## Review

- ML estimation for complete data

  Examples $t = 1, 2, \ldots, T$

  Data $\{ X_1^{(t)}, \ldots, X_n^{(t)} \}_{t=1}^{T}$

  ML estimates for CPTs:

  - nodes w parents: $P_{ML}(X_i = x | pa = \pi) = \dfrac{\text{Count}(X_i = x, pa_i = \pi)}{\text{Count}(pa = \pi)}$

$$= \frac{\sum_t I(x_i^{(t)}, x)\, I(pa_i^{(t)}, \pi)}{\sum_t I(pa_i^{(t)}, \pi)}$$

  - root nodes: $P_{ML}(X_i = x) = \frac{1}{T} \text{Count}(X_i = x) = \frac{1}{T} \sum_t I(X_i^{(t)}, x)$

- ML estimate for Incomplete data

  Examples $t = 1, 2, \ldots T$

  Visible nodes $V^{(t)}$

  $\boxed{\text{EM algorithm}}$

  Initialize CPTs to non zero values

  Repeat until convergence.

  $\boxed{\text{E-Step}}$ - compute posterior (Inference)

$$P(X_i = x, pa_i = \pi | V^{(t)})$$

  $\boxed{\text{M-Step}}$ update CPTs (Learning)

  nodes w/ parents $P(X_i = x | pa_i = \pi) \leftarrow \dfrac{\sum_t P(X_i = x, pa_i = \pi | V^{(t)})}{\sum_t P(pa_i = \pi | V^{(t)})}$

  root nodes $P(X_i = x) \leftarrow \frac{1}{T} \sum_t P(X_i = x | V^{(t)})$

Algorithm converges monotonically to local maximum of $\mathcal{L} = \sum_t \log P(V^{(t)})$

$$\mathcal{L} = \sum_t \log P(V^{(t)}) = \sum_t \log \sum_h P(H=h, V^{(t)})$$

## Example #1

$$(A) \to (B) \to (C)$$

A & C observed
B hidden.

Posterior probability $P(B=b \mid A=a, C=c) = \dfrac{P(C=c \mid B=b, A=a)\, P(B=b \mid A=a)}{P(C=c \mid A=a)}$ $\boxed{\text{Bayes Rule}}$

$$= \frac{P(C=c \mid B=b)\, P(B=b \mid A=a)}{\sum_{b'} P(C=c \mid B=b')\, P(B=b' \mid A=a)}$$

- Incomplete data set

| t | A | B | C |
|---|---|---|---|
| 1 | $a_1$ | ? | $c_1$ |
| 2 | $a_2$ | ? | $c_2$ |
| ⋮ | | | |
| T | $a_T$ | ? | $c_T$ |

$\{(a_t, c_t)\}_{t=1}^{T}$

Log-likelihood $\mathcal{L} = \sum_t \log P(a_t, c_t)$

$$= \sum_t \log \sum_b P(a_t, b, c_t) \quad \boxed{\text{Marginalization}}$$

$$= \sum_t \log \left\{ \sum_b \left[ P(a_t) P(b \mid a_t) P(c \mid b) \right] \right\}$$

### M-Step update CPTs

Node B

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

Simplify: $\boxed{P(B=b \mid A=a) \leftarrow \dfrac{\sum_t I(a, a_t)\, P(B=b \mid A=a_t, C=c_t)}{\sum_t I(a, a_t)}}$

# CSE 250A 05-Nov-2019

**Node C**

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

Simplify:

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

**Node A**

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

Simplify: $\boxed{P(A=a) \leftarrow \frac{1}{T}\sum_t I(a,a_t) = \frac{\text{count}(A=a)}{T}}$

Reduces to ML estimate for complete data

**Application** Markov models of language

- Let $w_\ell$ denote word in corpus at text

How to model $P(w_1, w_2, \dots w_L)$?

| Model | $P(\vec{w})$ | ML estimate | DAG |
|---|---|---|---|
| unigram: | $\prod_\ell P_1(w_\ell)$ | $P_1(w) = \frac{\text{count}(w)}{L}$ | $w_1 \quad w_2 \ \dots \ w_L$ |
| bigram: | $\prod_\ell P_2(w_\ell \mid w_{\ell-1})$ | $P_2(w'\mid w) = \frac{\text{count}(w \to w')}{\text{count}(w)}$ | $w_1 \to w_2 \to \dots \to w_L$ |

- Evaluating n-gram models

train on corpus A: $P_1(\vec{w}_A) \le P_2(\vec{w}_A) \le P_3(\vec{w}_A) \ldots$

test on corpus B: $P_2(\vec{w}_B) = 0$ if unseen bigrams

$P_3(\vec{w}_B) = 0$ if unseen trigrams.

$\boxed{\text{Word clustering}}$

- Alternative to bigram model $w \rightarrow w'$

replace it with $w \rightarrow z \rightarrow w'$ words $w, w'$ observed cluster label $z$ hidden.

* CPTs in BN

$P(z|w)$ - prob that word $w$ is mapped into cluster $z$.

$P(w'|z)$ - prob that word in cluster $z$ is followed by word $w'$

* In cluster model:

$$\underbrace{P(w'|w)}_{V \times V \text{ matrix}} = \sum_z P(w', z|w) \quad \boxed{\text{Marginalization}}$$

$$= \sum_z P(z|w)\, P(w'|z, w) \quad \boxed{\text{Prod rule}}$$

$$= \sum_z P(z|w)\, P(w'|z) \quad \boxed{\text{CI}}$$ (Product of smaller matrices)

* Compact representations:

\# words in vocabulary: $V$

\# clusters : $C$

\# parameters in cluster model : $2CV$

\# bigram parameters: $V^2$

\# unigram parameters: $V$

Setting $C = 1$, we recover unigram model.

Setting $C = V$, we recover bigram model.

# CSE250A 5-Nov 2019

* Experimental results

$V = 60000$ vocabulary size

$L = 80$ million word corpus of WSJ articles.

$\text{count}(w \to w') = \sum_{\ell=1}^{L} I(w_\ell, w)\, I(w_{\ell+1}, w')$ is 99.8% sparse

$C = 32$ model trained by EM

$P(z|w)$ & $P(w'|z)$ = approx 4 million parameters.

Converges in ~30 iterations.

* What clusters are discovered.

For each word $w$,

what is $\operatorname{argmax}_z P(z|w)$?


* How to estimate $P(z|w)$ and $P(w'|z)$?

E-step:

$$P(z|w,w') = \frac{P(w'|z,w)\,P(z|w)}{P(w'|w)} \quad \boxed{\text{Bayes Rule}}$$

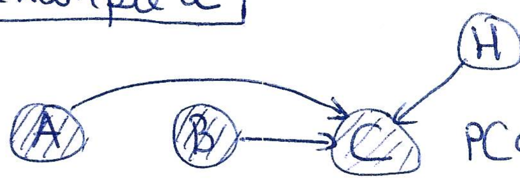
$$= \frac{P(w'|z)\,P(z|w)}{\sum_{z'} P(w'|z=z')\,P(z'|w)} \quad \boxed{\begin{array}{l}\text{CI}\\ \text{normalization}\end{array}}$$

Mstep: update CPTs

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

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

## Example 2




$P(C/A,B,H)$

Visible : $\{A, B, C\}$

Hidden : $H$

## Application


Trigram model


$z \in \{1, 2, 3\}$

chooses unigram
bigram
trigram

with weights $P(z=1), P(z=2), P(z=3)$