CSE 150 Lecture 10 2.8.2010

* Quiz on Wed. (closed book)
* Review (Q/A) session Tue 1-2 Room 4140

$\boxed{\text{Review}}$

* Learn CPTs from incomplete data.

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

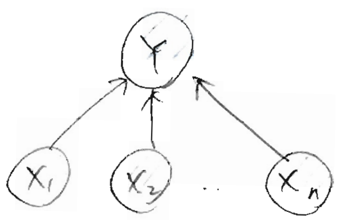
* Ex: Simple Chain with data $\{(a_t, c_t)\}_{t=1}^{T}$

$$A \to B \to C$$

Computed from Bayes rule

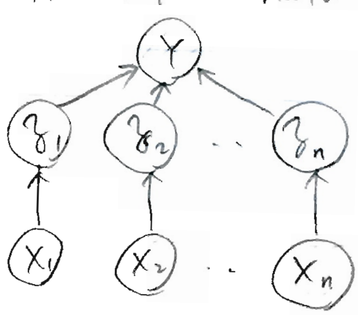
$$P(b \mid a) \leftarrow \frac{\sum_t I(a, a_t)\, P(b \mid a_t, c_t)}{\sum_t I(a, a_t)}$$

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



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

* 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 $\{(\vec{x}_t, y_t)\}_{t=1}^{T}$

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

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

$$= \sum_t \Big[\underset{\substack{\uparrow\\ \text{data}}}{(1-y_t)} \log \prod_{i=1}^{n} (1-p_i)^{x_{it}} + y_t \log\Big(1-\prod_{i=1}^{n}(1-p_i)^{x_{it}}\Big)\Big]$$

data data parameters

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

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:

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

Simplify:

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

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:

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

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

Markov models of language

* Let $w_\ell = \ell^{th}$ 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	$\prod_\ell P_1(w_\ell)$	$P_1(w) = \dfrac{\text{Count}(w)}{\sum_{w'} \text{Count}(w')}$	$w_1 \quad w_2 \quad \cdots \quad w_L$
bigram	$\prod_\ell P_2(w_\ell \mid w_{\ell-1})$	$P_2(w' \mid w) = \dfrac{\text{Count}(w \to w')}{\text{Count}(w)}$	$w_1 \to w_2 \to \cdots \to w_L$
⋮			

* Evaluating n-grams

  Train on Corpus A: $P_1(\vec{w}) \le P_2(\vec{w})$ on Corpus A
  Test on Corpus B: $P_1(\vec{w}) \ge P_2(\vec{w})$ if $P_2(\vec{w}) = 0$ (there are unseen bigrams).

Linear interpretation

* Also known as mixture model

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

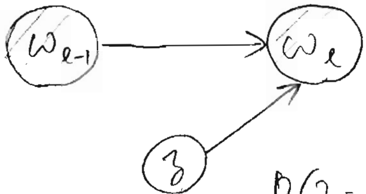
  How to estimate λ?

* 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 A: this would yield $\lambda = 0$ (always favor P_2)!
  Do not <u>estimate</u> λ on Corpus B: Cheating!

* Hidden Variable model.



$$P(w_l|w_{l-1}, z) = \begin{cases} P_1(w_l) & \text{if } z=1 \\ P_2(w_l|w_{l-1}) & \text{if } z=2 \end{cases}$$

$z \in \{1, 2\}$ $\left.\begin{array}{l} P(z=1) = \lambda \\ P(z=2) = 1-\lambda \end{array}\right\}$ How to estimate on corpus C?

Hidden Variable : z

Observed Variables : $\{(w_{l-1}, w_l)\}_{l=1}^{L}$ corpus C

In This model :

$$P(w_l|w_{l-1}) = \sum_{z=1}^{2} P(w_l, z| w_{l-1})$$

$$= \sum_z P(w_l| z, w_{l-1}) P(z|w_{l-1}) \quad \text{product rule}$$

$$= \sum_z P(w_l| z, w_{l-1}) P(z) \quad \text{Conditional independence}$$

$$= \lambda P_1(w_l) + (1-\lambda) P_2(w_l|w_{l-1})$$

$$= P_m(w_l|w_{l-1}) \quad \text{matches linear interpolation.}$$

* E-step: Compute posterior probability

$$P(z|w_{l-1}, w_l) = \frac{P(w_l|z, w_{l-1}) P(z|\cancel{w_{l-1}})}{P(w_l|w_{l-1})}$$

Conditional independence

Bayes rule.

$$P(z=1|w_{l-1}, w_l) = \frac{\lambda P_1(w_l)}{\lambda P_1(w_l) + (1-\lambda) P_2(w_l|w_{l-1})}$$

* 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:

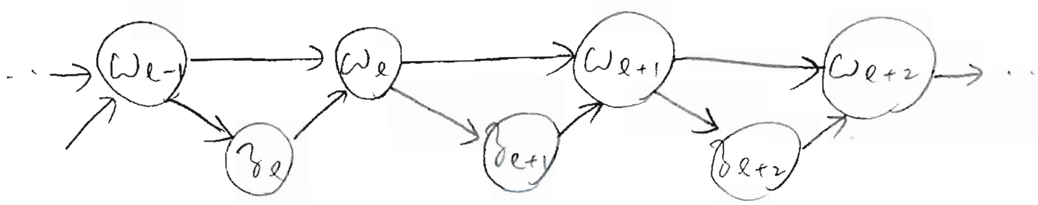
$$P(z=1) \longleftarrow \frac{\sum_\ell P(z=1 \mid w_{\ell-1}, w_\ell)}{\sum_{z'} \sum_\ell P(z=z' \mid w_{\ell-1}, w_\ell)}$$

$$\boxed{\lambda \longleftarrow \frac{\sum_\ell P(z=1 \mid w_{\ell-1}, w_\ell)}{L}}$$

* Iterate EM:
guaranteed improvement on corpus C of log-likelihood

$$\mathcal{L}(\lambda) = \sum_\ell \log P_m(w_\ell \mid w_{\ell-1})$$

* In real-world application, smoothing parameter would depend on previous word $w_{\ell-1}$.



$P(z=1 \mid w_{\ell-1}) = \lambda(w_{\ell-1})$ $\lambda(w)$ depends on previous word

$P(z=2 \mid w_{\ell-1}) = 1 - \lambda(w_{\ell-1})$

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

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