# CSE 150A-250A AI: Probabilistic Models

## Lecture 10

Fall 2025

Trevor Bonjour
Department of Computer Science and Engineering
University of California, San Diego

Slides adapted from previous versions of the course (Prof. Lawrence, Prof. Alvarado, Prof Berg-Kirkpatrick)

# Agenda

Review

EM Application

Hidden Markov Models

# Review

# ML estimation for complete data

- Notation

  Nodes $X_1, X_2, \ldots, X_n$
  Examples $t = 1, 2, \ldots, T$
  Complete data $\{(x_{1t}, x_{2t}, \ldots, x_{nt})\}_{t=1}^{T}$

- ML estimates for CPTs

| | |
|---|---|
| root nodes | $$P_{\mathrm{ML}}(X_i = x) = \frac{\mathrm{count}(X_i = x)}{T} = \frac{1}{T}\sum_t I(x_{it}, x)$$ |
| nodes with parents | $$P_{\mathrm{ML}}(X_i = x \vert \mathrm{pa}_i = \pi) = \frac{\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi)}{\mathrm{count}(\mathrm{pa}_i = \pi)} = \frac{\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)}{\sum_t I(\mathrm{pa}_{it}, \pi)}$$ |

# ML estimation for incomplete data

- Notation

  Nodes $X_1, X_2, \ldots, X_n$
  Examples $t = 1, 2, \ldots, T$
  Visible nodes $V_t = v_t$ for $t^{\text{th}}$ example

- EM algorithm

  Initialize CPTs to nonzero values.
  Repeat until convergence:

  E-step — compute posterior probabilities.
  M-step — update CPTs:

| | | | |
|---|---|---|---|
| root nodes | $P(X_i = x)$ | $\longleftarrow$ | $\frac{1}{T}\sum_t P(X_i = x \mid V_t = v_t)$ |
| nodes with parents | $P(X_i = x \mid \mathrm{pa}_i = \pi)$ | $\longleftarrow$ | $\frac{\sum_t P(X_i = x, \mathrm{pa}_i = \pi \mid V_t = v_t)}{\sum_t P(\mathrm{pa}_i = \pi \mid V_t = v_t)}$ |

# Complete versus incomplete data

- Complete data

| | |
|---|---|
| root nodes | $P_{\mathrm{ML}}(X_i = x) = \frac{1}{T} \sum_t I(x_{it}, x)$ |
| nodes with parents | $P_{\mathrm{ML}}(X_i = x \mid \mathrm{pa}_i = \pi) = \frac{\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)}{\sum_t I(\mathrm{pa}_{it}, \pi)}$ |

- Incomplete data

| | |
|---|---|
| root nodes | $P(X_i = x) \longleftarrow \frac{1}{T} \sum_t P(X_i = x \mid V_t = v_t)$ |
| nodes with parents | $P(X_i = x \mid \mathrm{pa}_i = \pi) \longleftarrow \frac{\sum_{t=1} P(X_i = x, \mathrm{pa}_i = \pi \mid V_t = v_t)}{\sum_{t=1}^{T} P(\mathrm{pa}_i = \pi \mid V_t = v_t)}$ |

# Key properties of EM

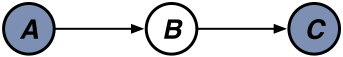
- No learning rate

  The updates do not require the tuning of a learning rate ($\eta > 0$), as in purely gradient-based methods.

- Monotonic convergence

  Changes to CPTs from the EM updates always increase the incomplete-data log-likelihood $\mathcal{L} = \sum_t \log P(V_t = v_t)$.

# EM Example




Incomplete data $\{(a_t, c_t)\}_{t=1}^{T}$
$A$ and $C$ are observed.
$B$ is hidden.

- **E-step** (Inference)

$$P(b|a_t, c_t) = \frac{P(c_t|b)\,P(b|a_t)}{\sum_{b'} P(c_t|b')\,P(b'|a_t)}$$

- **M-step** (Learning)

$$P(a) = \frac{1}{T}\,\mathrm{count}(A\!=\!a)$$

$$P(b|a) \longleftarrow \frac{\sum_t I(a, a_t)\,P(b|a_t, c_t)}{\sum_t I(a, a_t)}$$

$$P(c|b) \longleftarrow \frac{\sum_t I(c, c_t)\,P(b|a_t, c_t)}{\sum_t P(b|a_t, c_t)}$$

# EM Application

# Application

- Statistical language modeling

  Let $w_\ell$ denote the $\ell^{\text{th}}$ word in a corpus of text.
  How to model $P(w_1, w_2, \ldots, w_L)$?

- Markov models

| model | $P(w_1, w_2, \ldots, w_L)$ | ML estimate | DAG |
|---|---|---|---|
| unigram | $\prod_\ell P_1(w_\ell)$ | $P_1(w) = \frac{\text{count}(w)}{L}$ | $w_1 \quad w_2 \quad \cdots \quad w_L$ |
| bigram | $\prod_\ell P_2(w_\ell \mid w_{\ell-1})$ | $P_2(w' \mid w) = \frac{\text{count}(w \to w')}{\text{count}(w)}$ | $w_1 \to w_2 \to \cdots \to w_L$ |
| trigram | $\prod_\ell P_3(w_\ell \mid w_{\ell-1}, w_{\ell-2})$ | $\vdots$ | $\vdots$ |

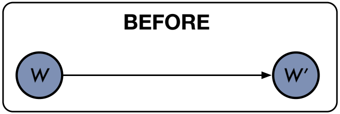
- Evaluating $n$-gram models

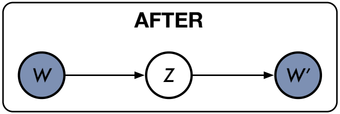
$$\text{Train on corpus } \mathcal{A} \implies P_1(\mathcal{A}) \leq P_2(\mathcal{A}) \leq P_3(\mathcal{A}) \ldots$$
$$\text{Test on corpus } \mathcal{B} \implies P_2(\mathcal{B}) = 0 \text{ if } \mathcal{B} \text{ has unseen bigrams.}$$

# Word clustering

- **Alternative to bigram model**

  Insert a hidden node $Z \in \{1, 2 \ldots, C\}$ between the previous and next words $W, W' \in \{1, 2, \ldots, V\}$.



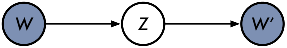




  Words $W$ and $W'$ are observed (as before).
  The node $Z$ is a latent variable to detect word clusters.

- **Conditional probability tables**

  $P(z|w)$ is the probability that word $w$ is mapped into cluster $z$.
  $P(w'|z)$ is the probability that word $w'$ follows any word in cluster $z$.

# Computing $P(w'|w)$




- Inference

$$
\begin{aligned}
P(w'|w) &= \sum_z P(w', z|w) && \boxed{\text{marginalization}} \\
&= \sum_z P(w'|z, w)\, P(z|w) && \boxed{\text{product rule}} \\
&= \sum_z P(w'|z)\, P(z|w) && \boxed{\text{conditional independence}}
\end{aligned}
$$

- Matrix factorization

  The above expresses the matrix $\overbrace{P(w'|w)}^{V\times V}$ as the product of the two smaller matrices $\underbrace{P(w'|z)}_{V\times C}$ and $\underbrace{P(z|w)}_{C\times V}$.

# Model complexity

- Parameter count

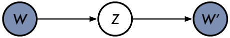
| | | |
|---|---|---|
| size of vocabulary | $V$ | |
| number of clusters | $C$ | |
| parameters in cluster model | $2CV$ | $P(w'\|z), P(z\|w)$ |
| parameters in bigram model | $V^2$ | $P(w'\|w)$ |
| parameters in unigram model | $V$ | $P(w)$ |

- Compact representations of complex worlds

  Setting $C = 1$, we recover the unigram model.
  Setting $C = V$, we recover the bigram model.
  In between, we are exploring a range of different models.

# EM algorithm

The model is the same as our previous example.
Only the variable names have changed!




- **E-step – Inference**

$$\color{orange}{P(z|w_\ell, w_{\ell+1})} = \frac{P(w_{\ell+1}|z)\, P(z|w_\ell)}{\sum_{z'} P(w_{\ell+1}|z')\, P(z'|w_\ell)}$$

- **M-step – Learning**

$$P(z|w) \longleftarrow \frac{\sum_\ell I(w, w_\ell) \color{orange}{P(z|w_\ell, w_{\ell+1})}}{\sum_\ell I(w, w_\ell)}$$

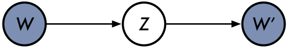
$$P(w'|z) \longleftarrow \frac{\sum_\ell I(w', w_{\ell+1})\, \color{orange}{P(z|w_\ell, w_{\ell+1})}}{\sum_\ell \color{orange}{P(z|w_\ell, w_{\ell+1})}}$$

# Experimental results

- **Data set**

  60K-word vocabulary
  80M-word corpus of news articles
  $\mathrm{count}(w \to w')$ is 99% sparse.

- **Model**




  The goal is to estimate $P(z|w)$ and $P(w'|z)$.
  For $C = 32$ clusters, these CPTs have 3.84M entries.
  EM converges in 30 iterations.

- **Results**

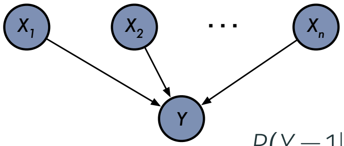
  The model has no prior knowledge of word meanings.
  Which words does it cluster? Look at $\mathrm{argmax}_z\, P(z|w)$.

# Word clusters

| | | | |
|---|---|---|---|
| 1 | as cents made make take | 19 | billion hundred million nineteen |
| 2 | ago day earlier Friday Monday month quarter reported said Thursday trading Tuesday Wednesday ⟨. . .⟩ | 20 | did ⟨”⟩ ⟨’⟩ |
| 3 | even get to | 21 | but called San ⟨:⟩ ⟨start-of-sentence⟩ |
| 4 | based days down home months up work years ⟨%⟩ | 22 | bank board chairman end group members number office out part percent price prices rate sales shares use |
| 5 | those ⟨,⟩ ⟨—⟩ | 23 | a an another any dollar each first good her his its my old our their this |
| 6 | ⟨.⟩ ⟨?⟩ | 24 | long Mr. year |
| 7 | eighty fifty forty ninety seventy sixty thirty twenty ⟨(⟩ ⟨·⟩ | 25 | business California case companies corporation dollars incorporated industry law money thousand time today war week ⟨)⟩ ⟨unknown⟩ |
| 8 | can could may should to will would | 26 | also government he it market she that there which who |
| 9 | about at just only or than ⟨&⟩ ⟨;⟩ | 27 | A. B. C. D. E. F. G. I. L. M. N. P. R. S. T. U. |
| 10 | economic high interest much no such tax united well | 28 | both foreign international major many new oil other some Soviet stock these west world |
| 11 | president | 29 | after all among and before between by during for from in including into like of off on over since through told under until while with |
| 12 | because do how if most say so then think very what when where | 30 | eight fifteen five four half last next nine oh one second seven several six ten third three twelve two zero ⟨-⟩ |
| 13 | according back expected going him plan used way | 31 | are be been being had has have is it’s not still was were |
| 15 | don’t I people they we you | 32 | chief exchange news public service trade |
| 16 | Bush company court department more officials police retort spokesman | | |
| 17 | former the | | |
| 18 | American big city federal general house military national party political state union York | | |

The table shows the most likely cluster assignments $\text{argmax}_z P(z|w)$ for the 300 most common tokens in the corpus.

# Example : Noisy-OR



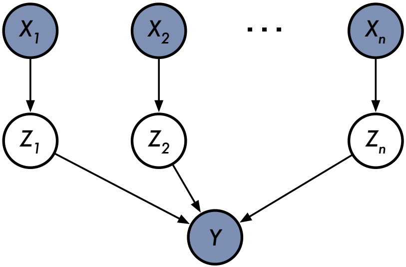

$x_i \in \{0, 1\}$

$$P(Y = 1 | x_1, x_2, \ldots, x_n) = 1 - \prod_{i=1}^{n} (1 - p_i)^{x_i}$$

The log (conditional) likelihood is $\sum_t \log P(y_t | x_t)$.
How to estimate parameters $p_i \in [0, 1]$ that maximize this?

EM — but how? Isn't the data complete?

# EM for noisy-OR




$x_i \in \{0, 1\}$

$z_i \in \{0, 1\}$
$P(Z_i = 1 | x_i) = p_i x_i$

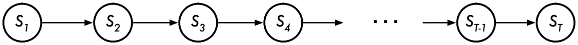
$$P(Y = 1 | Z_1, \ldots, Z_n) = \mathbf{OR}(Z_1, \ldots, Z_n)$$

HW 5

First you will show that this model is equivalent to noisy-OR.
Then you will derive the EM updates for $p_i \in [0, 1]$.

# Hidden Markov Models

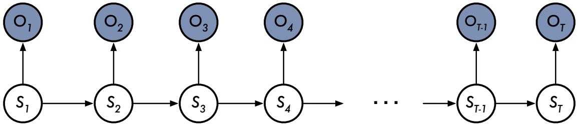
# Markov Models (Review)




Two simplifying assumptions:

1. Finite Context
2. Position Invariance

# Hidden Markov models (HMMs)




- Random variables

  $S_t \in \{1, 2, \ldots, n\}$ hidden state at time $t$

  $O_t \in \{1, 2, \ldots, m\}$ observation at time $t$

- States versus observations

  Each observation $O_t$ is a noisy, partial reflection of the true underlying (but hidden) state $S_t$ of the world at time $t$.

  What makes this model so useful?

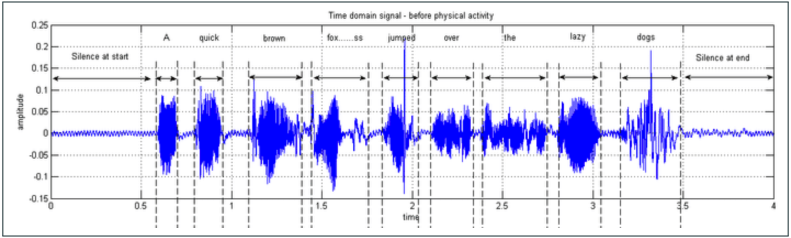
# Housetraining a puppy

This is Bubbles.
She's an english spanador.

$O_t \in \{\texttt{sleeping}, \texttt{eating}, \texttt{barking}, \texttt{waiting by door}, \texttt{etc.}\}$
$S_t \in \{\texttt{playful}, \texttt{hungry}, \texttt{tired}, \texttt{ready to burst}\}$

Does she need to go outside?
What is $P(s_t | o_1, o_2, \ldots, o_t)$?

# Speech recognition



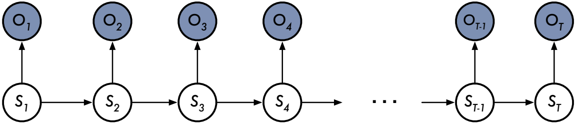

$O_t$ is the acoustic feature vector for windowed speech at time $t$.
$S_t$ is the unit of language (e.g., phoneme) being uttered at time $t$.

What did I just hear?
What is $\operatorname{argmax}_{s_1,s_2,\ldots,s_T} P(s_1, s_2, \ldots, s_T | o_1, o_2, \ldots, o_T)$?

$O_t$ encodes the sensor readings at time $t$.
$S_t$ encodes the robot location at time $t$.

**Location in the room: what is $P(s_t|o_1, o_2, \ldots, o_t)$?**

# HMMs as belief networks




Q. Which of the following statements are True?

A. $P(S_t|S_1, S_2, \ldots, S_{t-1}) = P(S_t|S_{t-1})$

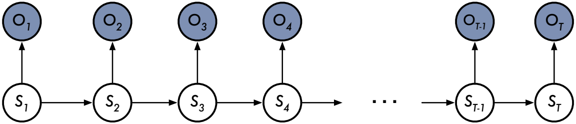
B. $P(O_t|S_1, S_2, \ldots, S_t) = P(O_t|S_t)$

C. $P(S_t|S_{t-1}) = P(S_t|S_{t-1}, O_t)$

D. A and B

E. A, B and C

# HMMs as belief networks




- Conditional independence assumptions

$$P(S_t|S_1, S_2, \ldots, S_{t-1}) = P(S_t|S_{t-1})$$
$$P(O_t|S_1, S_2, \ldots, S_T) = P(O_t|S_t)$$

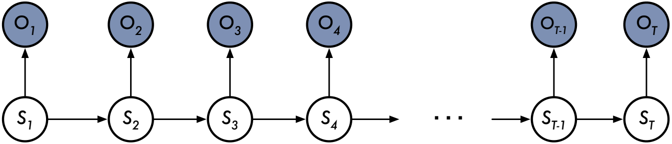
- CPTs are shared across time

$$P(S_t = s'|S_{t-1} = s) = P(S_{t+1} = s'|S_t = s)$$
$$P(O_t = o|S_t = s) = P(O_{t+1} = o|S_{t+1} = s)$$

- Joint distribution

$$P(\underbrace{S_1, \ldots, S_T}_{\vec{S}}, \underbrace{O_1, \ldots, O_T}_{\vec{O}}) = P(S_1)\, P(O_1|S_1) \prod_{t=2}^{T} \left[ P(S_t|S_{t-1})\, P(O_t|S_t) \right]$$

# Parameters of HMMs




Q. Which of the following is NOT a parameter of the model?

A. $P(S_t|S_{t+1})$

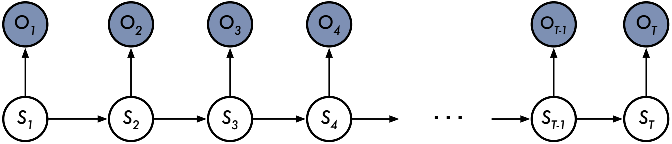
B. $P(S_1)$

C. $P(O_t|O_{t-1})$

D. $P(O_t|S_t)$

E. More than one of these is NOT a parameter of the model.

# Parameters of HMMs




$$a_{ij} = P(S_{t+1} = j | S_t = i)$$

$n \times n$ **transition matrix**

$$b_{ik} = P(O_t = k | S_t = i)$$

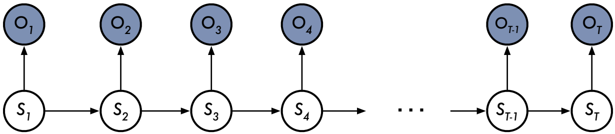
$n \times m$ **emission matrix**

$$\pi_i = P(S_1 = i)$$

$n \times 1$ **initial state distribution**

HMM is a polytree. **True or False?**

# Key computations in HMMs[1]




POLYTREE!

**Inference**

1. How to compute the likelihood $P(o_1, o_2, \ldots, o_T)$?
2. How to compute the most likely state sequence $\text{argmax}_{\vec{s}} P(\vec{s}|\vec{o})$?
3. How to update beliefs by computing $P(s_t|o_1, o_2, \ldots, o_t)$?

**Learning**

How to estimate parameters $\{\pi_i, a_{ij}, b_{ik}\}$ that maximize the log-likelihood of observed sequences?

[1]Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition.

That’s all folks!