Review

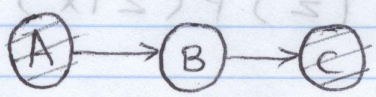
* Learning CPTs from incomplete data

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

E.M. Algorithm

* Example:

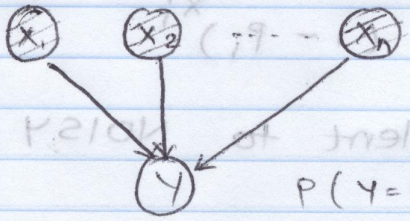
chain BN with data $\{(a_t, c_t)\}_{t=1}^{T}$



$$P(b|a) = \frac{\sum_{t=1}^{T} I(a, a_t) P(b|a_t, c_t)}{\sum_{t=1}^{T} I(a, a_t)}$$

← computed from Bayes rule and current CPTs

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

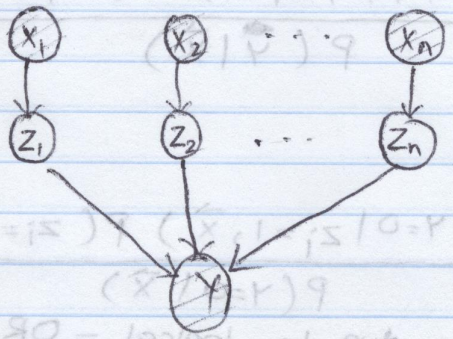


How to estimate p_i ?

$$P(Y=1 \mid X_1, X_2, \ldots, X_n) = 1 - \prod_{i=1}^{n} (1 - p_i)^{X_i}$$

with $p_i \in [0, 1]$

Consider alternate BN:



$$P(z_i = 1 \mid X_i = 1) = p_i$$

$$P(z_i = 1 \mid X_i = 0) = 0$$

$$Y = OR(X_1, X_2, \ldots, X_n)$$

Equivalently: $P(z_i=0|x_i) = (1-p_i)^{x_i} = \begin{cases} 1-p_i & \text{if } x_i=1 \\ 1 & \text{if } x_i=0 \end{cases}$

What is $P(Y=1|\vec{X})$ in this new model?

$$P(Y=1|\vec{X}) = \sum_{\vec{z}\in\{0,1\}^n} P(Y=1, \vec{z}\,|\vec{X}) \quad \text{marginalization}$$

$$= \sum_{\vec{z}} P(Y=1|\vec{z},\vec{x})\, P(\vec{z}|\vec{x}) \quad \text{product rule}$$

$$= \sum_{\vec{z}} P(Y=1|\vec{z})\, P(\vec{z}|\vec{x}) \quad \text{conditional independance}$$

$$= \sum_{\vec{z}\neq\{\vec{0}\}} (1)\, P(\vec{z}|\vec{x}) + 0 \cdot P(\vec{z}=\vec{0}|\vec{x})$$

$$= 1 - P(\vec{z}=\vec{0}|\vec{x}) \quad \text{due to normalization}$$

$$= 1 - \prod_{i=1}^{n} P(z_i=0|x_i) \quad \text{conditional independance}$$

$$= 1 - \prod_{i=1}^{n} (1-p_i)^{x_i}$$

Equivalent to NOISY-OR!

Inference for EM algorithm:
Compute posterior prob:

$$P(z_i=1|\vec{x},y) = \frac{P(y|\vec{x},z_i=1)\, P(z_i=1|\vec{x})}{P(y|\vec{x})}$$

Two cases:

$$P(z_i=1|\vec{x},y=0) = \frac{P(y=0|z_i=1,\vec{x})\, P(z_i=1|\vec{x})}{P(y=0|\vec{x})}$$

$= 0$ due to logical-OR

$$P(z_i=1 \mid \vec{x}, y=1) = \frac{P(y=1 \mid \vec{x}, z_i=1)\, P(z_i=1 \mid \vec{x})}{P(y=1 \mid \vec{x})}$$

(P_i if $x_i=1$, 0 if $x_i=0$)

$$= \frac{(1)\; p_i x_i}{1 - \prod_{i=1}^{n} (1-p_i)^{x_i}}$$

Combining:

$$P(z_i=1 \mid \vec{x}, y) = \frac{y\, p_i x_i}{1 - \prod_{i} (1-p_i)^{x_i}} \quad \text{valid for } y \in \{0,1\}$$

* How good is model?
Compute conditional log-likelihood of data $\{(\vec{x}_t, y_t)\}_{t=1}^{T}$

$$\mathcal{L} = \sum_{t=1}^{T} \log P(y_t \mid \vec{x}_t)$$

$$= \sum_{t=1}^{T} \left[(1-y_t) \log P(y=0 \mid \vec{x}_t) + y_t \log P(y=1 \mid \vec{x}_t) \right]$$

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

data: x_{it} — parameters to tune: p_i

Note: complicated, non-linear expression with respect to p_i!
Use EM to derive simple updates.
Shorthand: let T = total # examples
let $T_i = \sum_{t=1}^{T} x_{it}$ (count # times that $x_i=1$)

EM update rule:

$$P(z_i=1 \mid x_i=1) \leftarrow \frac{\sum_{t=1}^{T} P(z_i=1, x_i=1 \mid \vec{x}=\vec{x}_t, y=y_t)}{\sum_{t=1}^{T} P(x_i=1 \mid \vec{x}=\vec{x}_t, y=\vec{y}_t)}$$

Simplify:

$$P(z_i=1 \mid x_i=1) \leftarrow \frac{\sum_{t=1}^{T} I(x_{it},1)\, P(z_i=1 \mid \vec{x}_t, y_t)}{\sum_{t=1}^{T} I(x_{it},1)}$$

(computed from Bayes rule)

Final update:

$$p_i \leftarrow \left(\frac{1}{T_i}\right) \sum_{t=1}^{T} x_{it} \left(\frac{y_t\, p_i\, x_{it}}{1 - \prod_{j=1}^{n} (1-p_j)^{x_{jt}}} \right)$$

This update rule, applied in parallel to all p_i will monotonically increase $\mathcal{L} = \sum_t \log P(y_t \mid \vec{x}_t)$

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	$P(\vec{w})$	MLE estimate	DAG
unigram	$\prod_\ell P_1(w_\ell)$	$P_1(w) \propto \text{count}(w)$	w_1 w_2 ... w_ℓ
bigram	$\prod_\ell P_2(w_\ell \mid w_{\ell-1})$	$P_2(w' \mid w) \propto \text{count}(w \to w')$	$w_1 \to w_2 \to \cdots \to w_\ell$

* Evaluating n-gram models

  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 interpolation

* 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 value of λ?

* Methodology

  Train P_1, P_2 on corpus A = "training set"

  Fix P_1 and P_2

  Estimate λ on corpus C = "development set"

  Choose λ to maximize likelihood $\prod_\ell P_M(w_\ell \mid w_{\ell-1})$ on corpus C.

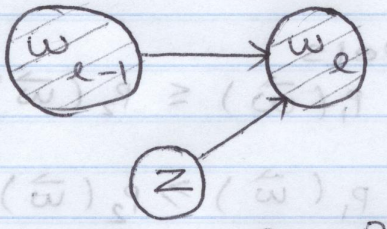
* What happens if we estimate λ on corpus A? This would yield $\lambda = 0$ always.

* corpus B = "testing set"

  Estimating λ on corpus B is cheating!

* How to estimate λ ?
Reformulate mixture model as hidden variable model where $\lambda \in [0,1]$ appears as value in some hidden node's CPT.

* Belief network



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

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

Hidden variable z
Observed variables w_t, w_{t-1}

In this model:

$$P(w_t | w_{t-1}) = \sum_{z=1}^{2} P(w_t, z | w_{t-1}) \quad \text{marginalization}$$

$$= \sum_{z=1}^{2} P(w_t | z, w_{t-1}) P(z | w_{t-1}) \quad \text{product rule}$$

$$= \sum_{z=1}^{2} P(w_t | z, w_{t-1}) P(z) \quad \text{conditional independence}$$

$$= \lambda P_1(w_t) + (1-\lambda) P_2(w_t | w_{t-1})$$

matches linear interpolation