Expectation Maximization, and Learning from Partly Unobserved Data

Recommended readings:

• Mitchell, Chapter 6.12

• "Text Classification from Labeled and Unlabeled Documents using EM", K.Nigam, et al., 2000. *Machine Learning,* 39. http://www.cs.cmu.edu/%7Eknigam/papers/emcat-mlj99.ps

Machine Learning 10-701
November 11, 2005

Tom M. Mitchell
Carnegie Mellon University

Outline

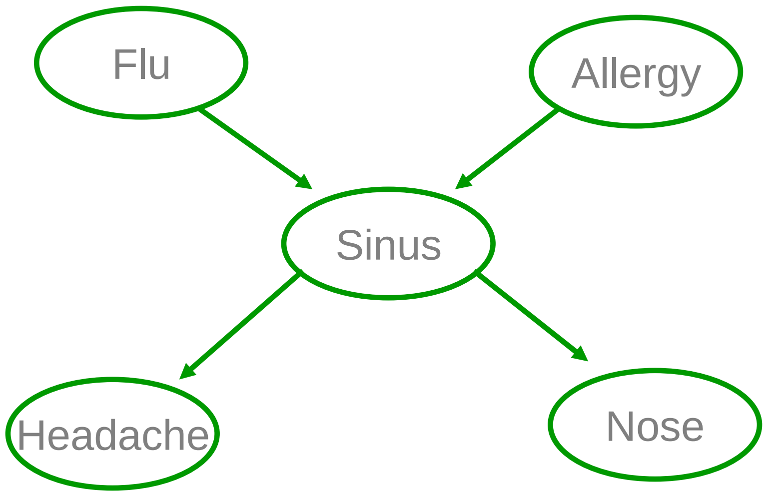
- EM_1: Learning Bayes network CPT's from partly unobserved data
- EM_2: Mixture of Gaussians – clustering
- EM: the general story
- Text application: learning Naïve Bayes classifier from labeled and unlabeled data

1. Learning Bayes net parameters from partly unobserved data

Learning CPTs from Fully Observed Data

- Example: Consider learning the parameter

$$\theta_{s|ij} \equiv P(S=1|F=i, A=j)$$



- MLE (Max Likelihood Estimate) is

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

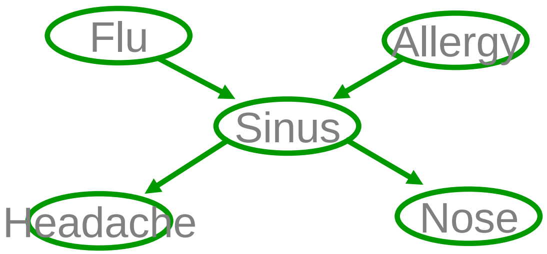
k^th training example

- Remember why?

MLE estimate of $\theta_{s|ij}$ from fully observed data

- Maximum likelihood estimate

$$\theta \leftarrow \arg\max_{\theta} \log P(data|\theta)$$



- Our case:

$$P(data|\theta) = \prod_{k=1}^{K} P(f_k, a_k, s_k, h_k, n_k)$$

$$P(data|\theta) = \prod_{k=1}^{K} P(f_k)P(a_k)P(s_k|f_k a_k)P(h_k|s_k)P(n_k|s_k)$$

$$\log P(data|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$$

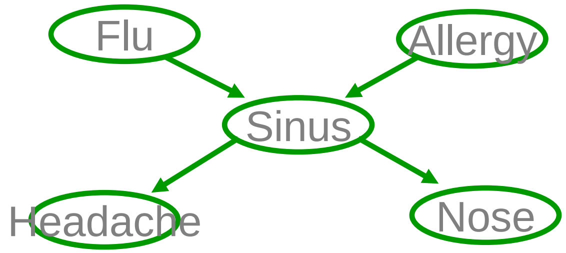
$$\frac{\partial \log P(data|\theta)}{\partial \theta_{s|ij}} = \sum_{k=1}^{K} \frac{\partial \log P(s_k|f_k a_k)}{\partial \theta_{s|ij}}$$

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

Estimate θ from partly observed data

- What if FAHN observed, but not S?
- Can't calculate MLE

$$\theta \leftarrow \arg\max_{\theta} \log \prod_k P(f_k, a_k, s_k, h_k, n_k|\theta)$$



- Let X be all *observed* variable values (over all examples)
- Let Z be all *unobserved* variable values
- Can't calculate MLE:

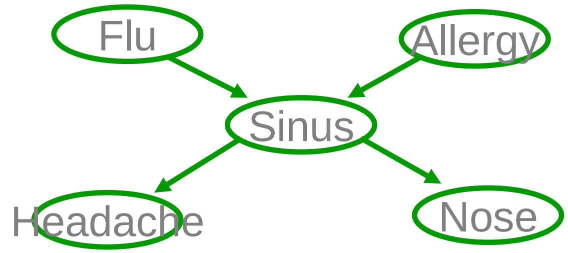
$$\theta \leftarrow \arg\max_{\theta} \log P(X, Z|\theta)$$

- EM seeks estimate:

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

- EM seeks estimate:

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



- here, observed X={F,A,H,N}, unobserved Z={S}

$$\log P(X,Z|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$$

$$E_{X|Z,\theta}[logP(X,Z|\theta)] = \sum_{k=1}^{K} \sum_{i=0}^{1} P(s_k = i|f_k, a_k, h_k, n_k)$$

$$[logP(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)]$$

EM Algorithm

EM is a general procedure for solving such problems

Given observed variables X, unobserved Z (X={F,A,H,N}, Z={S})

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

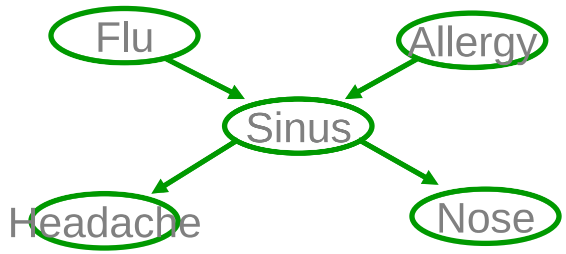
Iterate until convergence:

- E Step: Use X and current θ to estimate P(Z|X,θ)
- M Step: Replace current θ by

$$\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$$

Guaranteed to find local maximum. Each iteration increases $E_{Z|X,\theta}[\log P(X, Z|\theta)]$

E Step: Use X, θ, to Calculate P(Z|X,θ)

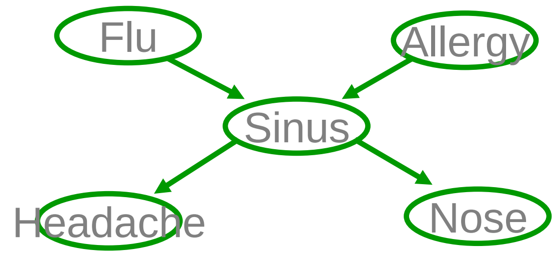


- How? Bayes net inference problem.

$$P(S_k = 1 | f_k a_k h_k n_k, \theta) =$$

$$P(S_k = 1 | f_k a_k h_k n_k, \theta) = \frac{P(S_k = 1, f_k a_k h_k n_k | \theta)}{P(S_k = 1, f_k a_k h_k n_k | \theta) + P(S_k = 0, f_k a_k h_k n_k | \theta)}$$

M step: modify this to achieve $\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$



- Maximum likelihood estimate

$$\theta \leftarrow \arg\max_{\theta} \log P(data|\theta)$$

- Our case:

$$P(data|\theta) = \prod_{k=1}^{K} P(f_k, a_k, s_k, h_k, n_k)$$

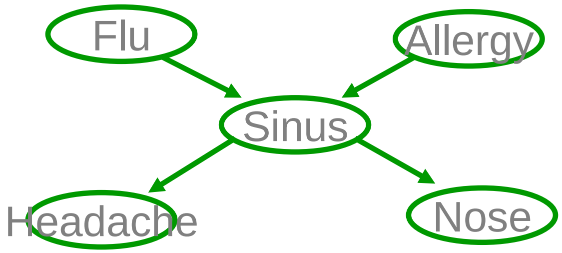
$$P(data|\theta) = \prod_{k=1}^{K} P(f_k)P(a_k)P(s_k|f_k a_k)P(h_k|s_k)P(n_k|s_k)$$

$$\log P(data|\theta) = \sum_{k=1}^{K} \log P(f_k) + \log P(a_k) + \log P(s_k|f_k a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)$$

$$\frac{\partial \log P(data|\theta)}{\partial \theta_{s|ij}} = \sum_{k=1}^{K} \frac{\partial \log P(s_k|f_k a_k)}{\partial \theta_{s|ij}}$$

$$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

EM and estimating $\theta_{s|ij}$



observed X = {F,A,H,N}, unobserved Z={S})

E step: Calculate for each training example, k

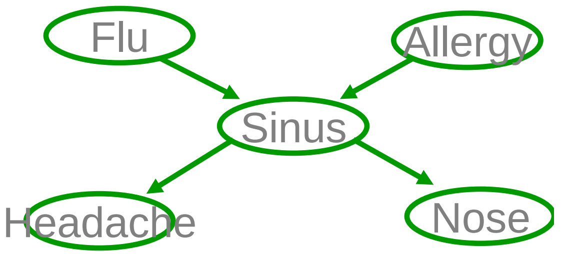
$$P(S_k = 1 | f_k a_k h_k n_k, \theta) = E[s_k] = \frac{P(S_k = 1, f_k a_k h_k n_k | \theta)}{P(S_k = 1, f_k a_k h_k n_k | \theta) + P(S_k = 0, f_k a_k h_k n_k | \theta)}$$

M step:

$$\theta_{s|ij} \leftarrow \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)\ E[s_k]}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

Recall MLE was: $$\theta_{s|ij} = \frac{\sum_{k=1}^{K} \delta(f_k = i, a_k = j, s_k = 1)}{\sum_{k=1}^{K} \delta(f_k = i, a_k = j)}$$

EM and estimating θ



More generally,
Given observed set X, unobserved set Z of boolean values

E step: Calculate for each training example, k

the expected value of each unobserved variable

M step:

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

$$\delta(Y=1) \rightarrow E_{Z|X,\theta}[Y] \qquad \delta(Y=0) \rightarrow (1 - E_{Z|X,\theta}[Y])$$

2. Usupervised clustering:
K-means and Mixtures of Gaussians

Clustering

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

K-means Clustering

Given data <x_1 … x_n>, and K, assign each x_i to one of K clusters, C_1 … C_K , minimizing $J = \sum_{j=1}^{K} \sum_{x_i \in C_j} ||x_i - \mu_j||^2$

Where μ_j is mean over all points in cluster C_j

<u>K-Means Algorithm</u>:

Initialize $\mu_1 \dots \mu_K$ randomly

Repeat until convergence:

1. Assign each point x_i to the cluster with the closest mean μ_j
2. Calculate the new mean for each cluster

$$\mu_j \leftarrow \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

K Means Applet

- Run K-means applet
  - http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html
- Try 3 clusters, 15 pts

Mixtures of Gaussians

K-means is EM'ish, but makes 'hard' assignments of x_i to clusters.

Let's derive a real EM algorithm for clustering.

What object function shall we optimize?

- Maximize data likelihood!

What form of P(X) should we assume?

- Mixture of Gaussians

Mixture of Gaussians:

- Assume $P(x)$ is a mixture of K different Gaussians
- Then each data point, x, is generated by 2-step process
  1. $z \leftarrow$ choose one of the K Gaussians, according to $\pi_1 \ldots \pi_{K-1}$
  2. Generate x according to the Gaussian $N(\mu_z, \Sigma_z)$

$$\boxed{P(\mathbf{x}) = \sum_{z=1}^{K} P(Z = z|\pi) N(\mathbf{x}|\mu_{\mathbf{z}}, \Sigma_z)}$$

Mixture Distributions

- $P(X|\phi)$ is a "mixture" of K different distributions: $P_1(X|\theta_1)$, $P_2(X|\theta_2)$, ... $P_K(X|\theta_K)$

  where $\phi = < \theta_1 \ldots \theta_K, \pi_1 \ldots \pi_{K-1} >$

- We generate a draw $X \sim P(X|\phi)$ in two steps:
  1. Choose $Z \in \{1, \ldots K\}$ according to $P(Z \mid \pi_1 \ldots \pi_{K-1})$
  2. Generate $X \sim P_k(X|\theta_k)$

$$\boxed{P(\mathbf{x}|\phi) = \sum_{k=1}^{K} P(Z = k|\pi) P_k(\mathbf{x}|\theta_\mathbf{k})}$$

EM for Mixture of Gaussians

Simplify to make this easier:

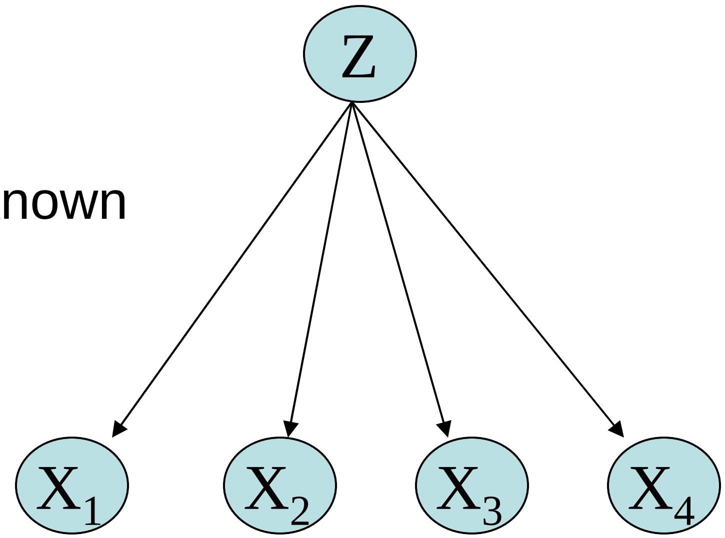
1. assume $X=\langle X_1 \ldots X_n \rangle$, and the X_i are conditionally independent given Z.

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

2. assume only 2 mixture components, and $\forall i, j, \sigma_{ji} = \sigma$

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

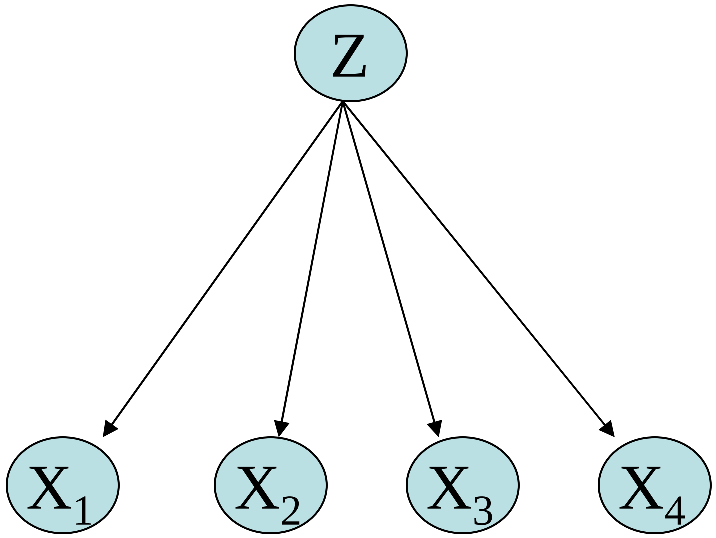
3. Assume σ known, $\pi_1 \ldots \pi_K, \mu_{1i} \ldots \mu_{Ki}$ unknown



Observed: $X=\langle X_1 \ldots X_n \rangle$

Unobserved: Z

EM



Given observed variables X, unobserved Z

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

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

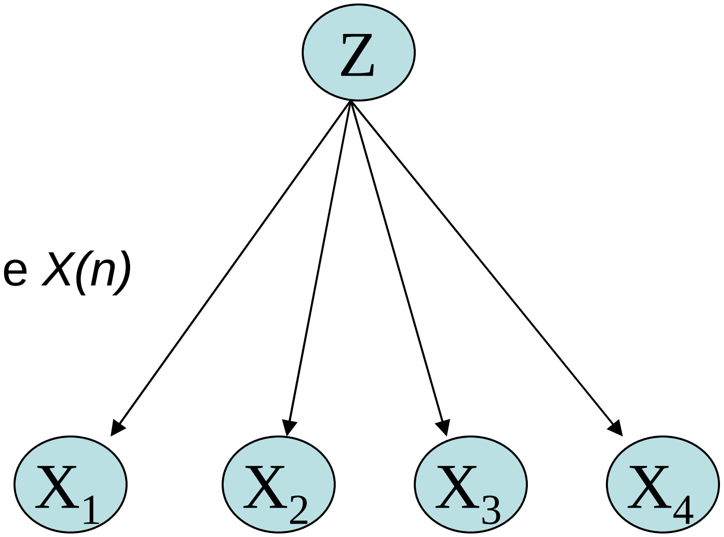
Iterate until convergence:

- 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 θ by

$$\theta \leftarrow \arg\max_{\theta'} Q(\theta'|\theta)$$

EM – E Step



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

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

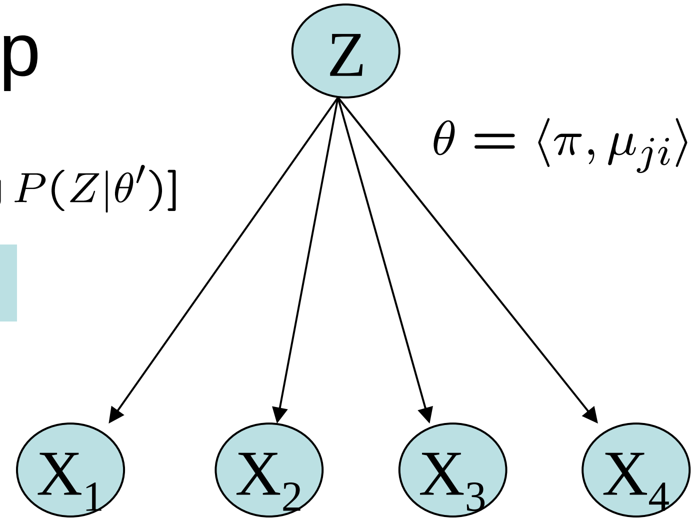
$$P(z(n)=k|x(n),\theta)=\frac{P(x(n)|z(n)=k,\theta)\quad P(z(n)=k|\theta)}{\sum_{j=0}^{1} p(x(n)|z(n)=j,\theta)\ P(z(n)=j|\theta)}$$

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

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

EM – M Step

First consider update for π



$$\theta = \langle \pi, \mu_{ji} \rangle$$

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

π' has no influence

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

Count z(n)=1

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

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

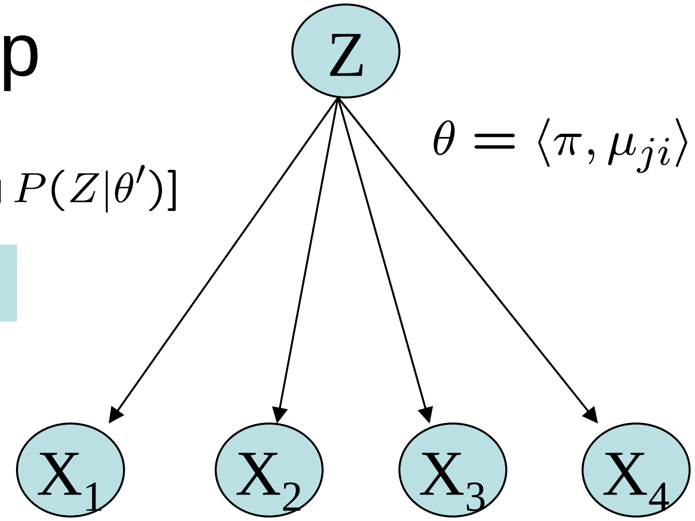
$$= \left(\sum_n E_{Z|X,\theta}[z(n)]\right)\log \pi' + \left(\sum_n E_{Z|X,\theta}[(1-z(n)])\right)\log(1-\pi')$$

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

$$\boxed{\pi \leftarrow \frac{\sum_{n=1}^N E[z(n)]}{\left(\sum_{n=1}^N E[z(n)]\right) + \left(\sum_{n=1}^N (1-E[z(n)])\right)} = \frac{1}{N}\sum_{n=1}^N E[z(n)]}$$

EM – M Step

Now consider update for μ_{ji}



$$\theta = \langle \pi, \mu_{ji} \rangle$$

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

μ_{ji}' has no influence

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

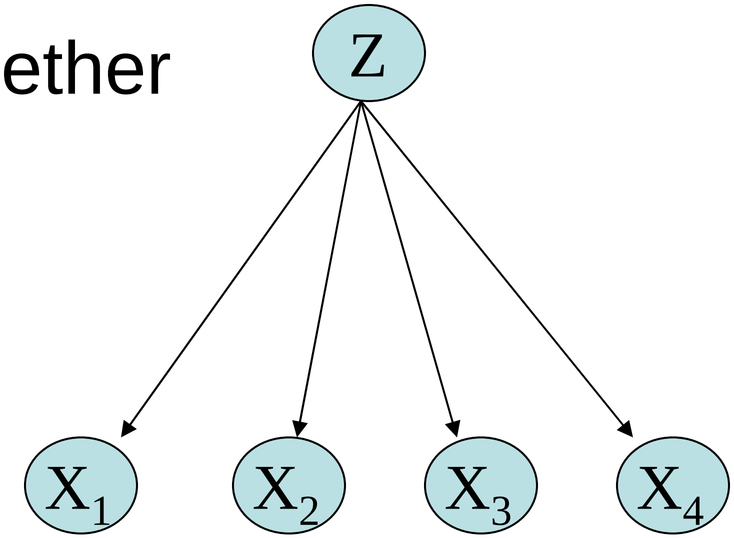
…
…
…

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

Compare above to MLE if Z were observable:

$$\mu_{ji} \leftarrow \frac{\sum_{n=1}^{N} \delta(z(n)=j) \quad x_i(n)}{\sum_{n=1}^{N} \delta(z(n)=j)}$$

EM – putting it together



Given observed variables X, unobserved Z

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

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

Iterate until convergence:

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

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

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

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

Mixture of Gaussians applet

- Run applet

http://www.neurosci.aist.go.jp/%7Eakaho/MixtureEM.html

K-Means vs Mixture of Gaussians

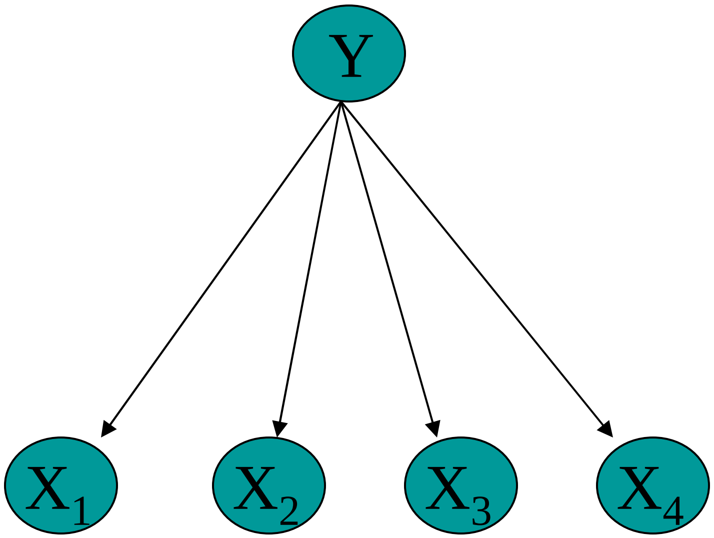
- Both are iterative algorithms to assign points to clusters

- Objective function
  - K Means: minimize $J = \sum_{j=1}^{K} \sum_{x_i \in C_j} ||x_i - \mu_j||^2$
  - MixGaussians: maximize $P(X|\theta)$

- Mixture of Gaussians is the more general formulation
  - Equivalent to K Means when $\Sigma_k = \sigma I$, and $\sigma \rightarrow 0$

Using Unlabeled Data to Help Train Naïve Bayes Classifier

Learn P(Y|X)



Y	X1	X2	X3	X4
1	0	0	1	1
0	0	1	0	0
0	0	0	1	0
?	0	1	1	0
?	0	1	0	1

- **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 $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (see Equations 5 and 6).
- Loop while classifier parameters improve, as measured by the change in $l_c(\theta|\mathcal{D}; \mathbf{z})$ (the complete log probability of the labeled and unlabeled data
  - **(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, $\mathrm{P}(c_j|d_i; \hat{\theta})$ (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 $\hat{\theta} = \arg\max_\theta \mathrm{P}(\mathcal{D}|\theta)\mathrm{P}(\theta)$ (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]

E Step:

$$\mathrm{P}(y_i = c_j | d_i; \hat{\theta}) = \frac{\mathrm{P}(c_j|\hat{\theta})\mathrm{P}(d_i|c_j;\hat{\theta})}{\mathrm{P}(d_i|\hat{\theta})}$$

$$= \frac{\mathrm{P}(c_j|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_j;\hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} \mathrm{P}(c_r|\hat{\theta}) \prod_{k=1}^{|d_i|} \mathrm{P}(w_{d_{i,k}}|c_r;\hat{\theta})}.$$

M Step:

w_t is t-th word in vocabulary

$$\hat{\theta}_{w_t|c_j} \equiv \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} N(w_t, d_i)\mathrm{P}(y_i = c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|\mathcal{D}|} N(w_s, d_i)\mathrm{P}(y_i = c_j|d_i)},$$

$$\hat{\theta}_{c_j} \equiv \mathrm{P}(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \mathrm{P}(y_i = c_j|d_i)}{|\mathcal{C}| + |\mathcal{D}|}.$$

Elaboration 1: Downweight the influence of unlabeled examples by factor λ

$$l_c(\theta|\mathcal{D};\mathbf{z}) = \log(\mathrm{P}(\theta)) + \sum_{d_i \in \mathcal{D}^l} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right)$$
$$+ \lambda \left(\sum_{d_i \in \mathcal{D}^u} \sum_{j=1}^{|\mathcal{C}|} z_{ij} \log\left(\mathrm{P}(c_j|\theta)\mathrm{P}(d_i|c_j;\theta)\right) \right).$$

Chosen by cross validation

New M step:

$$\hat{\theta}_{w_t|c_j} \equiv \mathrm{P}(w_t|c_j;\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \Lambda(i) N(w_t, d_i) \mathrm{P}(y_i = c_j|d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|\mathcal{D}|} \Lambda(i) N(w_s, d_i) \mathrm{P}(y_i = c_j|d_i)}.$$

$$\hat{\theta}_{c_j} \equiv \mathrm{P}(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \Lambda(i) \mathrm{P}(y_i = c_j|d_i)}{|\mathcal{C}| + |\mathcal{D}^l| + \lambda|\mathcal{D}^u|}$$

$$\Lambda(i) = \begin{cases} \lambda & \text{if } d_i \in \mathcal{D}^u \\ 1 & \text{if } d_i \in \mathcal{D}^l. \end{cases}$$

Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol D indicates an arbitrary digit.

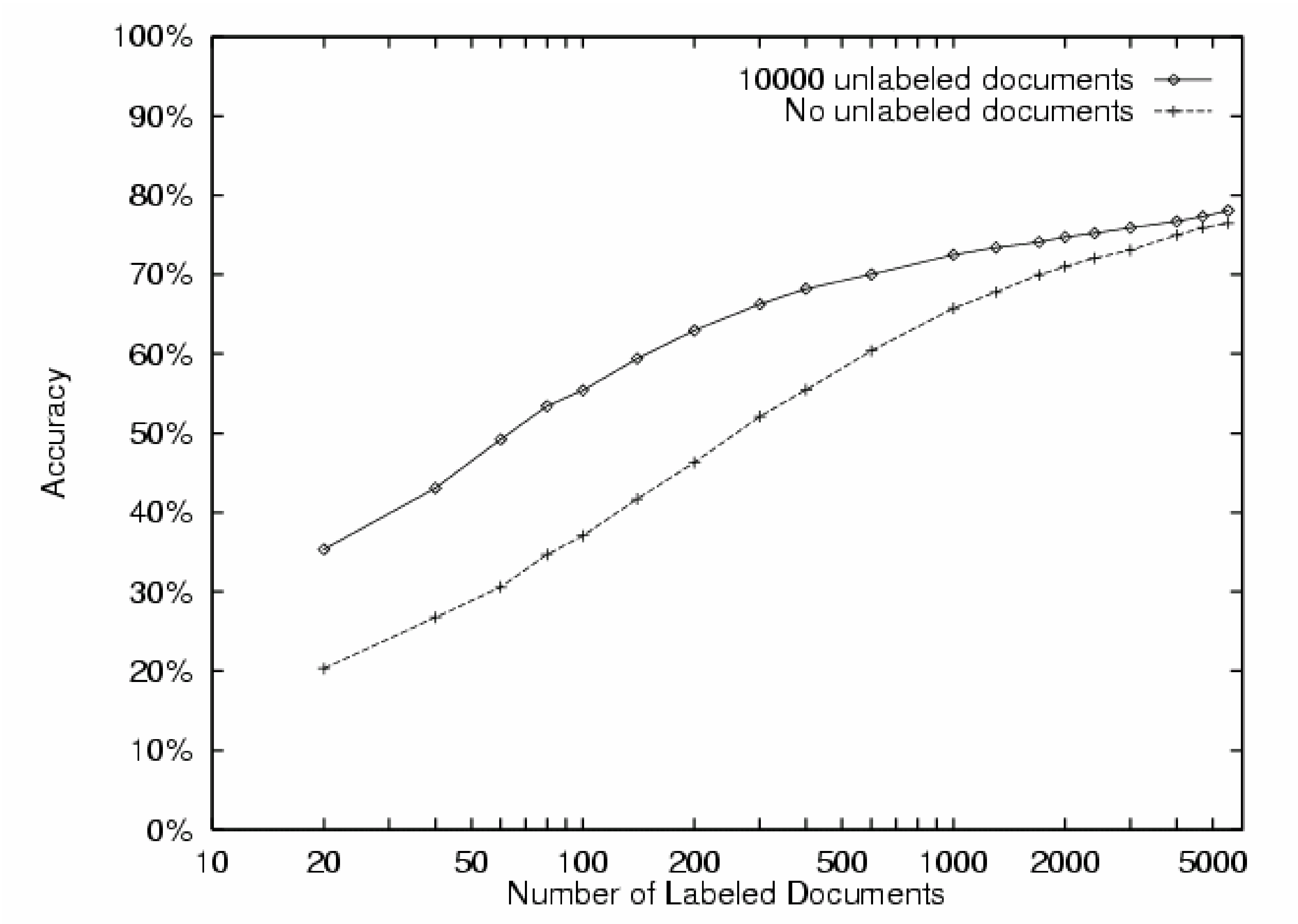
Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^{\star}$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^{\star}$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	DDam
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

Using one labeled example per class

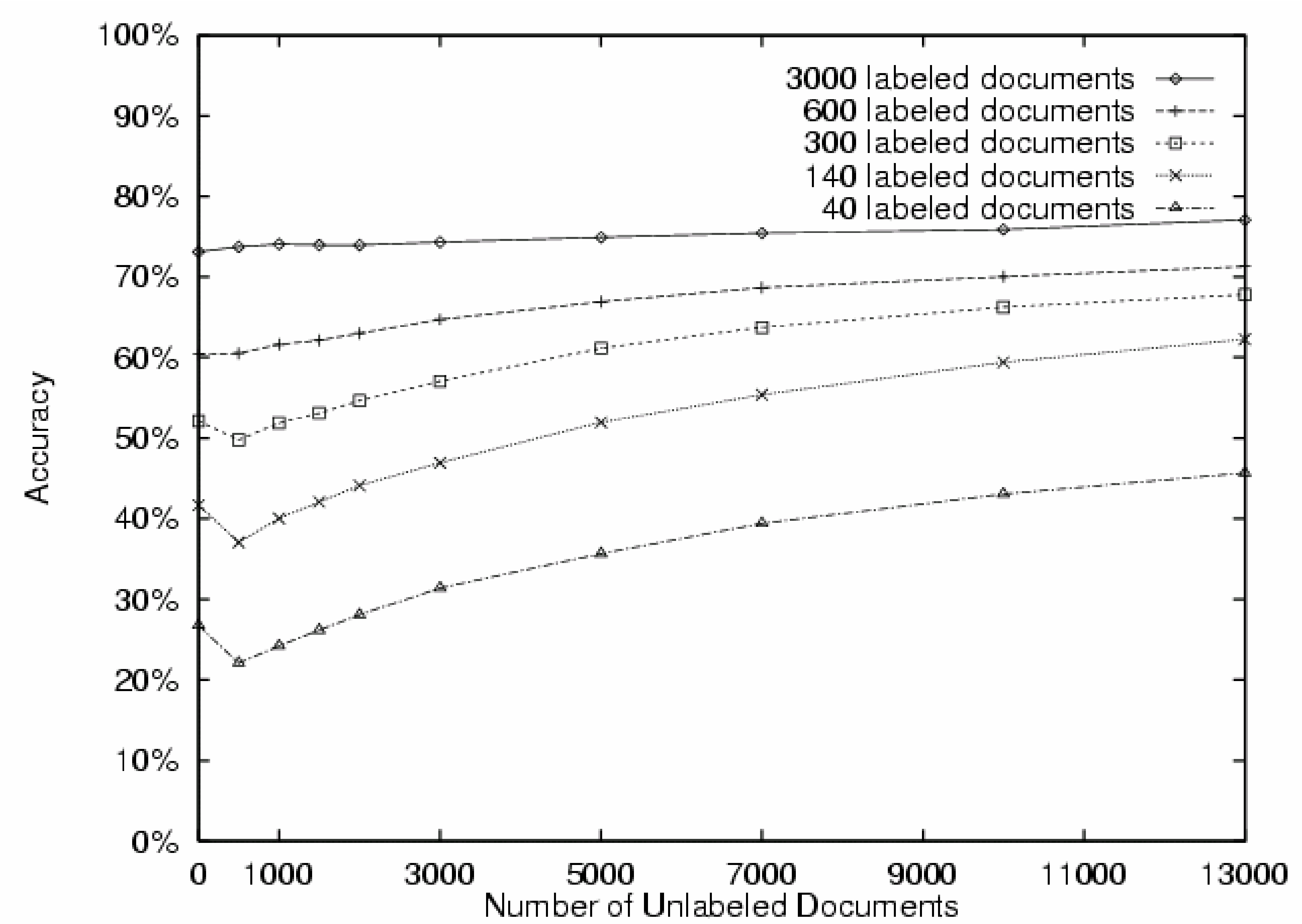
Experimental Evaluation

- Newsgroup postings
  - 20 newsgroups, 1000/group
- Web page classification
  - student, faculty, course, project
  - 4199 web pages
- Reuters newswire articles
  - 12,902 articles
  - 90 topics categories

20 Newsgroups



20 Newsgroups



What you should know about EM

- For learning from partly unobserved data
- MLEst of $\theta = \arg\max_{\theta} \log P(data|\theta)$
- 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

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

- EM for training Bayes networks
- Can also develop MAP version of EM
- Be able to derive your own EM algorithm for your own problem

Combining Labeled and Unlabeled Data

How else can unlabeled data be useful for supervised learning/function approximation?

Combining Labeled and Unlabeled Data

How can unlabeled data {x} be useful for learning f: X→Y

1. Using EM, if we know the form of P(Y|X)

2. By letting us estimate P(X) and reweight labeled examples

3. Co-Training [Blum & Mitchell, 1998]

4. To detect overfitting [Schuurmans, 2002]