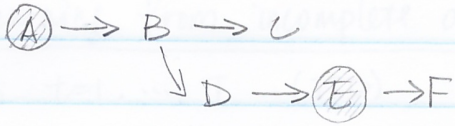
7-12

Review – Inference

* Tractable in polytrees — singly connected networks; no loops
* Key tools:
  - Bayes Rule
  - Marginalization
  - Product Rule

  (and conditionalized version of these)
  - Conditional Independence

Example



How to compute $P(B \mid E, A)$?

$$P(B \mid E, A) = \frac{P(E \mid B, A)\, P(B \mid A)}{P(E \mid A)}$$

Bayes rule = use it to rewrite LHS in terms of probabilities where nodes are conditioned on ancestors

$$P(E \mid B, A) = \sum_d P(E, D=d \mid B, A)$$

use marginalization to introduce parents b/c CPTs always express $P(\text{child} \mid \text{parents})$

$$P(E, D=d \mid B) = P(D=d \mid B)\, P(E \mid D=d, B)$$

product rule gets you even closer to CPTs, b/c CPTs predict one node at time.

conditional independence

Review – Learning from complete data

* examples $t = 1, 2, \dots, T$

  variables $X_1, X_2, \dots, X_n$

  data set $\{(x_1^t, x_2^t, \dots, x_n^t)\}_{t=1}^{T}$ (IID)

* choose CPTs to maximize log-likelihood $\mathcal{L} = \sum_t \log P(x_1^t, x_2^t, \dots, x_n^t)$

* ML estimates

$$P_{ML}(X_i = x \mid pa_i = \pi) = \frac{count(X_i = x, pa_i = \pi)}{count(pa_i = \pi)} = \frac{\sum_{t=1}^{T} I(x_i^t, x)\, I(pa_i, \pi)}{\sum_{t=1}^{T} I(pa_i, \pi)}$$

Review – Learning from incomplete data

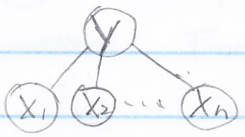
* examples $t = 1, \dots, T$ (IID)

  hidden nodes $H^{(t)}$

  visible nodes $V^{(t)}$

Why useful?

Ex: document clustering



$Y \in \{1, 2, \dots, K\}$ document topic $H^{(t)} = \{Y\}$

$X \in \{0, 1\}$ Does i^{th} word in dictionary appear in document? $V^{(t)} = \{X_1, X_2, \dots, X$

Question: can we learn such a model from a data set of unlabeled documents?

Data =

t	X_1	X_2	$\cdots$	X_n	Y
1	0	1	$\cdots$	1	?
2	0	0	$\cdots$	1	?
3	0	1	$\cdots$	0	?
$\vdots$					
T	1	0	$\cdots$	0	?

* Choose CPTs to maximize log-likelihood

$$\mathcal{L} = \sum_{t=1}^{T} \log P(V^{(t)})$$

$$= \sum_{t=1}^{T} \log \sum_{h} P(V^{(t)}, H=h)$$

$$= \sum_{t=1}^{T} \log \underbrace{\sum_{h} \prod_{i=1}^{n} P(X_i = x \mid pa_i = \pi)}_{\text{Much more difficult to compute and optimize}} \quad \Big| \text{ evaluate there } V^{(t)}, H=h$$

No "closed-form" solution for CPTs.

Alternative: seek an iterative solution.

Iterative solution is known as EM algorithm:

* EM algorithm:

1) Initialization

Choose values for elements of CPTs in BN

(should be valid probabilities, could be poor guesses)

2) Iterative steps — repeat until convergence.

(E) E-step — Inference

(expectation) For each example $t=1, 2, \ldots, T$ compute posterior distribution $P(H=h \mid V^{(t)})$

In particular, for each node, compute

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

(M) M-step: update CPTs to increase log-likelihood.

(maximization)

$$P(X_i = x \mid pa_i = \pi) \leftarrow \frac{\sum_{t=1}^{T} P(X_i = x, pa_i = \pi \mid V^{(t)})}{\sum_{x'} \left[\sum_{t=1}^{T} P(X_i = x', pa_i = \pi \mid V^{(t)}) \right]}$$

Simplify:

① nodes w/ parents:

$$P(X_i = x \mid pa_i = \pi) \leftarrow \frac{\sum_{t=1}^{T} P(X_i = x, pa_i = \pi \mid V^{(t)})}{\sum_{t=1}^{T} P(pa_i = \pi \mid V^{(t)})}$$

② root node (no parents)

$$P(X_i = x) \leftarrow \frac{1}{T} \sum_{t=1}^{T} P(X_i = x \mid V^{(t)})$$

Note:

1) RHS of these update depends on current values of CPTs

2) Inference is key subroutine of learning.

Intuition:

- Use posterior distribution $P(H \mid V^{(t)})$ to "fill in" missing values of hidden nodes
- expected statistics of posterior distribution $P(H \mid V^{(t)})$ substitute for observed counts that we had in complete data case
- Compare to ML estimate in complete data case

$$P_{ML}(X_i = x \mid pa_i = \pi) = \frac{count(X_i = x, pa_i = \pi)}{count(pa_i = \pi)} = \frac{\sum_{t=1}^{T} I(X_i^{(t)}, x)\, I(pa_i^{(t)}, \pi)}{\sum_{t=1}^{T} I(pa_i^{(t)}, \pi)}$$

↳ special case of EM when all data is observed.

Key properties of Expectation Maximization (EM) algorithm

1) Monotonic convergence

Each iteration of EM improves log-likelihood

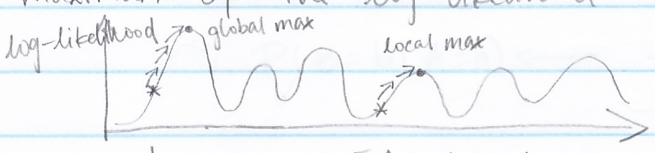
$$\mathcal{L} = \sum_{t=1}^{T} \log P(V^{(t)})$$

If $\mathcal{L}_k$ is log-likelihood at k^{th} iteration, then

$$\mathcal{L}_k \geq \mathcal{L}_{k-1}$$

↳ Really helpful debugging diagnostic

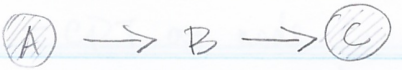
2) Converges in general to <u>local</u> (but not global) maximum of the log-likelihood $\mathcal{L} = \sum_t \log P(V^{(t)})$



Final output of EM depends on model initialization

3) No tuning parameters, no learning rates, no back-tracking

Example



A, C are observed (visible) nodes

B is hidden node

- First, consider inference in this BN.

Posterior: $P(B|A,C) = \frac{P(C|B,A)\,P(B|A)}{P(C|A)}$ Bayes Rule / conditional Independ

$= \frac{P(C|B)\,P(B|A)}{\sum_{b'} P(C|B=b')\,P(B=b'|A)}$ normalization

Bottom line: We can compute $P(b|a,c)$ in terms of model's CPTs.

* How do we learn from incomplete data $\{(a_t, c_t)\}_{t=}^{T}$

- EM algorithm

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

$= \sum_t \log \sum_b P(A=a_t, B=b, C=c_t)$

$= \sum_t \log \sum_b P(a_t)\,P(b|a_t)\,P(c_t|a_t,$

Reminder: general step of EM algorithm

$P(X_i = x | pa_i = \pi) \leftarrow \frac{\sum_{t=1}^{T} P(X_i = x, pa_i = \pi | V^{(t)})}{\sum_{t=1}^{T} P(pa_i = \pi | V^{(t)})}$

Now specialize to our example $V^{(t)} = (a_t, c_t)$

- CPT at node B:

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

Simplify:

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

(we've computed already in terms of CPTs; read off from data)

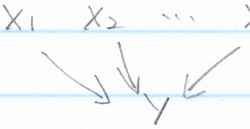
- CPT at node C

$$P(C=c \mid B=b) \leftarrow \frac{\sum_{t=1}^{T} P(B=b, C=c \mid a_t, c_t)}{\sum_{t=1}^{T} P(B=b \mid a_t, c_t)}$$

simplify:

$$P(C=c \mid B=b) \leftarrow \frac{\sum_{t=1}^{T} I(c, c_t)\, P(b \mid a_t, c_t)}{\sum_{t=1}^{T} P(b \mid a_t, c_t)}$$

Example Noisy-OR Model



diseases $X_i \in \{0,1\}$

symptom $Y \in \{0,1\}$

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

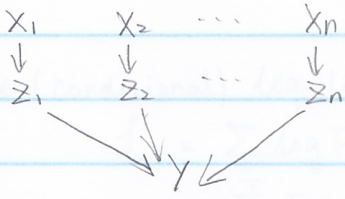
with $p_i \in [0,1]$

* From complete data $\{(\vec{X}_t, y_t)\}_{t=1}^{T}$ how do we estimate $p_i \in [0,1]$?

Note: Noisy-OR is a "parametric" model of CPT; there is no simple, closed-form ML estimate for $p_i \in [0,1]$ that maximizes

$$\sum_t \log P(Y = y^{(t)} \mid X_1 = x_1^{(t)}, \cdots, X_n = x_n^{(t)})$$

* Alternative formulation:



$X_i \in \{0,1\}$

$z_i \in \{0,1\}$ noisy copy of parent

$P(z_i=0 \mid x_i=0) = 1$
$P(z_i=1 \mid x_i=0) = 0$
$P(z_i=1 \mid x_i=1) = p_i$

$Y \in \{0,1\}$

$Y = OR(z_1, z_2, \cdots, z_n)$

$$P(Y=1 \mid z_1, \cdots, z_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} z_i > 0 \\ 0 & \text{if } z_i = 0 \text{ for all } i \end{cases}$$

$P(z_i=0 \mid x_i=0) = 1 \quad , \quad P(z_i=1 \mid x_i=0) = 0$

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

Equivalently:

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

What is $P(y=1 \mid \vec{x})$ in this new model?

$$\begin{aligned} P(y=1 \mid \vec{x}) &= \sum_{\vec{z} \in \{0,1\}^n} P(y=1, \vec{z} \mid \vec{x}) && \text{marginalization} \\ &= \sum_{\vec{z}} P(y=1 \mid \vec{z}, \vec{x}) P(\vec{z} \mid \vec{x}) && \text{product rule} \\ &= \sum_{\vec{z} \in \{0,1\}^n} P(y=1 \mid \vec{z}) P(\vec{z} \mid \vec{x}) && \text{cond. independence} \\ &= \sum_{\vec{z} \neq (0,0,0,\ldots,0)} P(\vec{z} \mid \vec{x}) + \underbrace{P(y=1 \mid \vec{z}=\vec{0})}_{0} P(\vec{z}=\vec{0} \mid \vec{x}) \end{aligned}$$

logical-OR, $y = \text{OR}(\vec{z})$

$$\begin{aligned} &= 1 - P(\vec{z}=\vec{0} \mid \vec{x}) && \text{normalization} \\ &= 1 - \prod_{i=1}^{n} P(z_i=0 \mid x_i) && \text{product rule e.I.} \\ &= 1 - \prod_{i=1}^{n} (1-p_i)^{x_i} \end{aligned}$$

Key insight:

alternative BN has same expression for $P(y=1 \mid \vec{x})$!!

→ estimate p_i parameters for noisy-OR CPT using EM in "expanded" network

First, let's do inference in this model:

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

$P(z_i=1 \mid \vec{x})$: $\begin{cases} 0 & \text{if } x_i=0 \\ p_i & \text{if } x_i=1 \end{cases}$

$P(y \mid z_i=1, \vec{x})$: $\begin{cases} 0 & \text{if } y=0 \\ 1 & \text{if } y=1 \end{cases}$

only care if $y=1$, b/c otherwise numerator vanish.

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

true for $x_i \in \{0,1\}$, $y \in \{0,1\}$

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

$$\begin{aligned} \mathcal{L} &= \sum_t \log P(y_t \mid \vec{x}_t) \\ &= \sum_{t=1}^{T} \left[y_t \log P(y=1 \mid \vec{x}_t) + (1-y_t) \log P(y=0 \mid \vec{x}_t) \right] \\ &= \sum_{t=1}^{T} \left[y_t \log \left(1 - \prod_{i=1}^{n} (1-p_i)^{x_i^{(t)}}\right) + (1-y_t) \log \prod_{i=1}^{n} (1-p_i)^{x_i^{(t)}} \right. \end{aligned}$$

$$\mathcal{L}(p_1, p_2, \cdots, p_n) = \sum_{t=1}^{T} y_t \log\left(1 - \prod_{i=1}^{n}(1-p_i)^{x_{it}}\right) + \sum_{t=1}^{T}\sum_{i=1}^{n}(1-y_t)\, x_{it} \log(1-p$$

y_t, x_{it} are data observed

p_i is parameter — how to choose?

Note:

Complicated nonlinear expression of p_i.

How to optimize? EM to the rescue

Shorthand: let T = total # of examples

let $T_i = \sum_{t=1}^{T} x_{it}$ = count $(x_i = 1)$

EM update rule for $p_i = P(z_i = 1 \mid x_i = 1)$

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

Simplify:

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

(x_{it} over $I(x_{it}, 1)$; have already from Bayes Rule over $P(z_i = 1 \mid \vec{x}_t, y_t)$; $\sum_{t=1}^{T} I(x_{it}, 1) = T_i$)

$$\frac{y_t \cdot p_i \cdot x_{it}}{1 - \prod_{i=1}^{n}(1-p_i)^{x_{it}}}$$

EM update for Noisy-OR

$$p_i \leftarrow \frac{p_i}{T_i} \sum_{t=1}^{T} \frac{y_t\, x_{it}}{1 - \prod_{i=1}^{n}(1-p_i)^{x_{it}}}$$

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

Recall from HW4:

$P_{unigram}(w)$

$P_{bigram}(w' \mid w)$

$P_{mixture}(w' \mid w) = (1-\lambda) P_{unigram}(w') + \lambda P_{bigram}(w' \mid w)$ where $\lambda \in [0,1]$

More generally:

$P_{mixture}(w' \mid w) = (1 - \lambda(w)) P_{unigram}(w') + \lambda(w) P_{bigram}(w' \mid w)$ where $\lambda(w) \in [0,1]$

Next Lecture: This is also a hidden variable model!

We can use EM to estimate λ.