CSE 250A. Principles of AI

Probabilistic Reasoning and Decision-Making

Lecture 12 – Latent variable models

Lawrence Saul
Department of Computer Science and Engineering
University of California, San Diego

Fall 2021

Outline

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**

$\boxed{\textbf{root nodes}}$

$$P_{\mathrm{ML}}(X_i = x) = \frac{\mathrm{count}(X_i = x)}{T} = \frac{1}{T} \sum_t I(x_{it}, x)$$

$\boxed{\textbf{nodes with parents}}$

$$P_{\mathrm{ML}}(X_i = x | \mathrm{pa}_i = \pi) = \frac{\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi)}{\mathrm{count}(\mathrm{pa}_i = \pi)} = \frac{\sum_t I(x_{it}, x)\, I(\mathrm{pa}_{it}, \pi)}{\sum_t I(\mathrm{pa}_{it}, \pi)}$$

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^{th} example

- **EM algorithm**

  Initialize CPTs to nonzero values.
  Repeat until convergence:

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

root nodes

$$P(X_i = x) \longleftarrow \frac{1}{T} \sum_t P(X_i = x | V_t = v_t)$$

nodes with parents

$$P(X_i = x | \mathrm{pa}_i = \pi) \longleftarrow \frac{\sum_t P(X_i = x, \mathrm{pa}_i = \pi | V_t = v_t)}{\sum_t P(\mathrm{pa}_i = \pi | V_t = v_t)}$$

Complete versus incomplete data

- **Complete data**

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

- **Incomplete data**

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

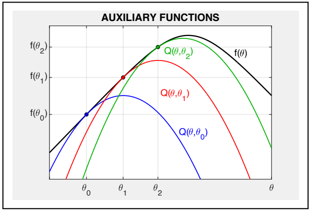
Key properties of EM

- **No learning rate**

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

- **Monotonic convergence**

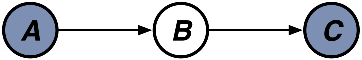
  Changes to CPTs from the EM updates always increase the incomplete-data log-likelihood $\mathcal{L} = \sum_t \log P(V_t = v_t)$.



Outline

1. Review
2. $\boxed{\text{Example}}$
3. Application
4. Preview

Example



Suppose that A and C are observed and B is hidden.

- **Inference**

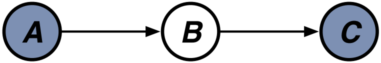
$$
\begin{aligned}
P(B=b|A=a, C=c) &= \frac{P(C=c|B=b, A=a)\, P(B=b|A=a)}{P(C=c|A=a)} & \boxed{\textbf{BR}} \\
&= \frac{P(C=c|B=b)\, P(B=b|A=a)}{P(C=c|A=a)} & \boxed{\textbf{CI}} \\
&= \frac{P(C=c|B=b)\, P(B=b|A=a)}{\sum_{b'} P(C=c|B=b')\, P(B=b'|A=a)} & \boxed{\textbf{normalized}}
\end{aligned}
$$

This is the only non-trivial posterior probability that we'll need for the EM updates in this example.

Log-likelihood

- **Incomplete data set**

t	A	B	C
1	a_1	?	c_1
2	a_2	?	c_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
T	a_T	?	c_T

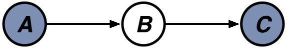


How to choose the CPTs to maximize the log-likelihood of this (incomplete) data?

- **Log-likelihood**

$$\begin{aligned}
\mathcal{L} &= \sum_t \log P(a_t, c_t) \\
&= \sum_t \log \sum_b P(a_t, b, c_t) && \boxed{\textbf{marginalization}} \\
&= \sum_t \log \sum_b P(a_t)\, P(b|a_t)\, P(c_t|a_t, b) && \boxed{\textbf{product rule}} \\
&= \sum_t \log \sum_b P(a_t)\, P(b|a_t)\, P(c_t|b) && \boxed{\textbf{conditional independence}}
\end{aligned}$$

EM update for $P(B|A)$



- **General form**

$$P(X_i = x | \mathrm{pa}_i = \pi) \longleftarrow \frac{\sum_t P(X_i = x, \mathrm{pa}_i = \pi | V_t = v_t)}{\sum_t P(\mathrm{pa}_i = \pi | V_t = v_t)}$$

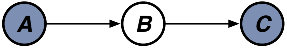
- **Update for this CPT**

$$P(B = b | A = a) \longleftarrow \frac{\sum_t P(B = b, A = a | A = a_t, C = c_t)}{\sum_t P(A = a | A = a_t, C = c_t)}$$

Simplify:

$$P(B = b | A = a) \longleftarrow \frac{\sum_t I(a, a_t) \overbrace{P(B = b | A = a_t, C = c_t)}^{\text{computed from Bayes rule}}}{\sum_t I(a, a_t)}$$

EM update for $P(C|B)$



- **General form**

$$P(X_i = x | \mathrm{pa}_i = \pi) \longleftarrow \frac{\sum_t P(X_i = x, \mathrm{pa}_i = \pi | V_t = v_t)}{\sum_t P(\mathrm{pa}_i = \pi | V_t = v_t)}$$

- **Update for this CPT**

$$P(C = c | B = b) \longleftarrow \frac{\sum_t P(C = c, B = b | A = a_t, C = c_t)}{\sum_t P(B = b | A = a_t, C = c_t)}$$

Simplify:

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

EM update for $P(A)$

$$A \longrightarrow B \longrightarrow C$$

- **General form**

$$P(X_i = x) \longleftarrow \frac{1}{T}\sum_t P(X_i = x | V_t = v_t) \qquad \boxed{\textbf{root node}}$$

- **Update for this CPT**

$$P(A = a) \longleftarrow \frac{1}{T}\sum_t P(A = a | A = a_t, C = c_t)$$

Simplify:

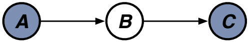
$$P(A = a) \longleftarrow \frac{1}{T}\sum_t I(a, a_t) = \frac{1}{T}\,\mathrm{count}(A = a)$$

The update reduces to the ML estimate for complete data—as it must, because A is observed and has no unobserved parents.

Summary of EM algorithm

- **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)

$$\begin{aligned} P(a) &= \frac{1}{T}\,\mathrm{count}(A{=}a) \\ P(b|a) &\longleftarrow \frac{\sum_t I(a, a_t)\,P(b|a_t, c_t)}{\sum_t I(a, a_t)} \\ P(c|b) &\longleftarrow \frac{\sum_t I(c, c_t)\,P(b|a_t, c_t)}{\sum_t P(b|a_t, c_t)} \end{aligned}$$

- **Convergence**

There are no learning rates to tune.
Each update increases the incomplete data log-likelihood:

$$\mathcal{L} = \sum_t \log \sum_b P(a_t)\,P(b|a_t)\,P(c_t|b)$$

Outline

1. **Review**
2. **Example**
3. $\boxed{\textbf{Application}}$
4. **Preview**

Application

- **Statistical language modeling**

  Let w_ℓ denote the ℓ^{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	$\prod_\ell P_1(w_\ell)$	$P_1(w) = \frac{\text{count}(w)}{L}$	$\boxed{w_1}\quad\boxed{w_2}\quad\cdots\quad\boxed{w_L}$
bigram	$\prod_\ell P_2(w_\ell \mid w_{\ell-1})$	$P_2(w' \mid w) = \frac{\text{count}(w \to w')}{\text{count}(w)}$	$\boxed{w_1}\to\boxed{w_2}\to\cdots\to\boxed{w_L}$
trigram	$\prod_\ell P_3(w_\ell \mid w_{\ell-1}, w_{\ell-2})$	$\vdots$	$\vdots$

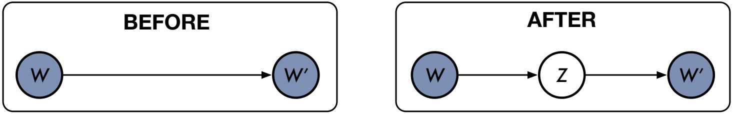
- **Evaluating *n*-gram models**

$$\begin{aligned} \text{Train on corpus } \mathcal{A} \quad &\Longrightarrow \quad P_1(\mathcal{A}) \leq P_2(\mathcal{A}) \leq P_3(\mathcal{A}) \ldots \\ \text{Test on corpus } \mathcal{B} \quad &\Longrightarrow \quad P_2(\mathcal{B}) = 0 \text{ if } \mathcal{B} \text{ has unseen bigrams.} \end{aligned}$$

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 $P(w'|w)$

$$w \longrightarrow z \longrightarrow w'$$

- **Inference**

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

- **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**

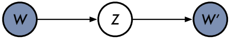
size of vocabulary	V	
number of clusters	C	
parameters in cluster model	$2CV$	$P(w'\|z), P(z\|w)$
parameters in bigram model	V^2	$P(w'\|w)$
parameters in unigram model	V	$P(w)$

- **Compact representations of complex worlds (lecture 1)**

  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**

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

- **M-step – Learning**

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

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

Experimental results

- **Data set**

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

- **Model** $w \longrightarrow z \longrightarrow w'$

  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

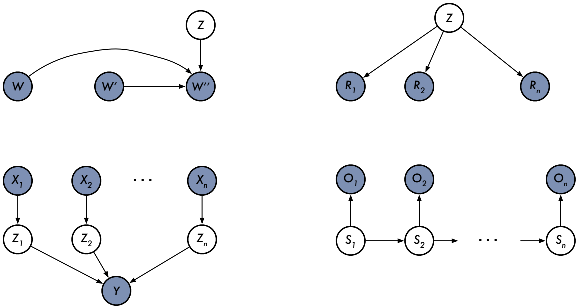
1	as cents made make take	19	billion hundred million nineteen
2	ago day earlier Friday Monday month quarter reported said Thursday trading Tuesday Wednesday ⟨...⟩	20	did ⟨"⟩ ⟨'⟩
3	even get to	21	but called San ⟨:⟩ ⟨start-of-sentence⟩
4	based days down home months up work years ⟨%⟩	22	bank board chairman 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.

Outline

1. **Review**
2. **Example**
3. **Application**
4. $\boxed{\textbf{Preview}}$

Preview



Many more examples of EM to come

- Linearly interpolated Markov models
- Noisy-OR models for medical diagnosis
- Naive Bayes models of recommender systems
- Hidden Markov models for sequence analysis