CSE 150 Lecture 10 [redacted]

[redacted]

Review

* Learn CPTs from incomplete data

$$P(X_i = x \mid pa_i = \pi) \leftarrow \frac{\sum_t P(X_i = x, pa_i = \pi \mid V^{(t)})}{\sum_t P(pa_i = \pi \mid V^{(t)})}$$

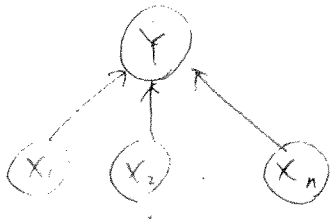
EM algorithm

* Ex: Simple chain with data $\{(a_t, c_t)\}_{t=1}^{T}$

$$P(b \mid a) \leftarrow \frac{\sum_t I(a, a_t) P(b \mid a_t, c_t)}{\sum_t I(a, a_t)}$$

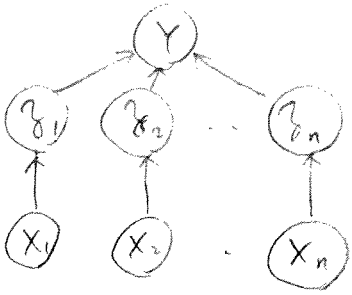
← computed from Bayes rule

* Ex: noisy-OR with data $\{(\vec{x}_t, \vec{y}_t)\}_{t=1}^{T}$



$$P(Y=1 \mid \vec{x}) = 1 - \prod_{i=1}^{n} (1-p_i)^{x_i}$$

* How to estimate noisy-OR parameters p_i?
* Alternative network:



$$P(z_i = 0 \mid x_i) = (1-p)^{x_i}$$

$$P(Y=1 \mid \vec{x}) = 1 - \prod_{i=1}^{n} (1-p_i)^{x_i}$$

equivalent to noisy-OR

* Conditional log-likelihood of data $\{(\vec{x}_t, y_t)\}_{t=1}^{T}$

$$\mathcal{L} = \sum_t \log P(y_t | \vec{x}_t)$$

$$= \sum_t \left[(1-y_t) \log P(y=0|\vec{x}_t) + y_t \log P(y=1|\vec{x}_t) \right]$$

$$= \sum_t \left[(1-y_t) \log \prod_{i=1}^{n} (1-p_i)^{x_{it}} + y_t \log \left(1 - \prod_{i=1}^{n} (1-p_i)^{x_{it}} \right) \right]$$

(data, data, parameters)

$$= \sum_t \left[(1-y_t) \sum_{i=1}^{n} x_{it} \log(1-p_i) + y_t \log \left(1 - \prod_{i=1}^{n} (1-p_i)^{x_{it}} \right) \right]$$

Note. Complicated, non-linear expression with respect to p_i!
EM to the rescue!

Shorthand: let T = total # examples.
let $T_i = \sum_{t=1}^{T} x_{it}$ (Count # times that $x_i = 1$)

EM update rule:

$$P(z_i=1 | x_i=1) \leftarrow \frac{\sum_t P(z_i=1, x_i=1 | \vec{X}=\vec{x}_t, Y=y_t)}{\sum_t P(x_i=1 | \vec{X}=\vec{x}_t, Y=y_t)}$$

Simplify:

$$P(z_i=1 | x_i=1) \leftarrow \frac{\sum_t I(x_{it}, 1) P(z_i=1 | \vec{x}_t, y_t)}{\sum_t I(x_{it}, 1)}$$

Recall from last lecture:

$$P(z_i=1 | \vec{x}, y) = \frac{y p_i x_i}{1 - \prod_{i=1}^{n} (1-p_i)^{x_i}}$$

EM update rule:

$$p_i \leftarrow \frac{1}{T_i} \sum_t I(x_{it}, 1) \left(\frac{y_t p_i x_{it}}{1 - \prod_{i=1}^{n} (1-p_i)^{x_{it}}} \right)$$

This update rule, applied in <u>parallel</u> to all $\{p_i\}_{i=1}^{n}$ will monotonically increase $\mathcal{L} = \sum_t \log(P_y | \vec{x}_t)$

Markov models of language

* Let w_ℓ = ℓ^{th} word in sentence.
  How to model $P(w_1, w_2, \ldots, w_L)$?
  Shorthand $\vec{w} = (w_1, w_2, \ldots, w_L)$.

Model	$P(\vec{w})$	ML estimate	DAG
unigram	$\prod_\ell P_1(w_\ell)$	$P_1(w) = \frac{\text{Count}(w)}{\sum_{w'} \text{Count}(w')}$	w_1 w_2 $\cdots$ w_L
bigram	$\prod_\ell P_2(w_\ell \mid w_{\ell-1})$	$P_2(w' \mid w) = \frac{\text{Count}(w \to w')}{\text{Count}(w)}$	$w_1 \to w_2 \to \cdots \to w_L$
⋮			

* Evaluating n-grams

  Train on corpus A: $P_1(\vec{w}) \leq P_2(\vec{w})$ on corpus A
  Test on corpus B: $P_1(\vec{w}) \geq P_2(\vec{w})$ if $P_2(\vec{w}) = 0$ (there are unseen bigrams)

Linear interpretation

* Also known as mixture model

$$P_m(w_\ell \mid w_{\ell-1}) = \lambda P_1(w_\ell) + (1-\lambda) P_2(w_\ell \mid w_{\ell-1})$$

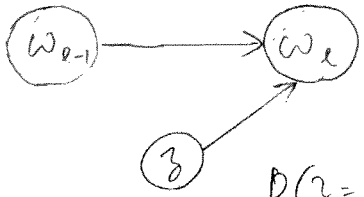
How to estimate λ?

* Methodology

  Train P_1, P_2 on corpus A — A = "training set"
  Fix P_1 and P_2 — B = "testing set"
  estimate λ on corpus C — C = "development set"
  Choose λ to maximize likelihood $\prod_\ell P_m(w_\ell \mid w_{\ell-1})$ on corpus C

* Don't <u>estimate</u> λ on corpus λ: this would yield $\lambda = 0$ (always favor P_2)!
  Do not <u>estimate</u> λ on corpus B: cheating!

* Hidden Variable model.



$$P(\omega_e|\omega_{e-1},\gamma) = \begin{cases} P_1(\omega_e) & \text{if } \gamma=1 \\ P_2(\omega_e|\omega_{e-1}) & \text{if } \gamma=2 \end{cases}$$

$\gamma \in \{1,2\}$ $\quad P(\gamma=1)=\lambda$, $P(\gamma=2)=1-\lambda$ } How to estimate on corpus C?

Hidden Variable: γ

Observed Variables: $\{(\omega_{e-1},\omega_e)\}_{e=1}^{L}$ corpus C

In This model:

$$P(\omega_e|\omega_{e-1}) = \sum_{\gamma=1}^{2} P(\omega_e,\gamma|\omega_{e-1})$$

$$= \sum_{\gamma} P(\omega_e|\gamma,\omega_{e-1})P(\gamma|\omega_{e-1}) \quad \text{product rule}$$

$$= \sum_{\gamma} P(\omega_e|\gamma,\omega_{e-1})P(\gamma) \quad \text{Conditional independence}$$

$$= \lambda P_1(\omega_e) + (1-\lambda)P_2(\omega_e|\omega_{e-1})$$

$$= P_m(\omega_e|\omega_{e-1}) \quad \text{matches linear interpolation.}$$

* E-step: Compute posterior probability

$$P(\gamma|\omega_{e-1},\omega_e) = \frac{P(\omega_e|\gamma,\omega_{e-1})P(\gamma|\omega_{e-1})}{P(\omega_e|\omega_{e-1})}$$ ← conditional independence; Bayes rule

$$P(\gamma=1|\omega_{e-1},\omega_e) = \frac{\lambda P_1(\omega_e)}{\lambda P_1(\omega_e)+(1-\lambda)P_2(\omega_e|\omega_{e-1})}$$

* M-step:

General update rule EM:

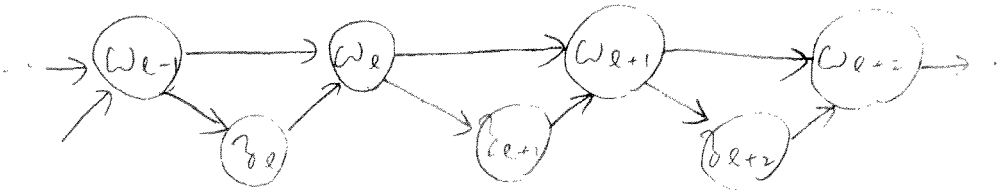
$$P(X_i=x|pa_i=\pi) \leftarrow \frac{\sum_t P(X_i=x, pa_i=\pi|V^{(t)})}{\sum_{x'}\sum_t P(X_i=x, pa_i=\pi|V^{(t)})}$$

In this problem:

$$P(\beta=1) \leftarrow \frac{\sum_\ell P(\beta=1 \mid \omega_{\ell-1}, \omega_\ell)}{\sum_{\beta'} \sum_\ell P(\beta=\beta' \mid \omega_{\ell-1}, \omega_\ell)}$$

$$\boxed{\lambda \leftarrow \frac{\sum_\ell P(\beta=1 \mid \omega_{\ell-1}, \omega_\ell)}{L}}$$

* Iterate EM:
  guaranteed improvement on corpus C of log-likelihood
  $\mathcal{L}(\lambda) = \sum_\ell \log P_m(\omega_\ell \mid \omega_{\ell-1})$
* In real-world application, smoothing parameter would depend on previous word $\omega_{\ell-1}$



$P(\beta=1 \mid \omega_{\ell-1}) = \lambda(\omega_{\ell-1})$ ← $\lambda(\omega)$ depends on previous word

$P(\beta=2 \mid \omega_{\ell-1}) = 1 - \lambda(\omega_{\ell-1})$

$P(\omega_\ell \mid \omega_{\ell-1}) = \lambda(\omega_{\ell-1}) P_1(\omega_\ell) + (1 - \lambda(\omega_{\ell-1})) P_2(\omega_\ell \mid \omega_{\ell-1})$

* EM used to estimate, say, $|V| \sim 10^5$ parameters
  ↑ vocabulary size.