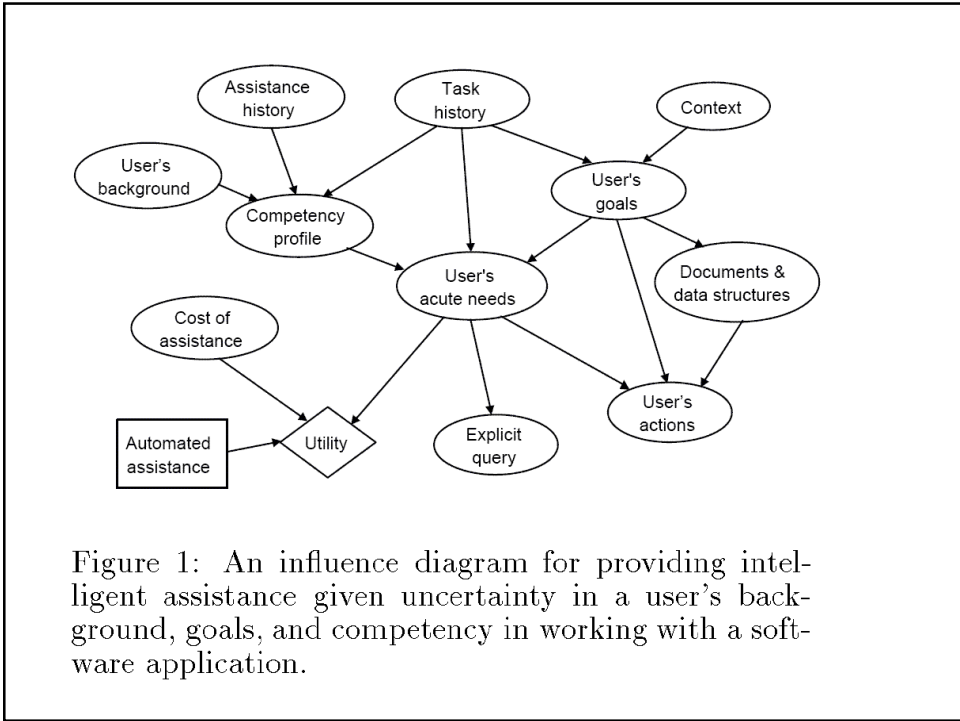
Issues in Bayesian Networks

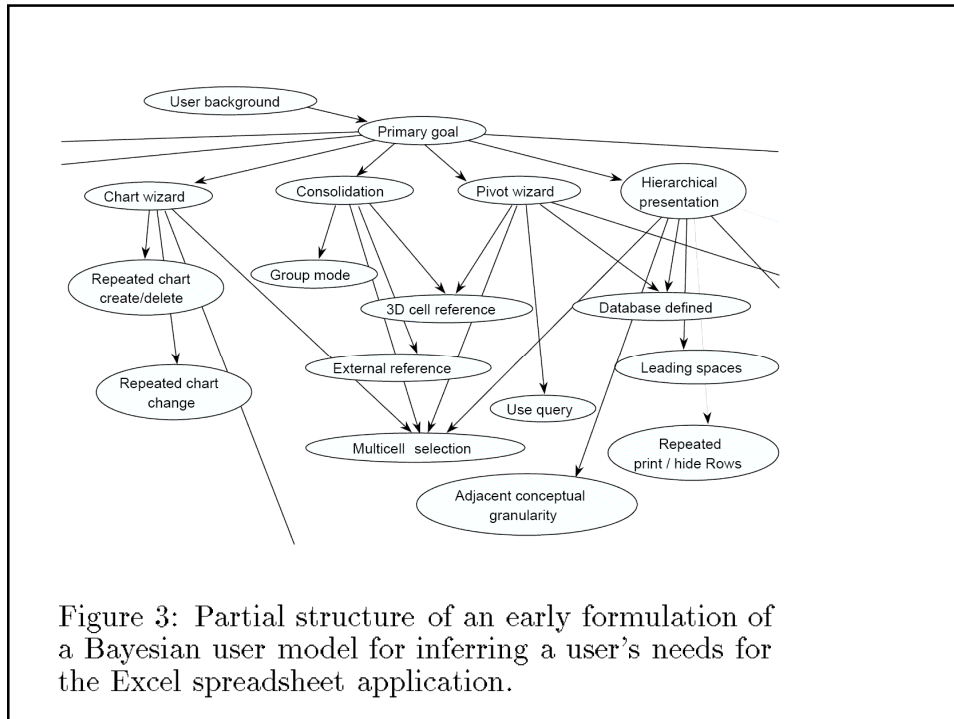
- Building / learning network topology
- Assigning / learning conditional probability tables
- Approximate inference via sampling

Real-World Example 1:

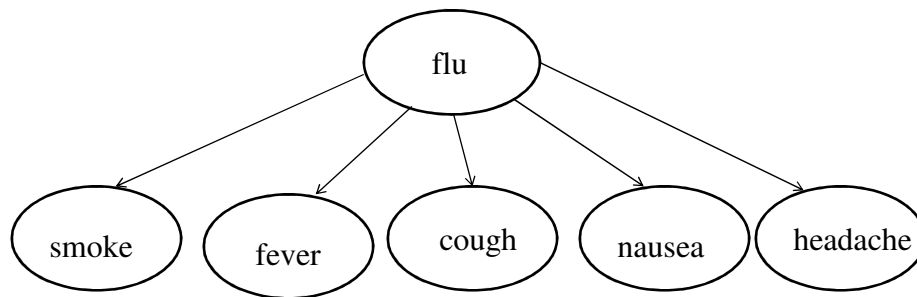
The Lumière Project at Microsoft Research

- 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”.*

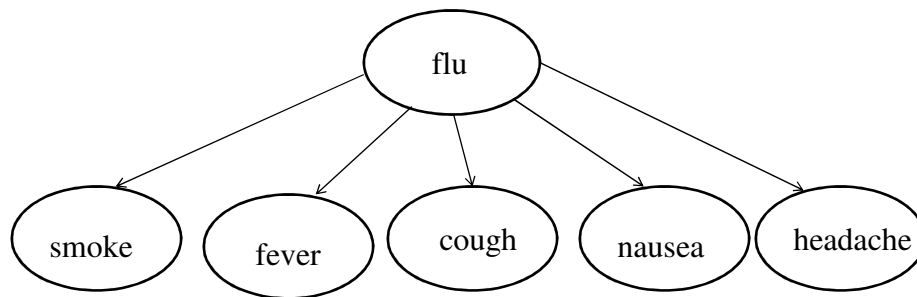




- Offspring of project was *Office Assistant* in Office 97.
- Video



$$\begin{aligned}
 &P(\text{flu} \wedge \text{smoke} \wedge \text{fever} \wedge \text{cough} \wedge \text{nausea} \wedge \text{headache}) \\
 &= P(\text{flu})P(\text{smoke} \mid \text{flu})P(\text{fever} \mid \text{flu})P(\text{cough} \mid \text{flu}) \\
 &\quad P(\text{nausea} \mid \text{flu})P(\text{headache} \mid \text{flu})
 \end{aligned}$$



More generally, for classification :


$$\begin{aligned}
 &P(C = c_j \mid X_1 = x_1, \dots, X_n = x_n) \\
 &= P(C = c_j) \prod_i P(X_i = x_i \mid C = c_j)
 \end{aligned}$$

“Naive Bayes”

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

Approximate inference via sampling

- Recall: We 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 .

- We can do diagnostic, causal, and inter-causal inference
- But if there are a lot of nodes in the network, this can be very slow!

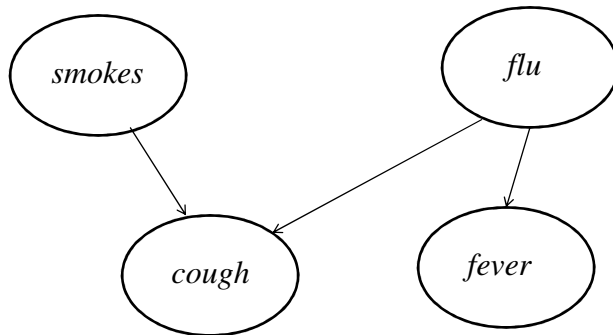
Need efficient algorithms to do approximate calculations!

Markov Chain Monte Carlo Sampling

- One of most common methods used in real applications.
- **Recall that:** By construction of Bayesian network, a node is conditionally independent of its non-descendants, given its parents.
- **Also recall that:** a node can be conditionally dependent on its children and on the other parents of its children. (Why?)
- **Definition:** The *Markov blanket* of a variable X_i is X_i 's parents, children, and children's other parents.

Example

- What is the Markov blanket of *cough*? of *flu*?



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

Markov Chain Monte Carlo (MCMC) Sampling

- Start with random sample from variables: (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 $P(\text{cough} | \text{smoke})$?**
- MCMC:
 - Random sample, with evidence variables fixed:
flu ***smoke*** *fever* *cough*
true **true** false true
 - Repeat:
 1. Sample *flu* probabilistically, given current values of its Markov blanket: *smoke* = true, *fever* = false, *cough* = true

Suppose result is *false*. New state:

flu ***smoke*** *fever* *cough*
false **true** false true

2. Sample *cough*, given current values of its Markov blanket:
smoke = *true* , *flu* = *false*

Suppose result is *true*.

New state:

<i>flu</i>	<i>smoke</i>	<i>fever</i>	<i>cough</i>
false	true	false	true

3. Sample *fever*, given current values of its Markov blanket:
flu = *false*

Suppose result is *true*.

New state:

<i>flu</i>	<i>smoke</i>	<i>fever</i>	<i>cough</i>
false	true	true	true

- Each sample contributes to estimate for query
 $P(\text{cough} \mid \text{smoke})$
- Suppose we perform 100 such samples, 20 with *cough* = true and 80 with *cough* = false.
- Then answer to the query is
 $P(\text{cough} \mid \text{smoke}) = .20$
- **Theorem:** MCMC settles into behavior in which each state is sampled exactly according to its posterior probability, given the evidence.

Applying Bayesian Reasoning to 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.

- 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., \mathbf{S} = 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 $P(\text{Words} | \mathbf{S})$.

- We know from Bayes rule:

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

- $P(\mathbf{S} | \text{Words})$, 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

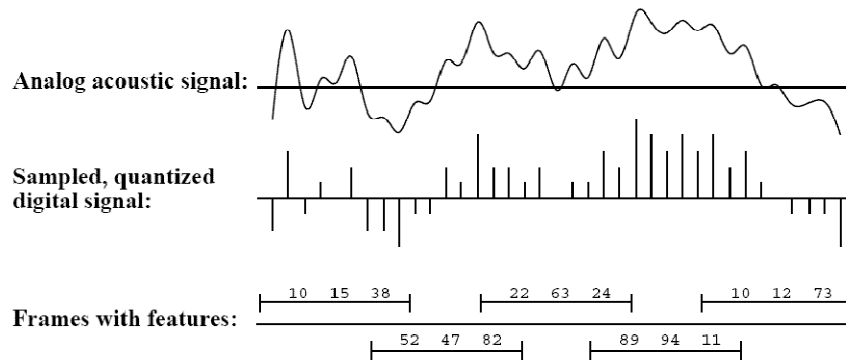
[iy]	beat	[b]	<u>b</u> et	[p]	<u>p</u> et
[ih]	bit	[ch]	<u>C</u> het	[r]	<u>r</u> at
[ey]	bet	[d]	<u>d</u> ebt	[s]	<u>s</u> et
[ao]	<u>b</u> ought	[hh]	<u>h</u> at	[th]	<u>t</u> hick
[ow]	<u>b</u> oat	[hv]	<u>h</u> igh	[dh]	<u>t</u> hat
[er]	B <u>e</u> rt	[l]	<u>l</u> et	[w]	<u>w</u> et
[ix]	ros <u>e</u> s	[ng]	<u>s</u> ing	[en]	but <u>t</u> on
:	:	:	:	:	:

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

From Russell and Norvig, *Artificial Intelligence*

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 Russell and Norvig, *Artificial Intelligence*

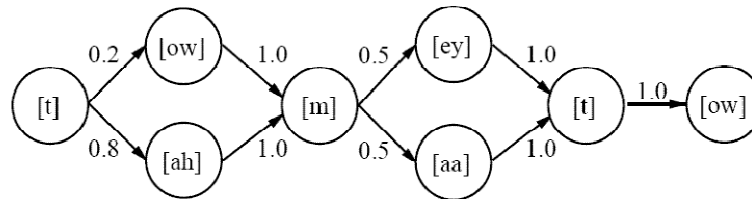
Hidden Markov Models

- **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.

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

From Russell and Norvig, *Artificial Intelligence*

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 firstly, 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

From Russell and Norvig, *Artificial Intelligence*