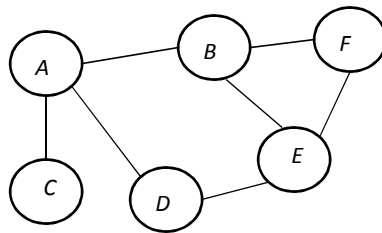


Reading:
Russell and Norvig, Chapter 15, Sections 15.1-15.3
will be on e-reserve

Markov Random Fields

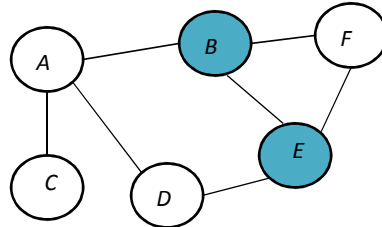
- **Undirected** graphical model



- **Random field:**
 - Each node is a random variable
 - Links represent conditional dependencies.
- **Markov Random Field**
 - Probabilistic dependencies only between neighboring nodes (Markov property)
 - Generalization of *Ising model* from statistical physics

Conditional Independence in Markov Random Fields

- Simpler than in Bayesian networks



- A node is conditionally independent of any other node iff all paths between those nodes are blocked (observed).
- Markov blanket of a node: all immediately neighboring nodes (Markov property)
- A node is conditionally independent of all other nodes, given its Markov blanket

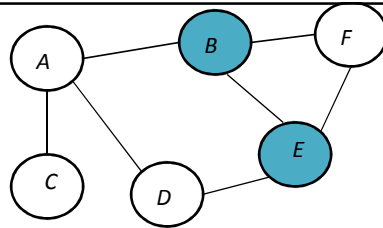
Factorization in Markov Random Fields

Consider nodes X_i and X_j that are not linked. Then

$$p(X_i, X_j | \mathbf{X} - \{X_i, X_j\}) = p(X_i | \mathbf{X} - \{X_i, X_j\})p(X_j | \mathbf{X} - \{X_i, X_j\})$$

Definition: A *clique* is a subset of nodes in a graph such that there exists a link between each pair of nodes in the subset.

Definition: A *maximal clique* is a clique such that it is not possible to include any other nodes from the graph in the subset without it ceasing to be a clique.



Denote a clique by C and the set of variables in that clique by \mathbf{X}_C

The conditional dependencies, or correlations, in a clique C are represented by a “potential function” of the variables in that clique: $\psi_C(\mathbf{X}_C)$

A joint probability distribution $p(X_1, \dots, X_N)$ can be written in terms of potential functions over maximal cliques as:

$$p(X_1, \dots, X_n) = \frac{1}{Z} \prod_C \psi_C(\mathbf{X}_C), \text{ where}$$

$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{X}_C)$$

- Usually express potential functions as:

$$\psi_C(\mathbf{X}_C) = e^{-E(\mathbf{X}_C)}$$

where E is an energy function. The exponential representation is called the *Boltzmann distribution*.

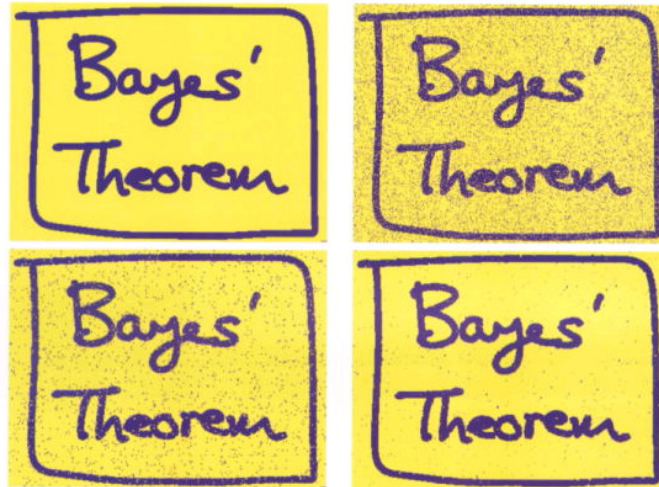
Joint distribution:

$$p(X_1, \dots, X_n) = \frac{1}{Z} \prod_C \psi_C(\mathbf{X}_C)$$

$$= \frac{1}{Z} \prod_C e^{-E(\mathbf{X}_C)}$$

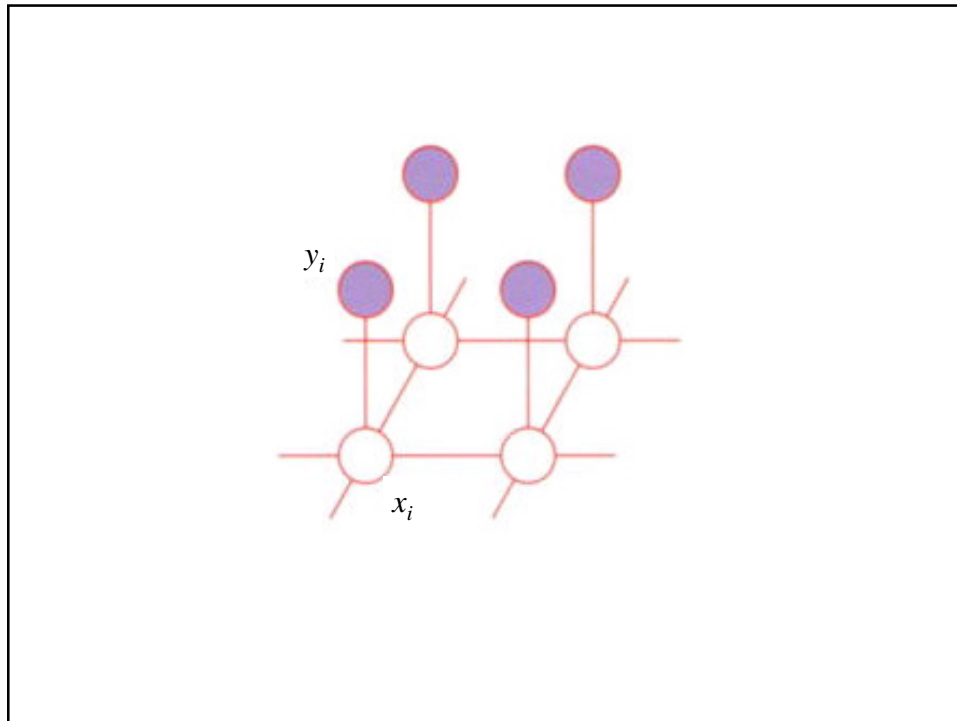
$$\ln p(X_1, \dots, X_n) = \left(\ln \frac{1}{Z} \right) \left(- \sum_C E(\mathbf{X}_C) \right)$$

Example application: Noise removal from a binary image (Bishop, 2006)



Example application: Noise removal from a binary image

- 10% of pixels sign flipped
- Node corresponds to binary pixels. Show Figure 8.31
- Cliques: pairs of nodes.
- Noise level is small, so strong correlation between x_i and y_i
- Also strong correlation in most images between neighboring pixels x_i and x_j .



- Energy function:

$$E(\mathbf{X}, \mathbf{Y}) = \sum_i X_i - \beta \sum_{i,j} X_i X_j - \eta \sum_i X_i Y_i$$

where β and η are constants.

Now, fix elements of $\mathbf{Y}=\mathbf{y}$. This gives a conditional distribution $p(\mathbf{X}|\mathbf{y})$. We want to find values of \mathbf{X} that have high probability. This corresponds to values that produce low energy.

One method for searching for such \mathbf{X} values: gradient ascent.

Gradient ascent in Markov Random Fields

- Initialize variables $\{X_i\}$ with random values of -1 and $+1$.
- For each node X_i , evaluate energy with $x_i = -1$ and $x_i = +1$, keeping all other variables unchanged. Choose assignment with lower total energy, breaking ties at random.
- Show results

Dynamic Bayesian Networks and Hidden Markov Models

- General dynamic Bayesian network: any number of random variables, which can be discrete or continuous
- Observations are taken in time steps.
- At each time step, observe some of the variables (*evidence* variables). Other variables are *unobserved* or “hidden”.
- Hidden Markov Model (HMM): State of process at time t is described by a single discrete hidden variable, X_t , with values in $\{1, \dots, S\}$, and a single evidence variable E_t .

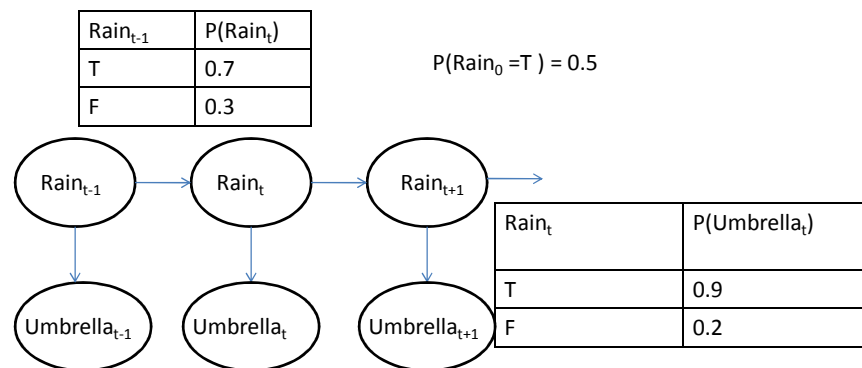
Simple example of HMM

(adapted from Russell and Norvig, Chapter 15)

You are a graduate student in a windowless office with no phone and no network connection. The only way you can get information about the weather outside is if your advisor shows up carrying an umbrella.

HMM for this scenario: Evidence variable: $Umbrella \in \{T, F\}$

Hidden variable $Rain \in \{T, F\}$



$$P(R_0, R_1, \dots, R_t, U_0, U_1, \dots, U_t) = P(R_0) \prod_{i=1}^t P(R_i | R_{i-1}) P(U_i | R_i)$$

Markov model since R_t depends only on R_{t-1} .

Inference in Hidden Markov Models

- **Inference tasks:**

- **Filtering** (or **monitoring**): Computing *belief state*—posterior distribution over current state, given all evidence to date:

$$P(X_t | \mathbf{e}_{1:t})$$

- **Prediction**: Computing posterior distribution over the future state, given all evidence to date:

$$P(X_{t+k} | \mathbf{e}_{1:t}), k > 0$$

- **Smoothing** (or **hindsight**): Computing posterior probability over a past state, given all evidence up to the present:

$$P(X_k | \mathbf{e}_{1:t}), 0 \leq k < t$$

- **Most likely explanation**: Given a sequence of observations, finding the sequence of states most likely to have generated those observations:

$$\arg \max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t} | \mathbf{e}_{1:t})$$

Inference algorithms

- **Filtering:** Can use recursive estimation

$$\begin{aligned}
 P(X_{t+1} | \mathbf{e}_{1:t+1}) &= P(X_{t+1} | \mathbf{e}_{1:t}, e_{t+1}) \quad (\text{dividing up the evidence}) \\
 &= \alpha P(e_{t+1} | X_{t+1}, \mathbf{e}_{1:t}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{by Bayes rule}) \\
 &= \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{evidence at } t+1 \text{ depends only on hidden state at } t+1)
 \end{aligned}$$

Inference algorithms

- **Filtering:** Can use recursive estimation

$$\begin{aligned}
 P(X_{t+1} | \mathbf{e}_{1:t+1}) &= P(X_{t+1} | \mathbf{e}_{1:t}, e_{t+1}) \quad (\text{dividing up the evidence}) \\
 &= \alpha P(e_{t+1} | X_{t+1}, \mathbf{e}_{1:t}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{by Bayes rule}) \\
 &= \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{evidence at } t+1 \text{ depends only on hidden state at } t+1)
 \end{aligned}$$

The value of the first term, $P(e_{t+1} | X_{t+1})$, is given explicitly in the network.

Inference algorithms

- **Filtering:** Can use recursive estimation

$$\begin{aligned}
 P(X_{t+1} | \mathbf{e}_{1:t+1}) &= P(X_{t+1} | \mathbf{e}_{1:t}, e_{t+1}) \quad (\text{dividing up the evidence}) \\
 &= \alpha P(e_{t+1} | X_{t+1}, \mathbf{e}_{1:t}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{by Bayes rule}) \\
 &= \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{evidence at } t+1 \text{ depends only on hidden state at } t+1)
 \end{aligned}$$

The value of the first term, $P(e_{t+1} | X_{t+1})$, is given explicitly in the network.

The value of the second term is:

$$P(X_{t+1} | \mathbf{e}_{1:t}) = \sum_{x_t} P(X_{t+1} | x_t, \mathbf{e}_{1:t}) P(x_t | \mathbf{e}_{1:t})$$

Inference algorithms

- **Filtering:** Can use recursive estimation

$$\begin{aligned}
 P(X_{t+1} | \mathbf{e}_{1:t+1}) &= P(X_{t+1} | \mathbf{e}_{1:t}, e_{t+1}) \quad (\text{dividing up the evidence}) \\
 &= \alpha P(e_{t+1} | X_{t+1}, \mathbf{e}_{1:t}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{by Bayes rule}) \\
 &= \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | \mathbf{e}_{1:t}) \quad (\text{evidence at } t+1 \text{ depends only on hidden state at } t+1)
 \end{aligned}$$

The value of the first term, $P(e_{t+1} | X_{t+1})$, is given explicitly in the network.

The value of the second term is:

$$P(X_{t+1} | \mathbf{e}_{1:t}) = \sum_{x_t} P(X_{t+1} | x_t, \mathbf{e}_{1:t}) P(x_t | \mathbf{e}_{1:t})$$

$$\begin{aligned}
 \text{Thus: } P(X_{t+1} | \mathbf{e}_{1:t+1}) &= \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(X_{t+1} | x_t, \mathbf{e}_{1:t}) P(x_t | \mathbf{e}_{1:t}) \\
 &= \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(X_{t+1} | x_t) P(x_t | \mathbf{e}_{1:t}) \quad \text{using the Markov property}
 \end{aligned}$$

Inference algorithms

From the network, we have everything except $P(x_t, \mathbf{e}_{1:t})$.

Can estimate recursively.

Thus:

$$\begin{aligned}
 P(X_{t+1} | \mathbf{e}_{1:t+1}) &= \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(X_{t+1} | x_t, \mathbf{e}_{1:t}) P(x_t, \mathbf{e}_{1:t}) \\
 &= \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(X_{t+1} | x_t) P(x_t, \mathbf{e}_{1:t}) \text{ using the Markov property}
 \end{aligned}$$

Umbrella example:

- **Day 1:** Umbrella₁ = U₁ = T

Prediction $t = 0$ to $t = 1$:

$$\begin{aligned}
 P(R_1) &= \sum_{r_0 \in \{T, F\}} P(R_1 | r_0) P(r_0) = \langle 0.7, 0.3 \rangle \times 0.5 + \langle 0.3, 0.7 \rangle \times 0.5 \\
 &= \langle 0.5, 0.5 \rangle
 \end{aligned}$$

Updating with evidence for $t=1$:

$$\begin{aligned}
 P(R_1 | u_1) &= \alpha P(u_1 | R_1) P(R_1) = \alpha \langle 0.9, 0.2 \rangle \langle 0.5, 0.5 \rangle \\
 &= \alpha \langle 0.45, 0.1 \rangle = \langle 0.818, 0.182 \rangle
 \end{aligned}$$

Prediction $t = 0$ to $t = 1$:

$$P(R_2 | u_1) = \sum_{r_1} P(R_2 | r_1)P(r_1 | u_1) = \langle 0.7, 0.3 \rangle \times 0.818 + \langle 0.3, 0.7 \rangle \times 0.182 = \langle 0.627, 0.373 \rangle$$

Updating with evidence for $t=2$:

$$\begin{aligned} P(R_1 | u_{1:2}) &= \alpha P(u_2 | R_2)P(R_2 | u_1) = \alpha \langle 0.9, 0.2 \rangle \langle 0.627, 0.373 \rangle \\ &= \alpha \langle 0.565, 0.075 \rangle = \langle 0.883, 0.117 \rangle \end{aligned}$$

Why does probability of rain increase from day 1 to day 2?

Hidden Markov Models: Matrix Representations

- **Transition model:** $P(X_t | X_{t-1}) = \mathbf{T}$ ($S \times S$ matrix) where

$$\mathbf{T}_{i,j} = P(X_t = j | X_{t-1} = i)$$

- For umbrella model:

$$\mathbf{T} = P(X_t | X_{t-1}) = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{matrix} T \\ F \end{matrix}$$

- **Sensor model:** $P(e_t | X_t = i) = \mathbf{O}$ ($S \times S$ diagonal matrix) where

$$\mathbf{O}_{i,j} = \begin{cases} P(e_t | X_t = i), i = j \\ 0 \text{ otherwise} \end{cases}$$

- For umbrella model:

$$\mathbf{O} = P(e_t | X_t) = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$$

- Forward/backward algorithm – will be described later

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 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**”)

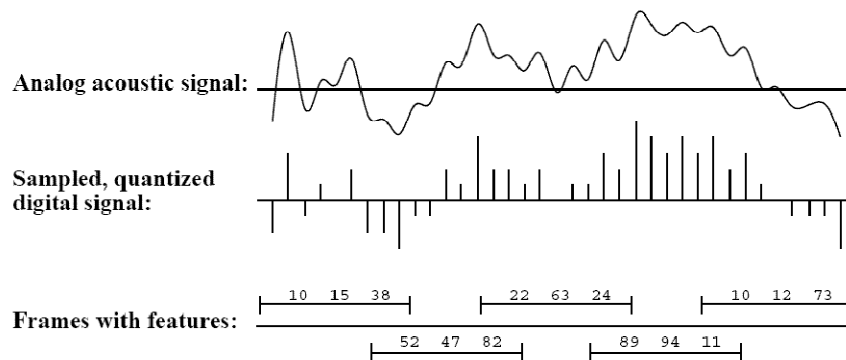
Speech Recognition

- Input: acoustic signal
- Inference: $P(\text{words} \mid \text{signal})$
- Bayes rule: $P(\text{words} \mid \text{signal}) = P(\text{signal} \mid \text{words}) P(\text{words})$
- $P(\text{signal} \mid \text{words})$: acoustic model
 - pronunciation model (for each word, distribution over possible phone sequences)
 - signal model (distribution of features of acoustic signal over phones)
- $P(\text{words})$: language model – prior probability of each utterance (e.g., bigram model)

Russell and Norvig, *Artificial Intelligence: A Modern Approach*, Chapter 15

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

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]	<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]	bu <u>t</u> ton
:	:	:	:	:	:

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

Phone model

$$P(\textit{phone} \mid \textit{frame features}) = \alpha P(\textit{frame features} \mid \textit{phone}) P(\textit{phone})$$

$P(\textit{frame features} \mid \textit{phone})$ often represented by Gaussian mixture model

Pronunciation model

Now we want

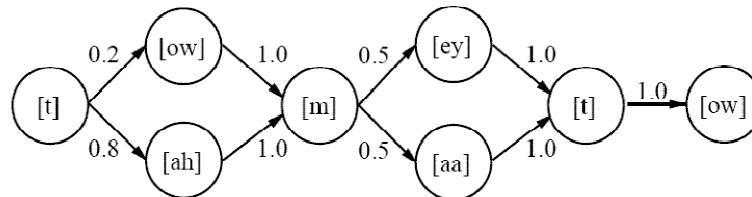
$$P(\textit{words} \mid \textit{phones}_{1:t}) = \alpha P(\textit{phones}_{1:t} \mid \textit{words}) P(\textit{words})$$

Represent $P(\textit{phones}_{1:t} \mid \textit{words})$ as an HMM

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

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

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
high 90s (%)?

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

How to learn HMMs

- EM algorithm can learn both transition and sensor models from data!