Advanced Data Science

Slides Adapted from Tom M. Mitchell
Agenda

Topics Covered:
- Graphical models
- Learning
  - Learning from fully labeled data
  - Learning from partly observed data
  - EM
- Clustering
  - Mixture model clustering
- Learning Bayes Net structure
  - Chow-Liu for trees

Additional Reading:
- Bishop chapter 8, through 8.2
- Jordan “Graphical Models”
- Murphy “Intro to Graphical Models”
Learning of Bayes Nets

- Four categories of learning problems
  - Graph structure may be known/unknown
  - Variable values may be fully observed / partly unobserved

- Easy case: learn parameters for graph structure is *known*, and data is *fully observed*

- **Interesting case:** graph *known*, data *partly known*

- Gruesome case: graph structure *unknown*, data *partly unobserved*
EM and estimating $\theta$

Given observed set $X$, unobserved set $Y$ of boolean values

E step: Calculate for each training example, $k$
the expected value of each unobserved variable $Y$

$$E_{P(Y|X_1...X_N)}[y(k)] = P(y(k) = 1|x_1(k), ... x_N(k); \theta) = \frac{P(y(k) = 1) \prod_i P(x_i(k)|y(k) = 1)}{\sum_{j=0}^{1} P(y(k) = j) \prod_i P(x_i(k)|y(k) = j)}$$

M step: Calculate estimates similar to MLE, but
replacing each count by its expected count

let's use $y(k)$ to indicate value of $Y$ on $k$th example
EM and estimating $\theta$

Given observed set $X$, unobserved set $Y$ of boolean values

E step: Calculate for each training example, $k$

the expected value of each unobserved variable $Y$

$$E_{P(Y|X_1...X_N)}[y(k)] = P(y(k) = 1|x_1(k),...x_N(k); \theta) = \frac{P(y(k) = 1) \prod_i P(x_i(k)|y(k) = 1)}{\sum_{j=0}^1 P(y(k) = j) \prod_i P(x_i(k)|y(k) = j)}$$

M step: Calculate estimates similar to MLE, but replacing each count by its expected count

$$\theta_{ij|m} = \hat{P}(X_i = j|Y = m) = \frac{\sum_k P(y(k) = m|x_1(k)...x_N(k)) \delta(x_i(k) = j)}{\sum_k P(y(k) = m|x_1(k)...x_N(k))}$$

MLE would be:

$$\hat{P}(X_i = j|Y = m) = \frac{\sum_k \delta((y(k) = m) \land (x_i(k) = j))}{\sum_k \delta(y(k) = m)}$$
- **Inputs:** Collections $\mathcal{D}^l$ of labeled documents and $\mathcal{D}^u$ of unlabeled documents.
- Build an initial naive Bayes classifier, $\hat{\theta}$, from the labeled documents, $\mathcal{D}^l$, only. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg \max_\theta P(\mathcal{D}|\theta)P(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve, as measured by the change in $l_c(\theta|\mathcal{D}; z)$ (the complete log probability of the labeled and unlabeled data, and the prior) (see Equation 10):
  - (E-step) Use the current classifier, $\hat{\theta}$, to estimate component membership of each unlabeled document, i.e., the probability that each mixture component (and class) generated each document, $P(c_j|d_i; \hat{\theta})$ (see Equation 7).
  - (M-step) Re-estimate the classifier, $\hat{\theta}$, given the estimated component membership of each document. Use maximum a posteriori parameter estimation to find $\hat{\theta} = \arg \max_\theta P(\mathcal{D}|\theta)P(\theta)$ (see Equations 5 and 6).
- **Output:** A classifier, $\hat{\theta}$, that takes an unlabeled document and predicts a class label.

From [Nigam et al., 2000]
Unsupervised clustering

Just extreme case for EM with zero labeled examples…
Clustering

- Given set of data points, group them
- Unsupervised learning
- Which patients are similar? (or which earthquakes, customers, faces, web pages, …)
Mixture Distributions

Model joint $P(X_1 \ldots X_n)$ as mixture of multiple distributions. Use discrete-valued random variable $Z$ to indicate which distribution is being use for each random draw.

So

$$P(X_1 \ldots X_n) = \sum_i P(Z = i) \ P(X_1 \ldots X_n | Z)$$

Mixture of Gaussians:

• Assume each data point $X=<X_1, \ldots X_n>$ is generated by one of several Gaussians, as follows:

• Assume each data point, $x$, is generated by 2-step process

  1. randomly choose Gaussian $i$, according to $P(Z=i)$
  2. randomly generate a data point $<x_1,x_2 \ldots x_n>$ according to $N(\mu_i, \Sigma_i)$
EM for Mixture of Gaussian Clustering

Let’s simplify to make this easier:

1. Assume \( X = \langle X_1 \ldots X_n \rangle \), and the \( X_i \) are conditionally independent given \( Z \) (our data points are independent given the cluster)

   \[
P(X|Z = j) = \prod_i N(X_i | \mu_{ji}, \sigma_{ji})
   \]

2. Assume only 2 clusters (values of \( Z \)), and \( \forall i, j, \sigma_{ji} = \sigma \)

   \[
P(X) = \sum_{j=1}^{2} P(Z = j | \pi) \prod_i N(x_i | \mu_{ji}, \sigma)
   \]

   Value of \( Z \)

3. Assume \( \sigma \) known, \( \pi_1 \ldots \pi_K, \mu_{i1} \ldots \mu_{Ki} \) unknown

   For training example \( k \) probability to belong to a \( Z=i \)
   For training example \( k \) the \( \mu \) for each cluster

Observed: \( X = \langle X_1 \ldots X_n \rangle \)

Unobserved: \( Z \)
Given observed variables $X$, unobserved $Z$
Define $Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')]$
where $\theta = (\pi, \mu_{ji})$

Iterate until convergence:
• E Step: Calculate $P(Z(n)|X(n), \theta)$ for each example $X(n)$. Use this to construct $Q(\theta'|\theta)$
• M Step: Replace current $\theta$ by
  $$\theta \leftarrow \text{arg max}_{\theta'} Q(\theta'|\theta)$$
EM – E Step

Calculate $P(Z(n)|X(n), \theta)$ for each observed example $X(n)$

$X(n)=<x_1(n), x_2(n), \ldots x_T(n)>.$

$$P(z(n) = k|x(n), \theta) = \frac{P(x(n)|z(n) = k, \theta) \cdot P(z(n) = k|\theta)}{\sum_{j=0}^{1} p(x(n)|z(n) = j, \theta) \cdot P(z(n) = j|\theta)}$$

$$P(z(n) = k|x(n), \theta) = \frac{[\prod_i P(x_i(n)|z(n) = k, \theta)] \cdot P(z(n) = k|\theta)}{\sum_{j=0}^{1} \prod_i P(x_i(n)|z(n) = j, \theta) \cdot P(z(n) = j|\theta)}$$

$$P(z(n) = k|x(n), \theta) = \frac{[\prod_i N(x_i(n)|\mu_{k,i}, \sigma)] \cdot (\pi^k (1 - \pi)^{(1-k)})}{\sum_{j=0}^{1} [\prod_i N(x_i(n)|\mu_{j,i}, \sigma)] \cdot (\pi^j (1 - \pi)^{(1-j)})}$$
First consider update for $\pi = P(Z = 1)$

$$Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')] = E[\log P(X|Z, \theta') + \log P(Z|\theta')]$$

$\pi \leftarrow \arg \max_{\pi'} E_{Z|X,\theta}[\log P(Z|\pi')]$

$$E_{Z|X,\theta}[\log P(Z|\pi')] = E_{Z|X,\theta}\left[ \log \left( \pi' \sum_n z(n) (1 - \pi') \sum_n (1 - z(n)) \right) \right]$$

$$= E_{Z|X,\theta}\left[ \left( \sum_n z(n) \right) \log \pi' + \left( \sum_n (1 - z(n)) \right) \log (1 - \pi') \right]$$

$$= \left( \sum_n E_{Z|X,\theta}[z(n)] \right) \log \pi' + \left( \sum_n E_{Z|X,\theta}[(1 - z(n))] \right) \log (1 - \pi')$$

$$\frac{\partial E_{Z|X,\theta}[\log P(Z|\pi')]}{\partial \pi'} = \left( \sum_n E_{Z|X,\theta}[z(n)] \right) \frac{1}{\pi'} + \left( \sum_n E_{Z|X,\theta}[(1 - z(n))] \right) \frac{(-1)}{1 - \pi'}$$

$$\pi \leftarrow \frac{\sum_{n=1}^{N} E[z(n)]}{(\sum_{n=1}^{N} E[z(n)]) + (\sum_{n=1}^{N} (1 - E[z(n)]))} = \frac{1}{N} \sum_{n=1}^{N} E[z(n)]$$
Now consider update for $\mu_{ji}$

$$Q(\theta' | \theta) = E_{Z|X,\theta} [\log P(X, Z|\theta')] = E[\log P(X|Z, \theta') + \log P(Z|\theta')]$$

$\mu_{ji}'$ has no influence

$$\mu_{ji} \leftarrow \arg \max_{\mu_{ji}'} E_{Z|X,\theta} [\log P(X|Z, \theta')]$$

$$\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta) \ x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|x(n), \theta)}$$

Compare above to MLE if $Z$ were observable:

$$\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} \delta(z(n) = j) \ x_i(n)}{\sum_{n=1}^{N} \delta(z(n) = j)}$$
EM – putting it together

Given observed variables X, unobserved Z

Define $Q(\theta' | \theta) = E_{Z|X,\theta} [\log P(X, Z|\theta')]$

where $\theta = \langle \pi, \mu_{j,i} \rangle$

Iterate until convergence:

• E Step: For each observed example $X(n)$, calculate $P(Z(n)|X(n), \theta)$

$$P(z(n) = k \mid x(n), \theta) = \frac{[\prod_i N(x_i(n)|\mu_{k,i},\sigma)] \ (\pi^k(1-\pi)^{1-k})}{\sum_{j=0}^{1} [\prod_i N(x_i(n)|\mu_{j,i},\sigma)] \ (\pi^j(1-\pi)^{1-j})}$$

• M Step: Update $\theta \leftarrow \arg\max_{\theta'} Q(\theta' | \theta)$

$$\pi \leftarrow \frac{1}{N} \sum_{n=1}^{N} E[z(n)]$$

$$\mu_{j,i} \leftarrow \frac{\sum_{n=1}^{N} P(z(n) = j|X(n), \theta) \ x_i(n)}{\sum_{n=1}^{N} P(z(n) = j|X(n), \theta)}$$
Mixture of Gaussians Demo

https://youtu.be/qMTuMa86NzU
What you should know about EM

- For learning from partly unobserved data
- MLEst of $\theta = \arg \max_\theta \log P(\text{data}|\theta)$
- EM estimate: $\theta = \arg \max_\theta E_{Z|X,\theta}[\log P(X, Z|\theta)]$
  Where $X$ is observed part of data, $Z$ is unobserved

- EM for training Bayes networks
- Can also develop MAP version of EM
- Can also derive your own EM algorithm for your own problem
  - write out expression for $E_{Z|X,\theta}[\log P(X, Z|\theta)]$
  - E step: for each training example $X^k$, calculate $P(Z^k|X^k, \theta)$
  - M step: chose new $\theta$ to maximize $E_{Z|X,\theta}[\log P(X, Z|\theta)]$
Learning Bayes Net Structure
How can we learn Bayes Net graph structure?

In general case, open problem
- can require lots of data (else high risk of overfitting)
- can use Bayesian methods to constrain search

One key result:
- Chow-Liu algorithm: finds “best” tree-structured network \( T(X) \)
- What’s best?
  - suppose \( P(X) \) is true distribution, \( T(X) \) is our tree-structured network, where \( X = \langle X_1, \ldots, X_n \rangle \)
  - Chou-Liu minimizes Kullback-Leibler divergence:

\[
KL(P(X) \parallel T(X)) = \sum_k P(X = k) \log \frac{P(X = k)}{T(X = k)}
\]
Chow-Liu Algorithm

**Key result:** To minimize $KL(P \parallel T)$, it suffices to find the tree network $T$ that maximizes the sum of mutual informations over its edges.

Mutual information for an edge between variable $A$ and $B$:

$$I(A, B) = \sum_a \sum_b P(a, b) \log \frac{P(a, b)}{P(a)P(b)}$$

This works because for tree networks with nodes $X \equiv \langle X_1 \ldots X_n \rangle$:

$$KL(P(X) \parallel T(X)) \equiv \sum_k P(X = k) \log \frac{P(X = k)}{T(X = k)}$$

By definition is:

$$= - \sum_i I(X_i, Pa(X_i)) + \sum_i H(X_i) - H(X_1 \ldots X_n)$$

Only 1 in tree

determines the quality of the approximation

independent of the dependency ordering in the tree (edges)
Chow-Liu Algorithm

Only the sum of the pairwise mutual information determines the quality of the approximation

If every branch (edge) on the tree is given a weight corresponding to the mutual information between the variables at its vertices,

Then the tree which provides the optimal approximation to the target distribution is just the maximum-weight tree.
Chow-Liu Algorithm

1. for each pair of vars A,B, use data to estimate $P(A,B)$, $P(A)$, $P(B)$

2. for each pair of vars A,B calculate mutual information

$$I(A, B) = \sum_a \sum_b P(a, b) \log \frac{P(a, b)}{P(a)P(b)}$$

3. calculate the maximum spanning tree over the set of variables, using edge weights $I(A,B)$
   (given N vars, this costs only $O(N^2)$ time)

4. add arrows to edges to form a directed-acyclic graph

5. learn the CPD’s for this graph
Chow-Liu algorithm example
Greedy Algorithm to find Max-Spanning Tree

Make a DAG as you wish!
Chow-Liu algorithm example
Greedy Algorithm to find Max-Spanning Tree

[Diagram showing the progression of the Chow-Liu algorithm with annotations indicating theInfo gain and the steps leading to a tree structure.]

[Courtesy A. Singh, C. Guestrin]
Bayes Nets – What You Should Know

• Representation
  – Bayes nets represent joint distribution as a DAG + Conditional Distributions
  – D-separation lets us decode conditional independence assumptions

• Inference
  – NP-hard in general
  – For some graphs, closed form inference is feasible
  – Approximate methods too, e.g., Monte Carlo methods, …

• Learning
  – Easy for known graph, fully observed data (MLE’s, MAP est.)
  – EM for partly observed data
  – Learning graph structure: Chow-Liu for tree-structured networks