Expectation Maximization, and Learning from Partly Unobserved Data (part 2)

Machine Learning 10-701
April 2005

Tom M. Mitchell
Carnegie Mellon University

Outline

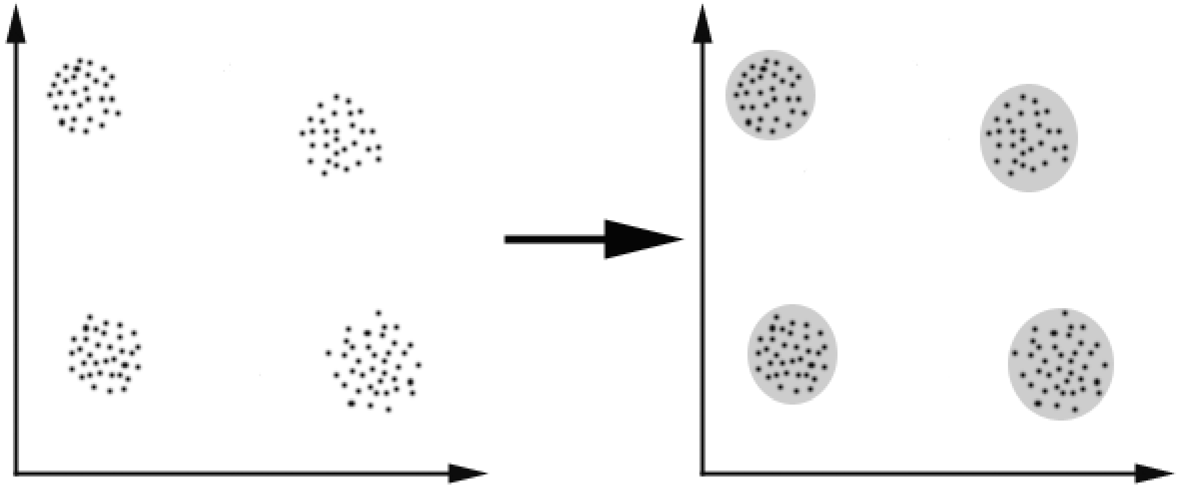
- Clustering
  - K means
  - EM: Mixture of Gaussians

- Training classifiers with partially unlabeled data
  - Naïve Bayes and EM
  - Reweighting the labeled examples using P(X)
  - Co-training
  - Regularization based on

1. Unsupervised 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, …)



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 $\mu_1 \cdots \mu_K$ 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 applet
- 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 distribution:

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

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

EM for Mixture of Gaussians

Simplify to make this easier

1. assume X_i are conditionally independent given Z.

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

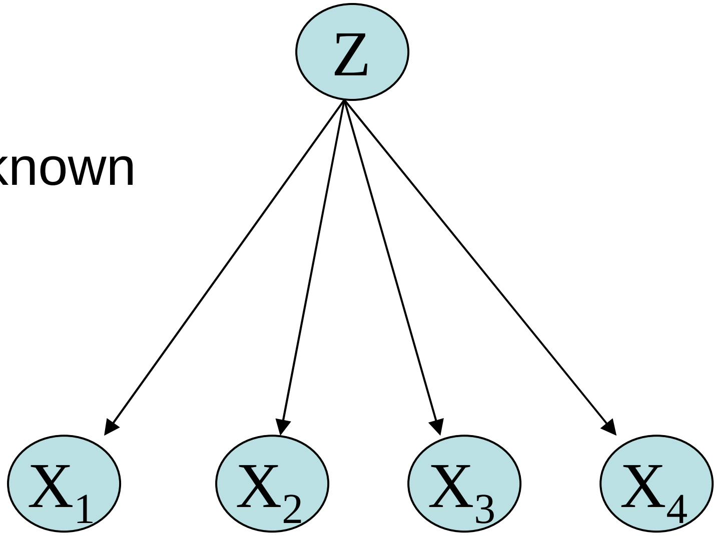
2. assume only 2 classes, and assume $\forall i, j, \sigma_{ji} = \sigma$

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

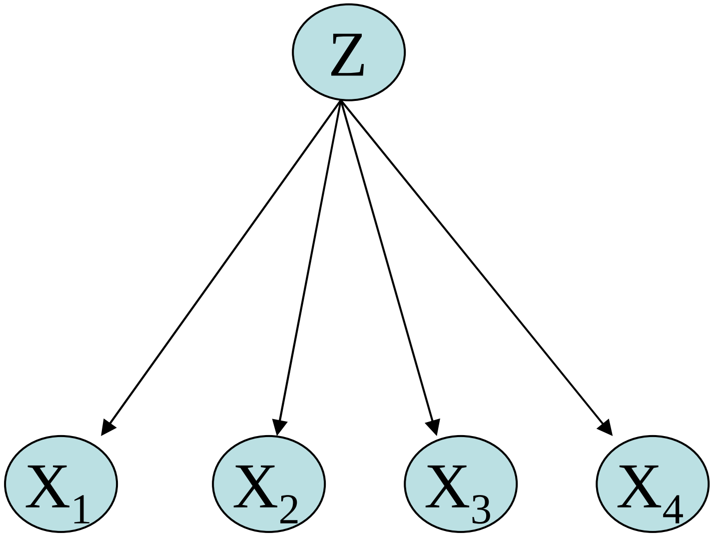
3. Assume σ known, $\pi_1 \ldots \pi_K, \mu_{1i} \ldots \mu_{Ki}$ unknown

Observed: X

Unobserved: Z



EM



Given observed variables X, unobserved Z

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

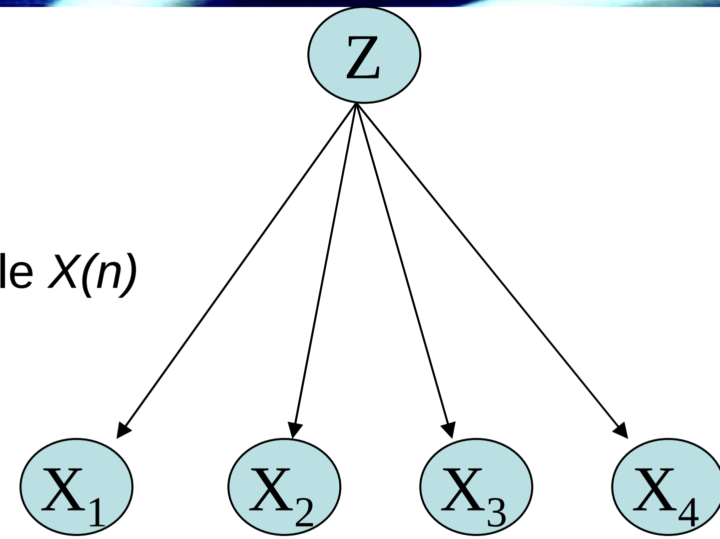
where $\theta = \langle \pi, \mu_{ji} \rangle$

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 $X(n)$

$X(n) = \langle x_1(n), x_2(n), \ldots x_T(n) \rangle$.

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

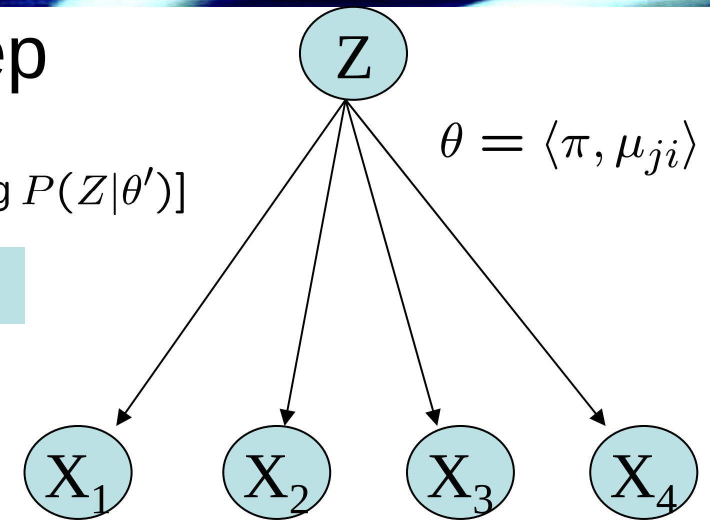
$$P(z(n)=k|x(n),\theta) = \frac{[\prod_i P(x_i(n)|z(n)=k,\theta)] \quad P(z(n)=k|\theta)}{\sum_{j=0}^{1} \prod_i P(x_i(n)|z(n)=j,\theta) \;\; P(z(n)=j|\theta)}$$

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

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

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

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

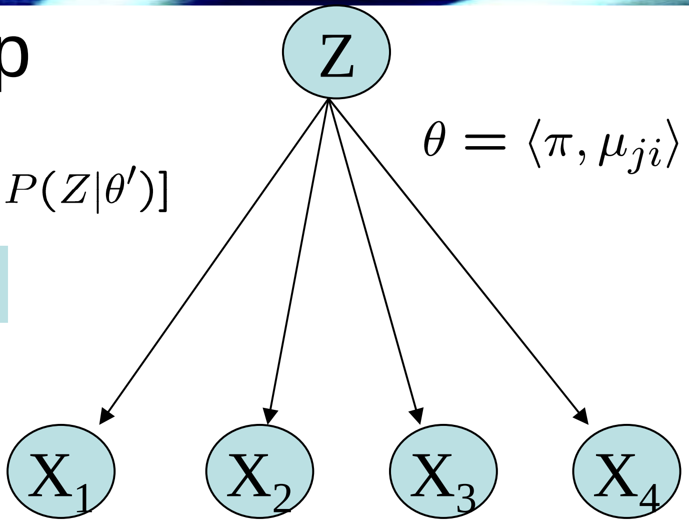
$$= \left(\sum_n E_{Z|X,\theta}[z(n)]\right)\log \pi' + \left(\sum_n E_{Z|X,\theta}[(1-z(n))]\right)\log(1-\pi')$$

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

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

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:

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

Mixture of Gaussians applet

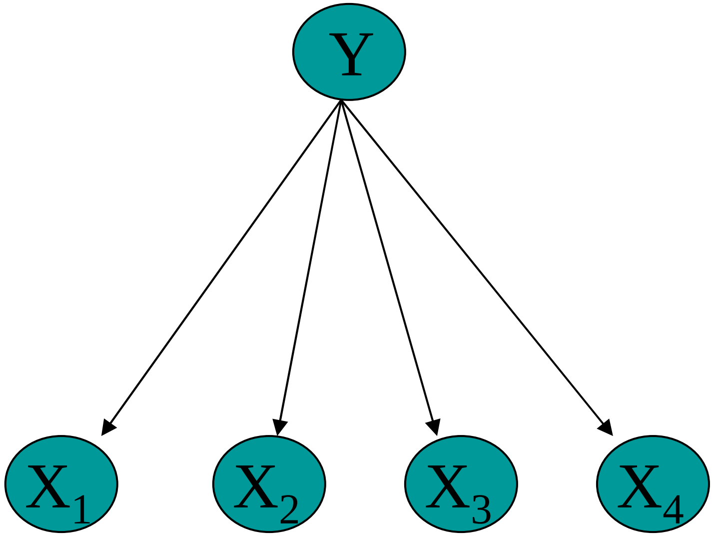
- Run applet
- Try 2 clusters
- See different local minima with different random starts

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 $P(X|\theta)$
- MixGaussians is the more general formulation
  - Equivalent to K Means when $\Sigma_k = \sigma I$, and $\sigma \to 0$

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.

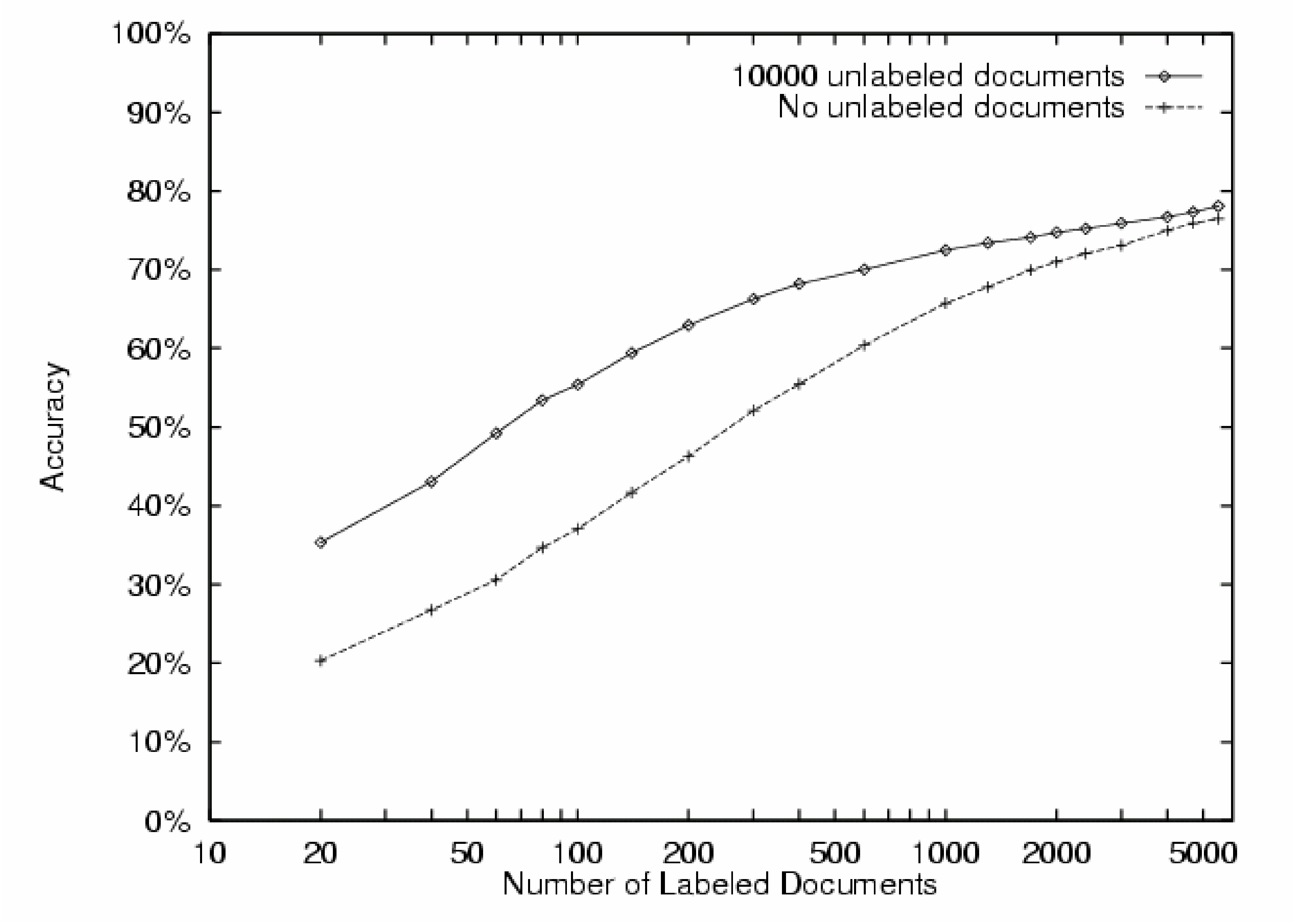
Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^{\star}$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^{\star}$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
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

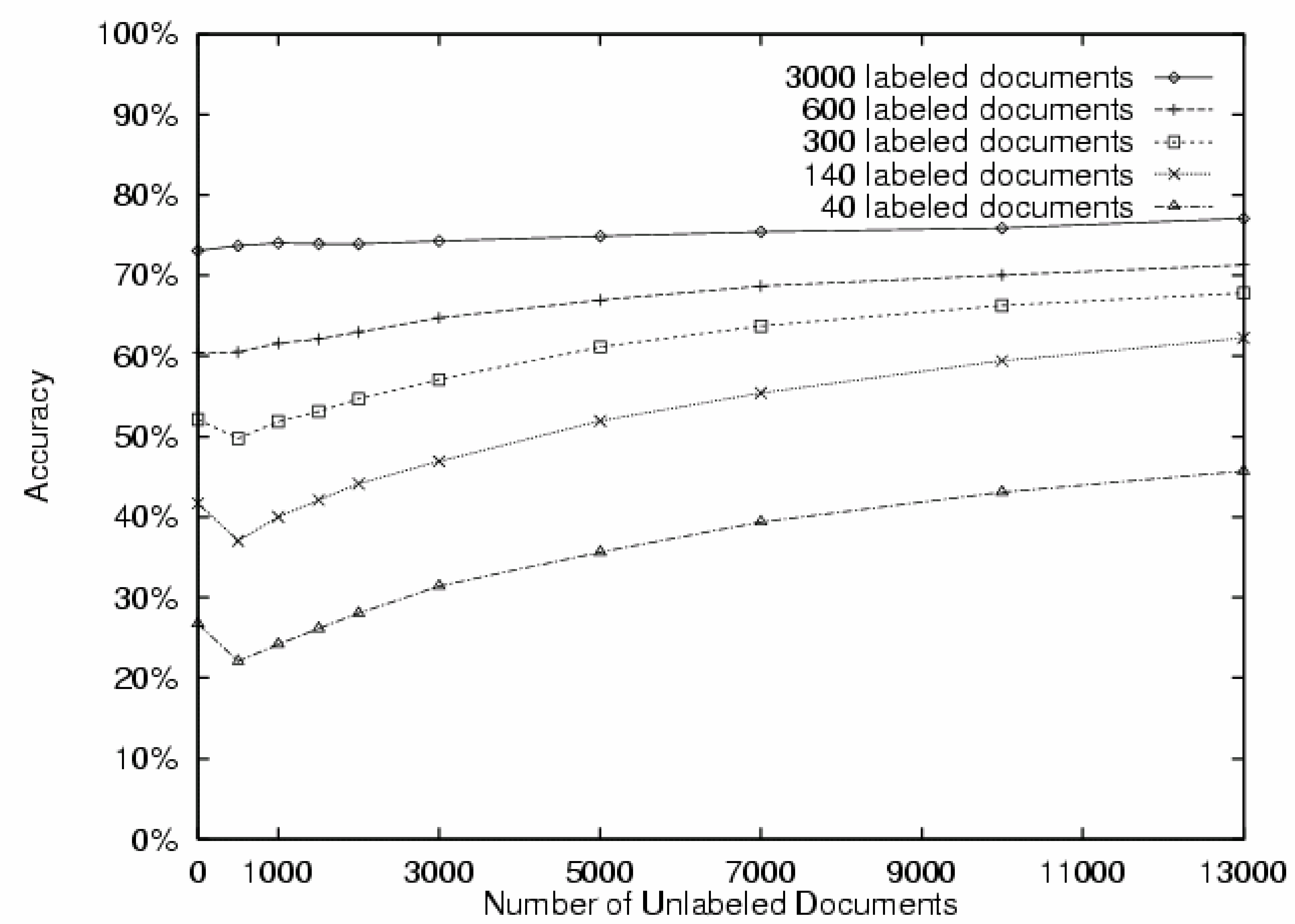
Experimental Evaluation

- 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



Combining Labeled and Unlabeled Data

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

1. Use U to reweight labeled examples

Can use $U \rightarrow \hat{P}(X)$ to alter optimization problem

- Wish to find

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) P(x)$$

1 if hypothesis *h* disagrees with true function *f*, else 0

- Often approximate as

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \frac{1}{|L|} \sum_{\langle x,y \rangle \in L} \delta(h(x) \neq y)$$

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x,L)}{|L|}$$

- Can use U for improved approximation:

$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x,L) + n(x,U)}{|L| + |U|}$$

3. If Problem Setting Provides Redundantly Sufficient Features, use CoTraining

- In some settings, available data features are redundant and we can train two classifiers using different features

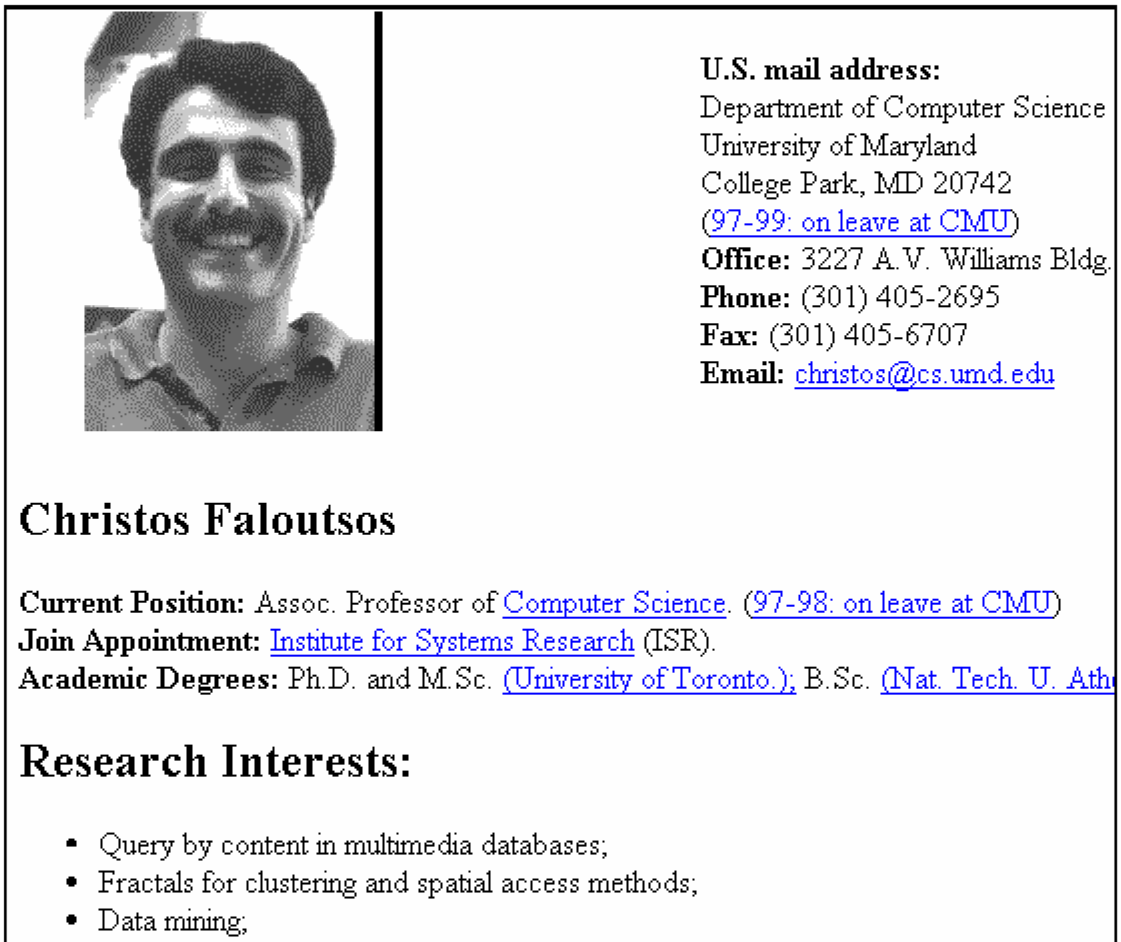
- In this case, the two classifiers should at least agree on the classification for each unlabeled example

- Therefore, we can use the unlabeled data to constrain training of both classifiers, forcing them to agree

Redundantly Sufficient Features

Professor Faloutsos

my advisor



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

Research Interests:

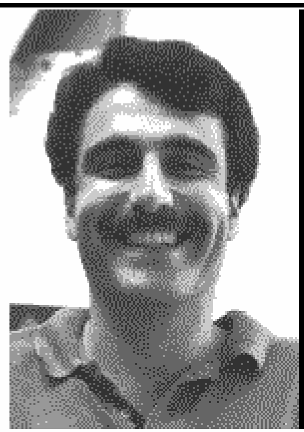
- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

Redundantly Sufficient Features

Professor Faloutsos

my advisor

Redundantly Sufficient Features



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

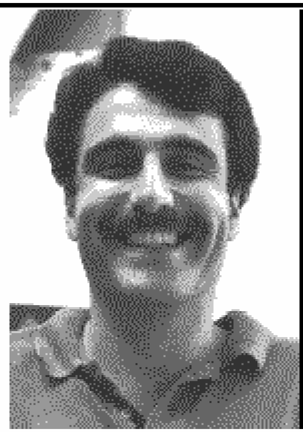
Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

Redundantly Sufficient Features

Professor Faloutsos

my advisor



U.S. mail address:
Department of Computer Science
University of Maryland
College Park, MD 20742
(97-99: on leave at CMU)
Office: 3227 A.V. Williams Bldg.
Phone: (301) 405-2695
Fax: (301) 405-6707
Email: christos@cs.umd.edu

Christos Faloutsos

Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.); B.Sc. (Nat. Tech. U. Ath

Research Interests:

- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

CoTraining Algorithm #1

[Blum&Mitchell, 1998]

Given: labeled data L,

unlabeled data U

Loop:

Train g1 (hyperlink classifier) using L

Train g2 (page classifier) using L

Allow g1 to label *p* positive, *n* negative examps from U

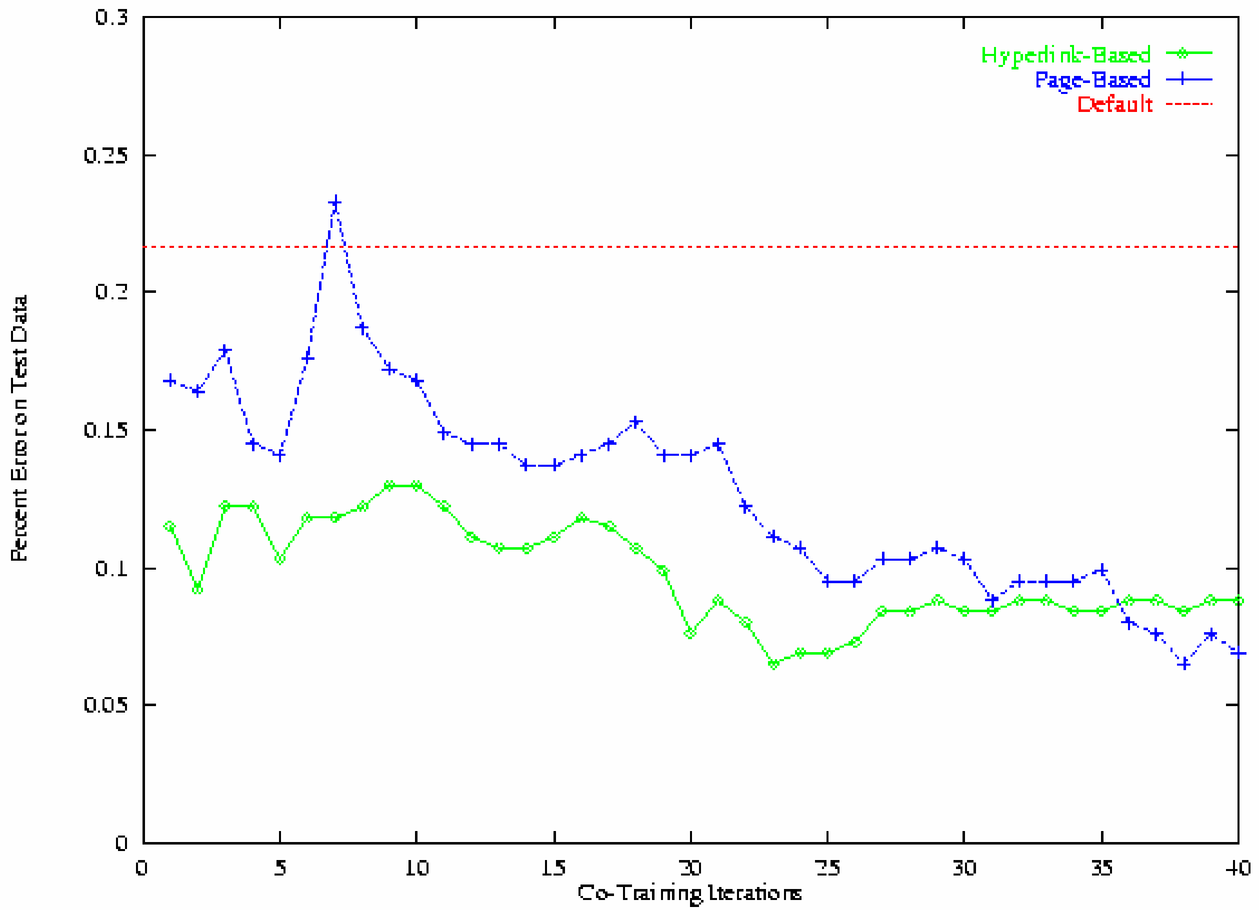
Allow g2 to label *p* positive, *n* negative examps from U

Add these self-labeled examples to L

CoTraining: Experimental Results

- begin with 12 labeled web pages (academic course)
- provide 1,000 additional unlabeled web pages
- average error: learning from labeled data 11.1%;
- average error: cotraining 5.0%

Typical run:

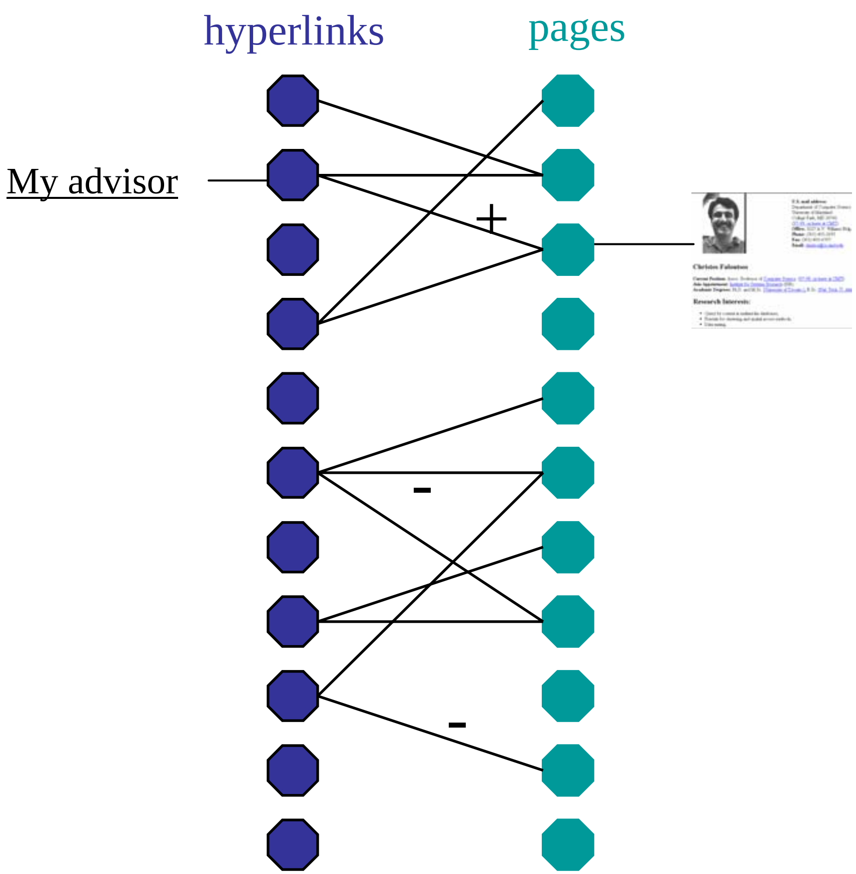


CoTraining Setting

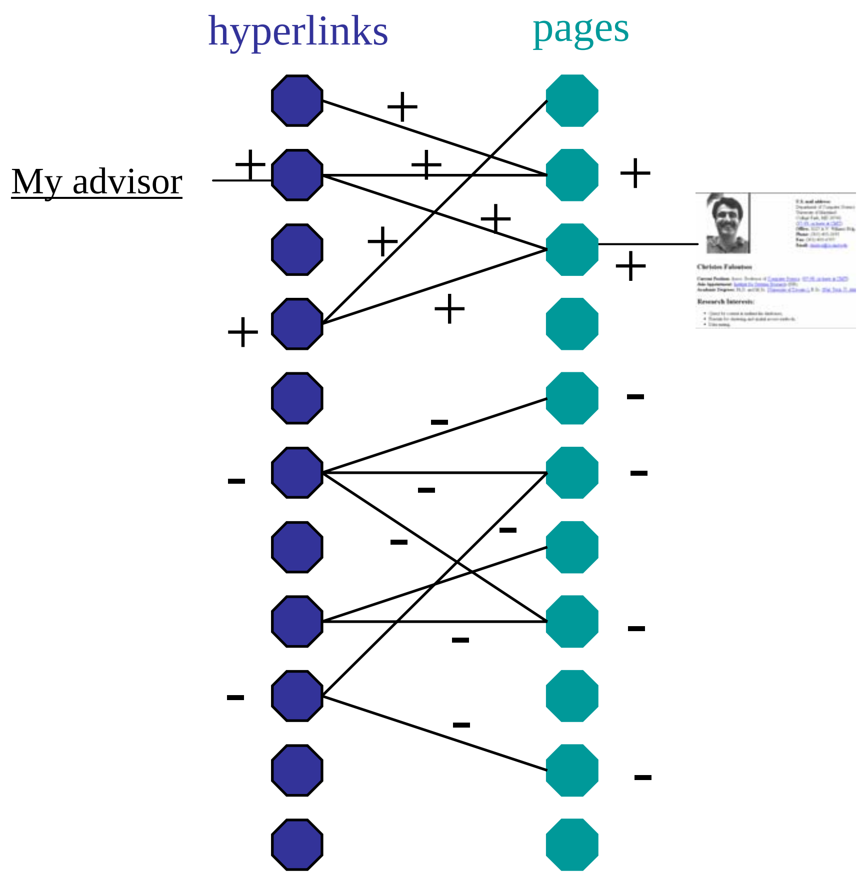
$$learn \quad f : X \rightarrow Y$$
$$where \quad X = X_1 \times X_2$$
$$where \quad x \quad drawn \quad from \quad unknown \quad distribution$$
$$and \quad \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x)$$

- If
  - x1, x2 conditionally independent given y
  - f is PAC learnable from noisy *labeled* data
- Then
  - f is PAC learnable from weak initial classifier plus *unlabeled* data

Co-Training Rote Learner



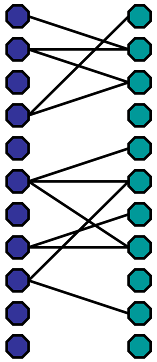
Co-Training Rote Learner



Expected Rote CoTraining error given *m* examples

$$CoTraining \quad setting:$$
$$learn \quad f : X \rightarrow Y$$
$$where \quad X = X_1 \times X_2$$
$$where \quad x \quad drawn \quad from \quad unknown \quad distribution$$
$$and \quad \exists g_1, g_2 \quad (\forall x) g_1(x_1) = g_2(x_2) = f(x)$$

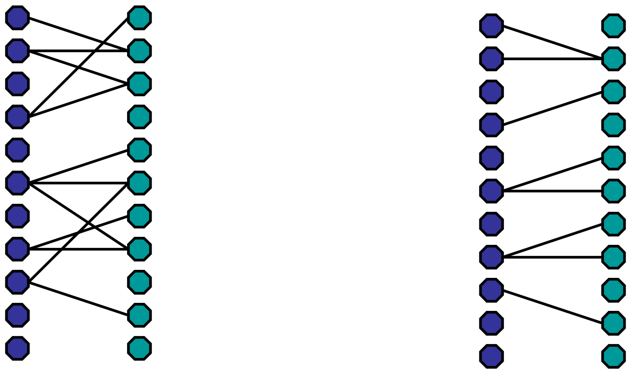
$$E[error] = \sum_j P(x \in g_j)(1 - P(x \in g_j))^m$$



Where g_j is the *j*th connected component of graph

How many *unlabeled* examples suffice?

Want to assure that connected components in the underlying distribution, G_D, are connected components in the observed sample, G_S



G_D G_S

$O(\log(N)/\alpha)$ examples assure that with high probability, G_S has same connected components as G_D [Karger, 94]

N is size of G_D, α is min cut over all connected components of G_D

PAC Generalization Bounds on CoTraining

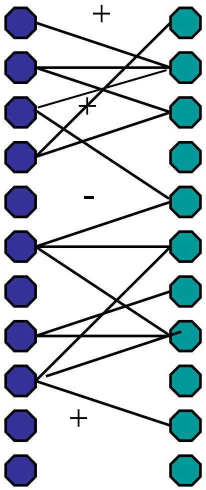
[Dasgupta et al., NIPS 2001]

Theorem 1 *With probability at least $1-\delta$ over the choice of the sample S, we have that for all h_1 and h_2, if $\gamma_i(h_1, h_2, \delta) > 0$ for $1 \leq i \leq k$ then (a) f is a permutation and (b) for all $1 \leq i \leq k$,*

$$P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp) \leq \frac{\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp) + \epsilon_i(h_1, h_2, \delta)}{\gamma_i(h_1, h_2, \delta)}.$$

The theorem states, in essence, that if the sample size is large, and h_1 and h_2 largely agree on the unlabeled data, then $\widehat{P}(h_1 \neq i \mid h_2 = i, h_1 \neq \perp)$ is a good estimate of the error rate $P(h_1 \neq i \mid f(y) = i, h_1 \neq \perp)$.

What if CoTraining Assumption Not Perfectly Satisfied?



- Idea: Want classifiers that produce a *maximally consistent* labeling of the data
- If learning is an optimization problem, what function should we optimize?

What Objective Function?

$$E = E1 + E2 + c_3 E3 + c_4 E4$$

Error on labeled examples

$$E1 = \sum_{<x,y>\in L} (y - \hat{g}_1(x_1))^2$$

Disagreement over unlabeled

$$E2 = \sum_{<x,y>\in L} (y - \hat{g}_2(x_2))^2$$

$$E3 = \sum_{x\in U} (\hat{g}_1(x_1) - \hat{g}_2(x_2))^2$$

Misfit to estimated class priors

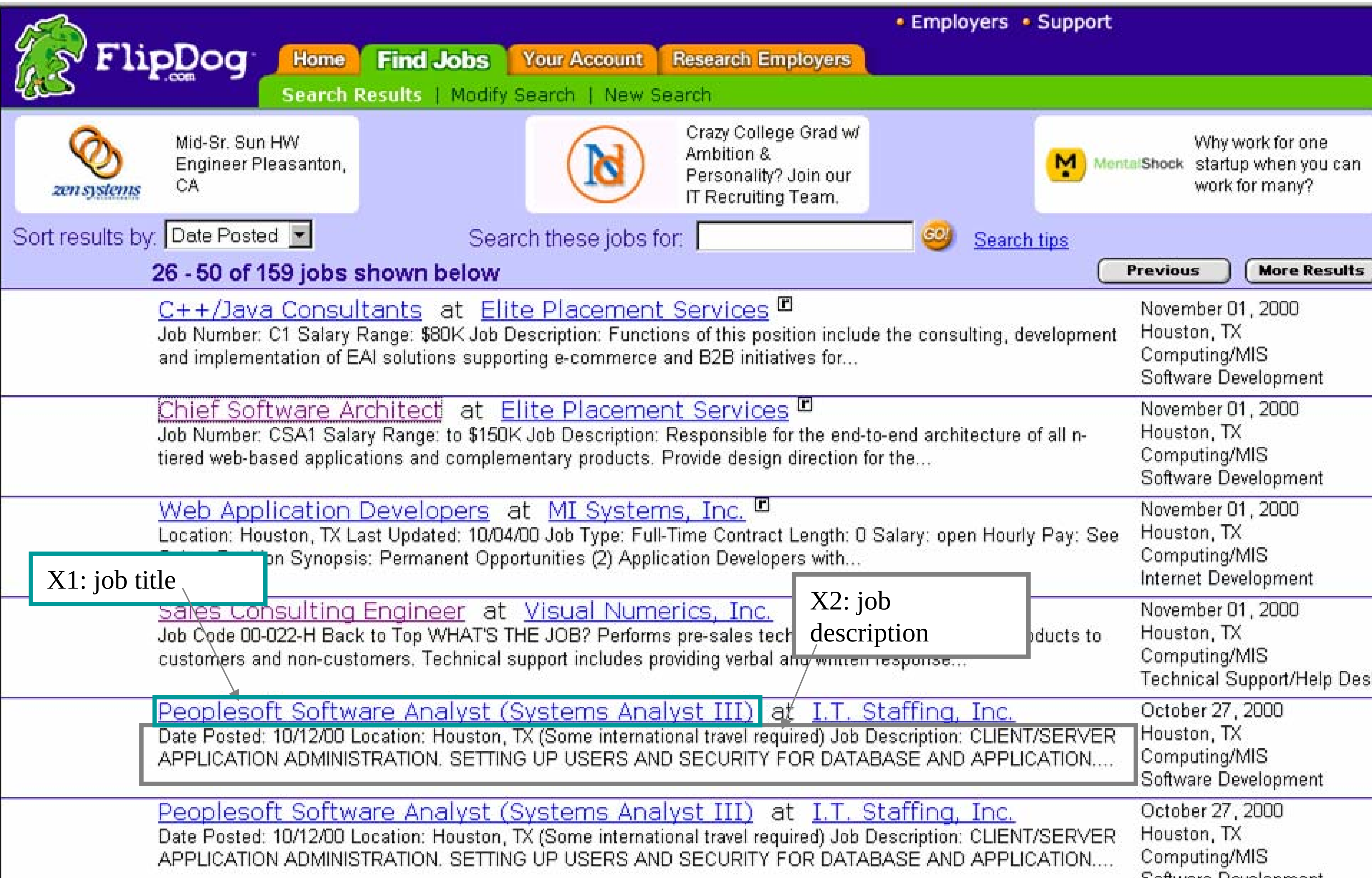
$$E4 = \left(\left(\frac{1}{|L|} \sum_{<x,y>\in L} y \right) - \left(\frac{1}{|L|+|U|} \sum_{x\in L\cup U} \frac{\hat{g}_1(x_1) + \hat{g}_2(x_2)}{2} \right) \right)^2$$

What Function Approximators?

$$\hat{g}_1(x) = \frac{1}{1 + e^{\sum_j w_{j,1} x_j}} \qquad \hat{g}_2(x) = \frac{1}{1 + e^{\sum_j w_{j,2} x_j}}$$

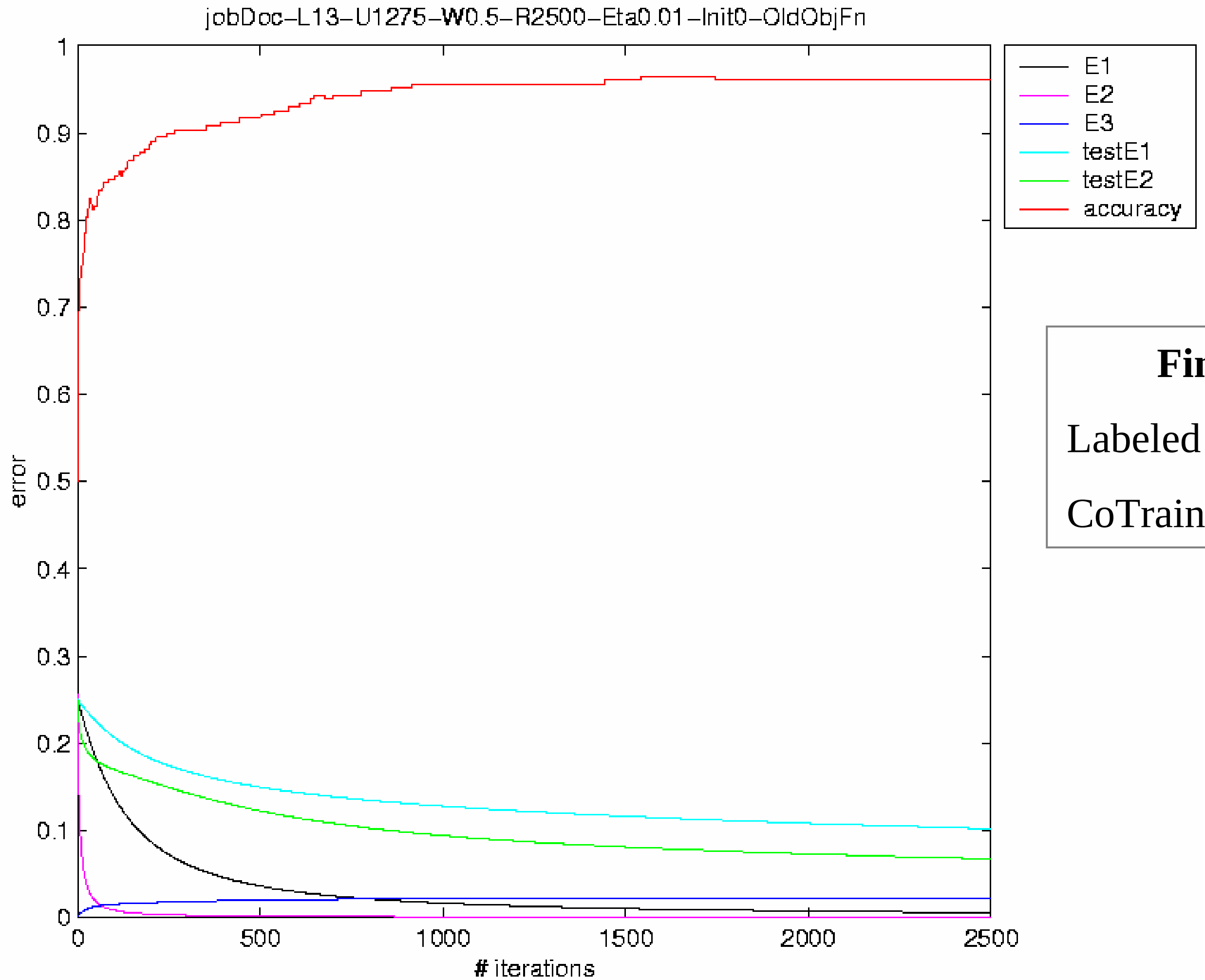
- Same fn form as Naïve Bayes, Max Entropy
- Use gradient descent to simultaneously learn g1 and g2, directly minimizing E = E1 + E2 + E3 + E4
- No word independence assumption, use both labeled and unlabeled data

Classifying Jobs for FlipDog



Gradient CoTraining

Classifying FlipDog job descriptions: SysAdmin vs. WebProgrammer



Final Accuracy

Labeled data alone: 86%

CoTraining: 96%

Gradient CoTraining

Classifying Upper Case sequences as Person Names

Error Rates

	25 labeled 5000 unlabeled	*2300 labeled 5000 unlabeled*
Using labeled data only	.24	.13
Cotraining	.15 *	.11 *
Cotraining without fitting class priors (E4)	.27 *	

* sensitive to weights of error terms E3 and E4

CoTraining Summary

- Unlabeled data improves supervised learning when example features are redundantly sufficient
  - Family of algorithms that train multiple classifiers

- Theoretical results
  - Expected error for rote learning
  - If X1,X2 conditionally independent given Y, Then
    - PAC learnable from weak initial classifier plus unlabeled data
    - error bounds in terms of disagreement between g1(x1) and g2(x2)

- Many real-world problems of this type
  - Semantic lexicon generation [Riloff, Jones 99], [Collins, Singer 99]
  - Web page classification [Blum, Mitchell 98]
  - Word sense disambiguation [Yarowsky 95]
  - Speech recognition [de Sa, Ballard 98]

What you should know

- Clustering:
  - K-means algorithm : hard labels
  - EM for mixtures of Gaussians : probabilistic labels
- Be able to derive your own EM algorithm

- Using unlabeled data to help with supervised classification
  - Naïve Bayes augmented by unlabeled data
  - Using unlabeled data to reweight labeled examples
  - Co-training
  - Using unlabeled data for regularization