

# Semi-supervised Learning

# Why Semi-Supervised Learning

- Unsupervised and Supervised learning
  - Two extreme learning paradigms
  - Unsupervised learning
    - e.g., collection of documents without any labels
    - easy to collect
  - Supervised learning
    - each object labeled with a class.
    - Expensive to do
- Real life applications are often somewhere in between – Semi-supervised Learning

# Semi-Supervised Problem Setup

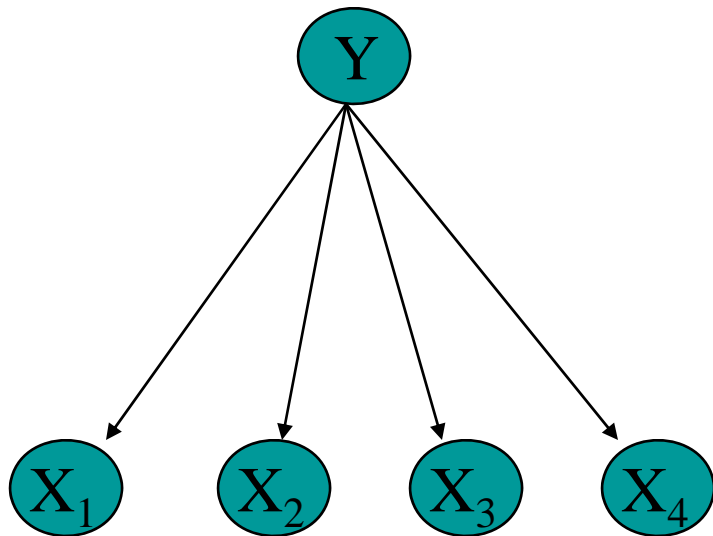
- Goal: predict  $Y$  from features  $X$
- Have some labeled training data  $L$ 
  - I.e.  $X$ 's paired with corresponding  $Y$ 's
  - E.g. faces classified by male or female
- Lots of unlabeled data  $U$ 
  - I.e. just a set of  $X$ 's
  - E.g. a database of unlabeled faces
- Can we somehow use the unlabeled data to arrive at a more accurate classifier than if we just trained a classifier using  $L$ ?
  - If so then we can significantly reduce labeling effort

# Semi-supervised Naïve Bayes on EM

(Nigam et al., ML2000)

Naïve Bayes Model

Learn  $P(Y|X)$



	Y	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>
L	1	0	0	1	1
	0	0	1	0	0
	0	0	1	1	0
U	?	0	0	0	1
	?	0	1	0	1

labeled and **unlabeled** training data

# Semi-Supervised EM

- We can view the missing labels in  $U$  as variables whose value are hidden and apply EM
- Use initial labeled data  $L$  to get initial parameter estimates.
- In each iteration
  - **E-step:** use current model parameters to estimate distribution over hidden class labels
    - Note that labels of data in  $L$  do not get altered by this step
  - **M-step:** use all data (labeled and unlabeled) to re-estimate the parameters
- Repeat until converge.

Example for text document classification. The instances are documents  $d_i$  and NB model is viewed as generating document by drawing words independently conditioned on the class

E-Step: (do this only for unlabeled data)

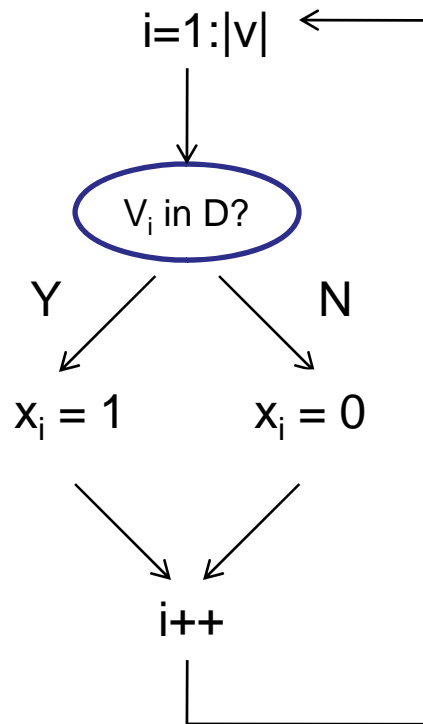
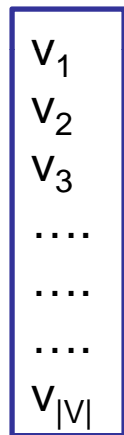
$$\begin{aligned} P(y_i = c_j | d_i; \hat{\theta}) &= \frac{P(c_j | \hat{\theta}) P(d_i | c_j; \hat{\theta})}{P(d_i | \hat{\theta})} \\ &= \frac{P(c_j | \hat{\theta}) \prod_{k=1}^{|d_i|} P(w_{d_i, k} | c_j; \hat{\theta})}{\sum_{r=1}^{|\mathcal{C}|} P(c_r | \hat{\theta}) \prod_{k=1}^{|d_i|} P(w_{d_i, k} | c_r; \hat{\theta})}. \end{aligned}$$

M-Step: maximum likelihood estimate of the parameters as if the expected values of  $y$  computed in E step is the true value of the missing data

# Side track: Naïve bayes multi-nomial model

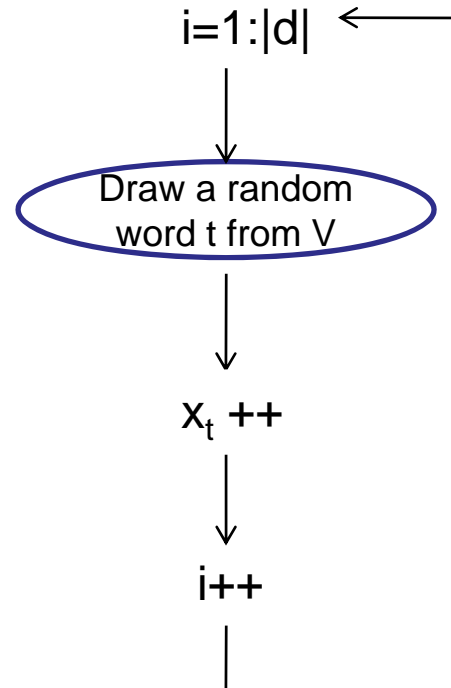
It is common to use a multi-nomial model for word distribution instead of previously seen binomial distribution. Let's see briefly what is the difference

Binomial case:



$P(x_i=1|y=1), i=1, \dots, |V|,$   
 $P(x_i=1|y=0), i=1, \dots, |V|$

Multinomial case: initialize  $\mathbf{x} = \mathbf{0}$



$P(x_i=1|y=1), i=1, \dots, |V|-1,$   
 $P(x_i=1|y=0), i=1, \dots, |V|-1$

M-step:

$w_t$  is t-th word in vocabulary

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

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

Laplace  
smoothing

# Augmented EM: Weight unlabeled examples

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

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

Chosen by cross validation

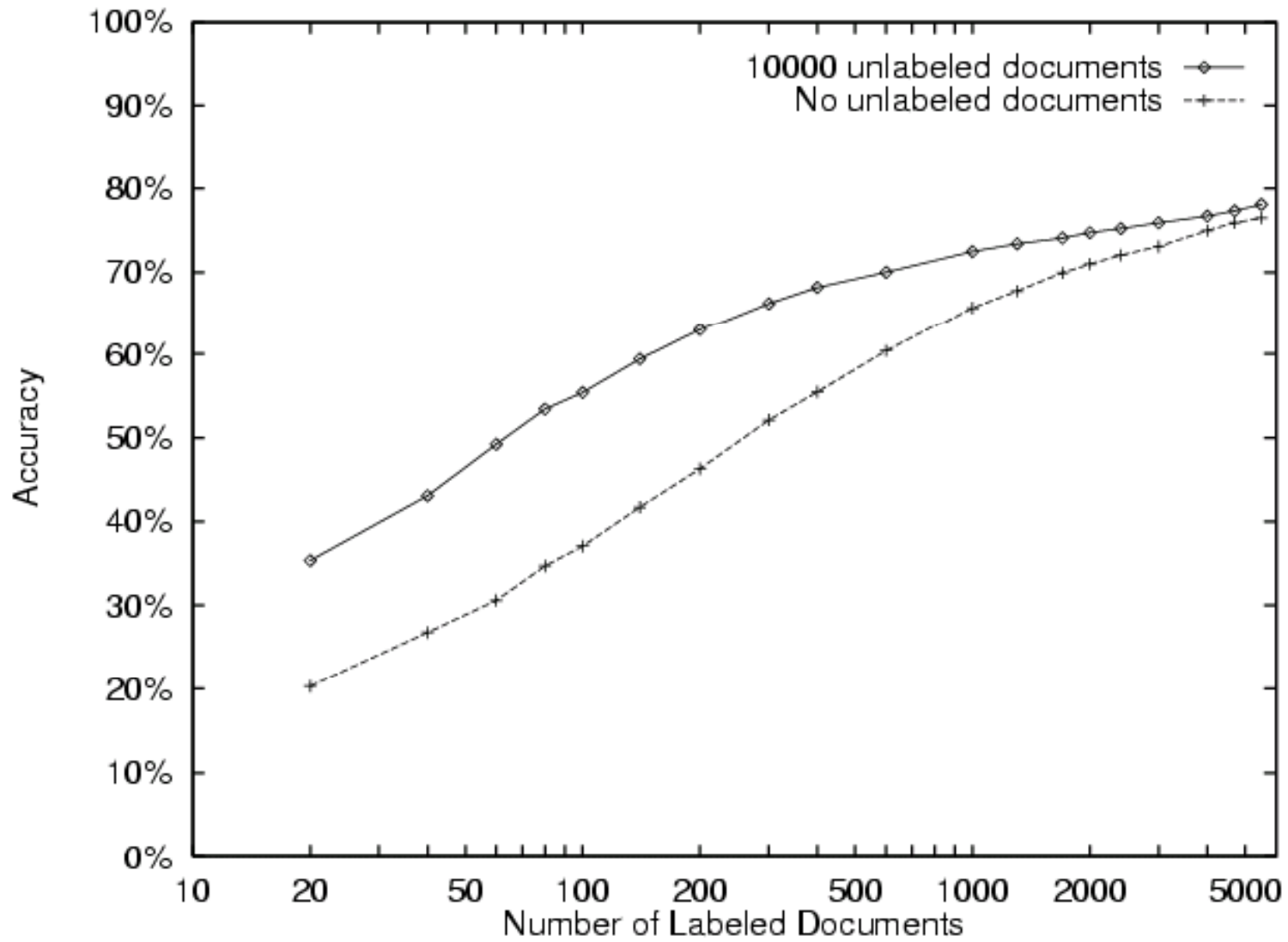
New M step:

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

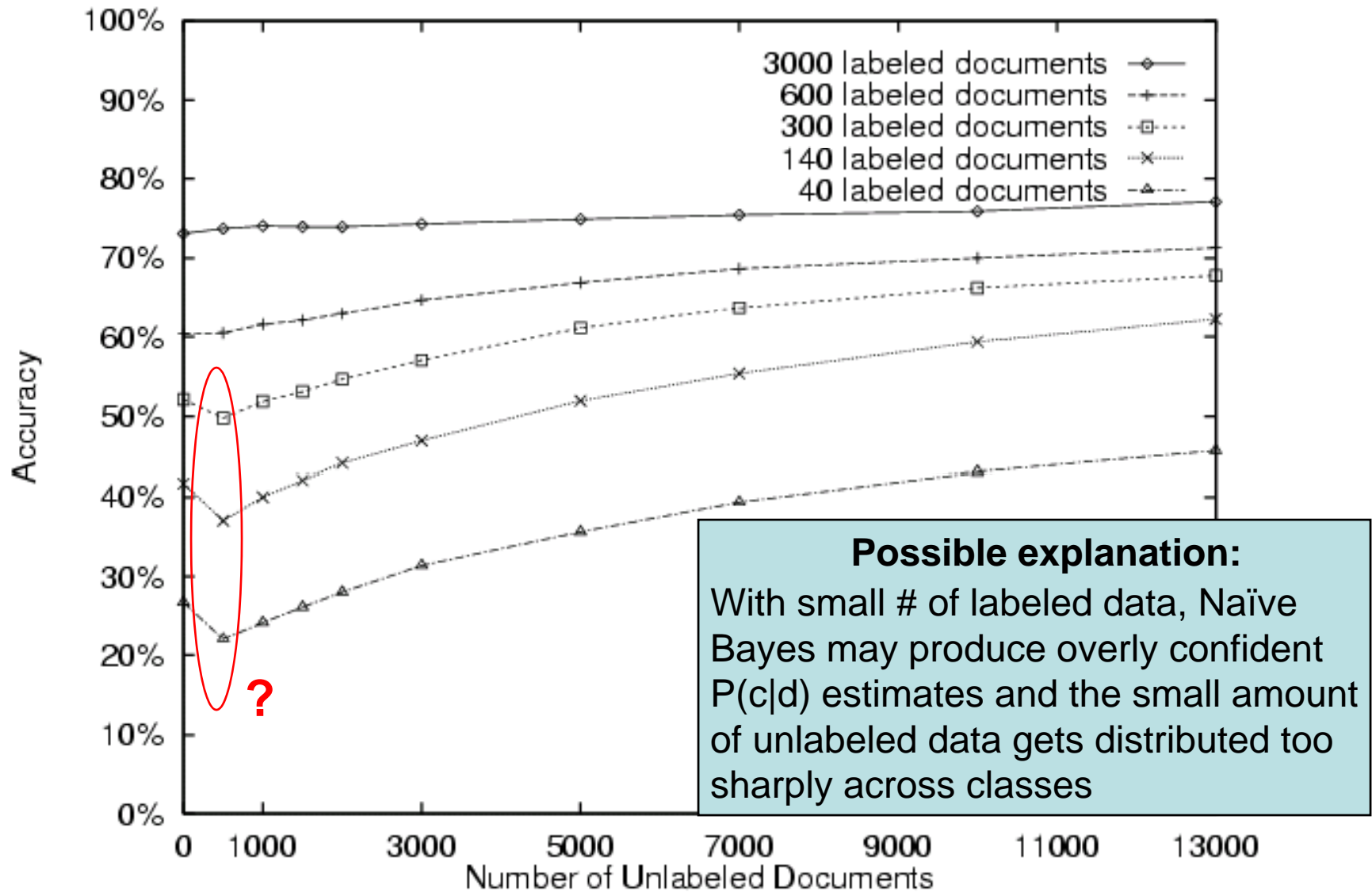
$$\hat{\theta}_{c_j} \equiv P(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|\mathcal{D}|} \Lambda(i) P(y_i = c_j | d_i)}{|\mathcal{C}| + |\mathcal{D}^l| + \lambda |\mathcal{D}^u|}$$

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

# 20 News Groups



# 20 News Groups



# Comments

- EM helps naïve bayes find more accurate classifier by optimizing the posterior model probability, not classification accuracy directly
- If the generative model is perfect (i.e, matches the true underlying distribution perfect, then model fit and classification accuracy are expected to be strongly correlated and EM is expected to help.
- Semi-supervised learning using a generative model with EM leans more heavily on the correctness of the generative model


# Co-training for Semi-Supervised Learning

(Blum and Mitchell 1998)

- Assumes feature  $X$  is very expressive and has redundant information
- Exploits redundant information for semi-supervised learning
- Redundant info:
  - Text in the document
  - Anchor text for hyperlinks

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@cc.umd.edu](mailto:christos@cc.umd.edu)

**Christos Faloutsos**

**Current Position:** Assoc. Professor of [Computer Science](#). [\(97-98: on leave at CMU\)](#)  
**Joint Appointment:** [Institute for Systems Research](#) (ISR).  
**Academic Degrees:** Ph.D. and M.Sc. ([University of Toronto](#)), B.Sc. ([Nat. Tech. U. Athens](#))

**Research Interests:**

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

# Basic Setup

- Goal: predict  $Y$  from features  $X$
- Have some labeled data  $L$
- Lots of unlabeled data  $U$
- $X$  is very expressive and has multiple parts that contain redundant information
  - $X = \{X_1, X_2\} = \{\textit{hyperlink text}, \textit{page text}\}$
  - We can learn
    - $g(X_1) \mapsto Y$
    - $g(X_2) \mapsto Y$

# Co-training Algorithm

Given:

- a set  $L$  of labeled training examples
- a set  $U$  of unlabeled examples

Create a pool  $U'$  of examples by choosing  $u$  examples at random from  $U$

Loop for  $k$  iterations:

Use  $L$  to train a classifier  $h_1$  that considers only the  $x_1$  portion of  $x$

Use  $L$  to train a classifier  $h_2$  that considers only the  $x_2$  portion of  $x$

Allow  $h_1$  to label  $p$  positive and  $n$  negative examples from  $U'$

Allow  $h_2$  to label  $p$  positive and  $n$  negative examples from  $U'$

Add these self-labeled examples to  $L$

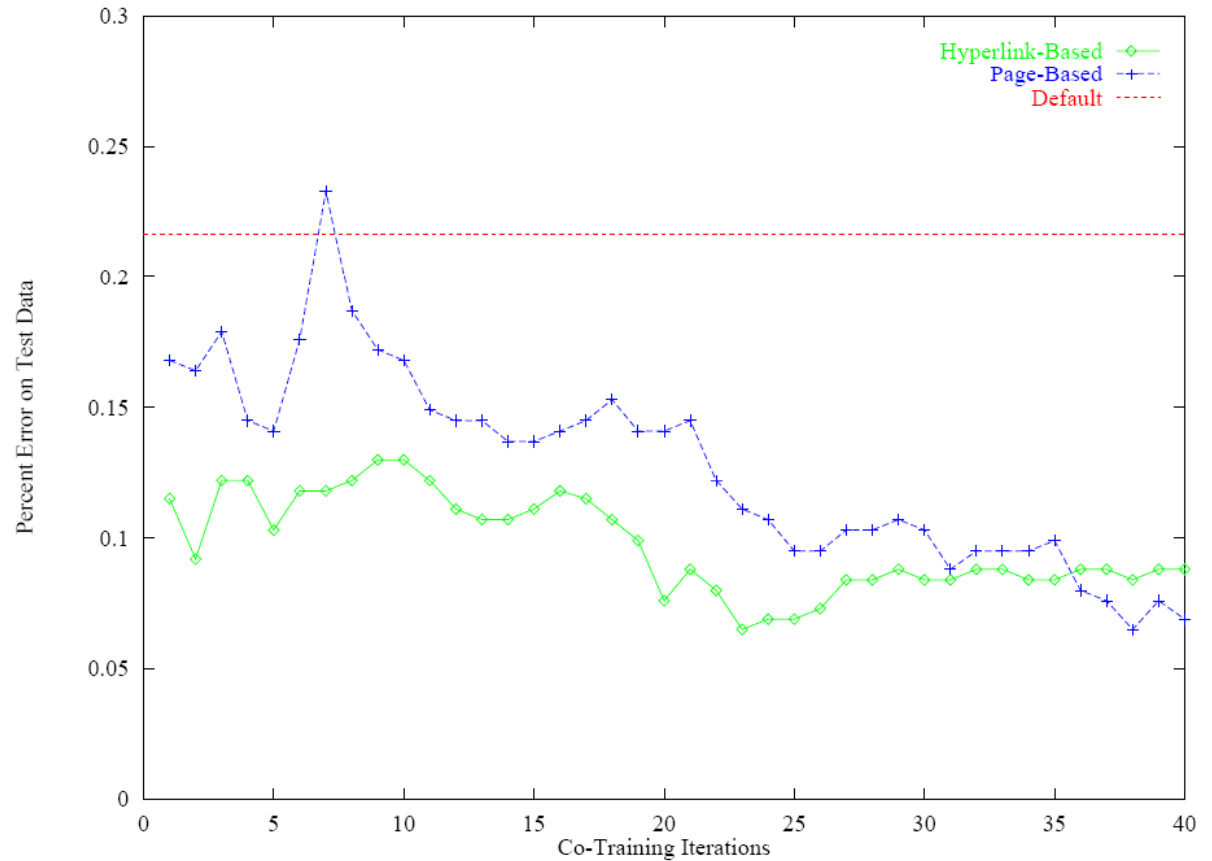
Randomly choose  $2p + 2n$  examples from  $U$  to replenish  $U'$

**The classifiers label the  $p$  and  $n$  examples that they are most confident about**

**If they label different instances then they can effectively bootstrap each other**

# Experimental Results

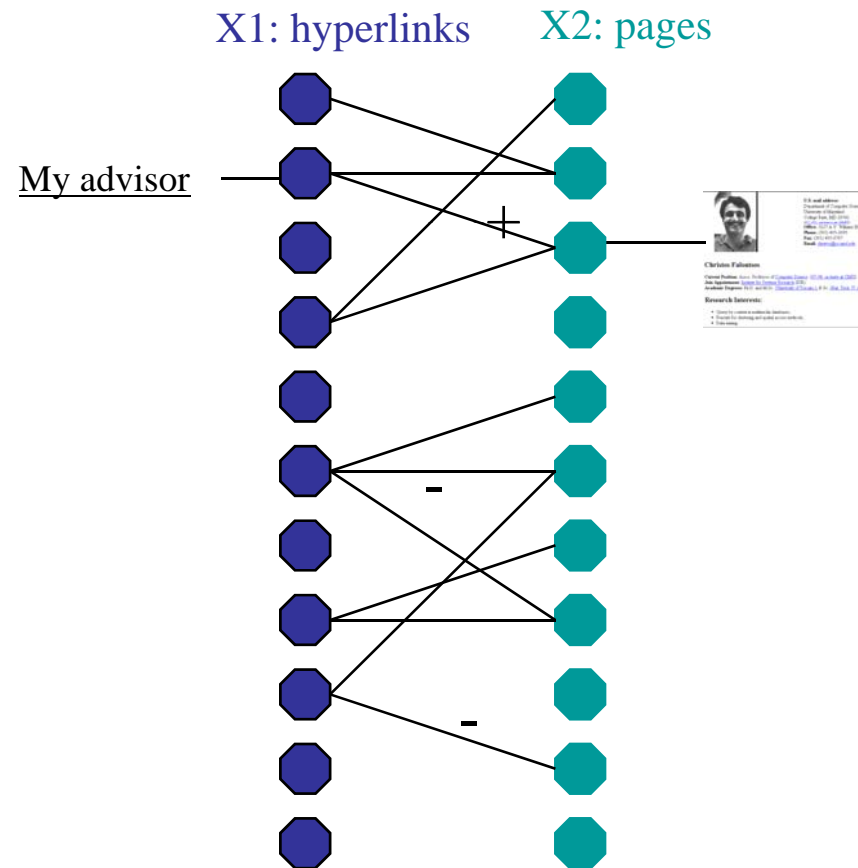
- 12 labeled web pages
- 1,000 additional unlabeled web pages
- Learning algorithm:  
Naïve Bayes
- Average error:
  - learning from labeled data only using combined classifier: ~11%
  - Co-training: ~5%



# Co-training Theory

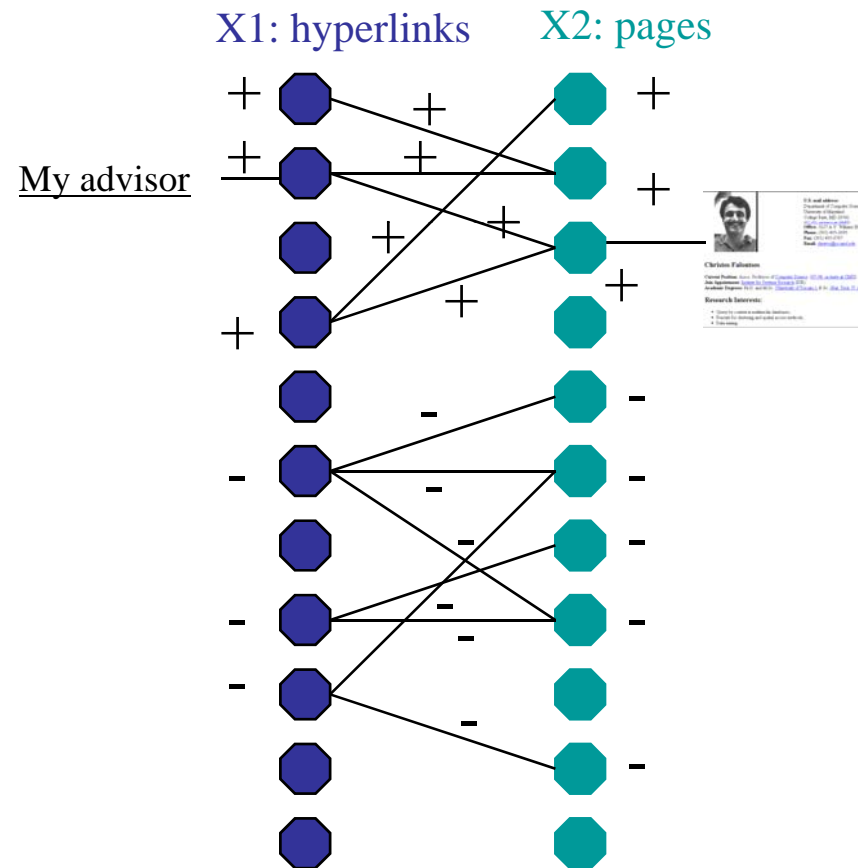
- Want to predict  $Y$  from features  $\mathbf{X}$ 
  - $f(\mathbf{X}) \mapsto Y$
- Co-training assumption:  $\mathbf{X}$  is very expressive
  - $\mathbf{X}=(\mathbf{X}_1, \mathbf{X}_2)$  and  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are *conditionally independent*
  - Want to learn  $g_1(X_1) \mapsto Y$  and  $g_2(X_2) \mapsto Y$
- Assumption:  
 $\exists g_1, g_2, \forall \mathbf{X} g_1(X_1) = f(\mathbf{X}), g_2(X_2)=f(\mathbf{X})$   
Each set of features is sufficient for classification
- Some intuition:
  - The learners for  $X_1$  and  $X_2$  must generally agree on the unlabeled data. This constrains the space of hypotheses they may output
  - Reduces effective VC-dimension

# Understanding Co-Training: A Simple Setting



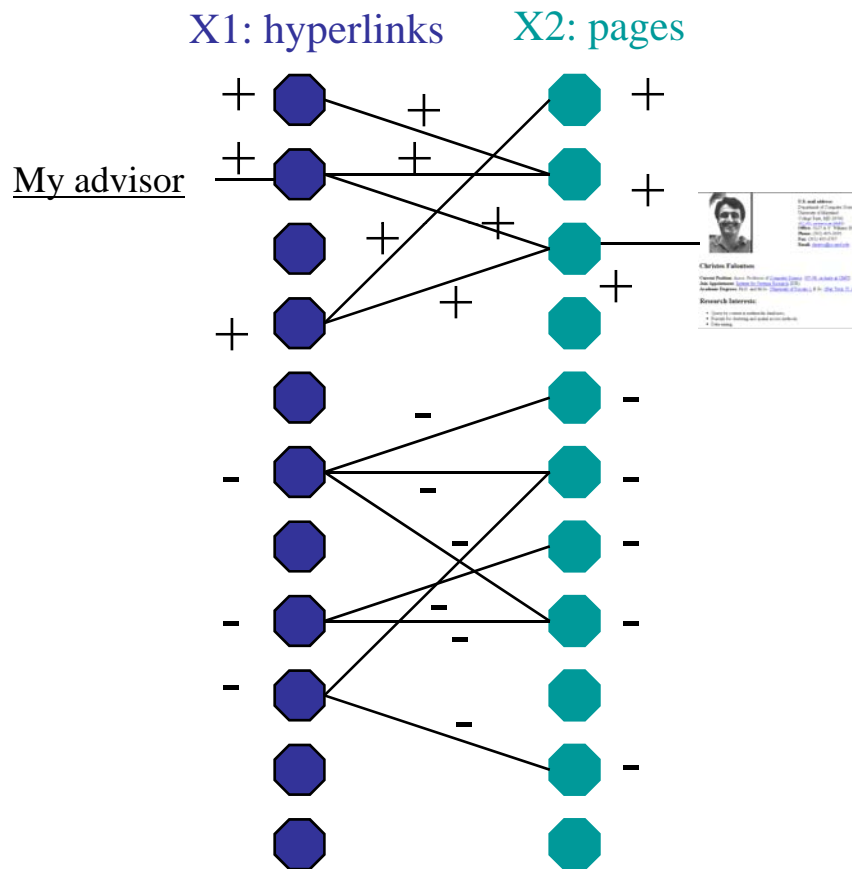
Edges represent examples (i.e. X1,X2 pairs). Some examples are labeled.

# Understanding Co-Training: A Simple Setting



Edges represent examples (i.e. X1,X2 pairs). Some examples are labeled.

# Understanding Co-Training: A Simple Setting



- Unlabeled data defines the connected components
  - Each component can have only one label
- Suppose we have infinite unlabeled data, we obtain the correct bipartite graph
- Labeled data provides labels to the connected components
- Each component will only need one labeled data
- Co-training with unlabeled data reduces the number of labeled data points needed

# Co-Training Theoretical Result

(Blum and Mitchell COLT1998)

- If
  - $\mathbf{x}_1, \mathbf{x}_2$  are conditionally independent given  $y$
  - and  $f$  is PAC learnable from noisy *labeled* data
    - e.g., give me an  $\varepsilon$  and  $\delta$ , I give you  $h$  such that error  $\leq \varepsilon$  with prob. at least  $1 - \delta$
- Then
  - $f$  is PAC learnable from **weak initial classifier** plus *unlabeled* data
    - Basic idea:  $h_1(x_1)$  can be considered as the noisy label for  $x_2$  and vice versa

# Summary of Co-Training

- Unlabeled data improves supervised learning when example features have two “independent” views
  - Train a classifier on one view to provide labels for learning in another view
- Understanding Co-training
  - Unlabeled data reduces the number of required labeled examples
  - If  $\mathbf{X}_1 \perp \mathbf{X}_2 \mid Y$  and  $f$  is PAC learnable from noisy labeled data, then  $f$  is PAC learnable from weak initial classifier plus unlabeled data

# Co-training vs. Semi-Supervised EM

- Co-training assumes two sets of features that are conditionally independent from each other and **split** them while training
- Naïve bayes also assumes feature conditional independence but **doesn't split**
- Co-training **incrementally** uses the unlabeled data
- EM probabilistically labels all the data at each round and **iteratively** uses the unlabeled data.

# Comparing co-training with EM: Web-KB course database

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	788	-0-	3.3%
Co-training	12	776	5.4%
EM	12	776	4.3%
Naive Bayes	12	-0-	13.0%

- EM performs slightly better than co-training here
- Both are close to supervised method when trained on more labeled data.

# The News 2\*2 dataset

- A semi-artificial dataset
- Conditional independence assumption holds.

Algorithm	# Labeled	# Unlabeled	Error
Naive Bayes	1006	-0-	3.9%
Co-training	6	1000	3.7%
EM	6	1000	8.9%
Naive Bayes	6	-0-	34.0%

Co-training outperforms EM and the “oracle” result.

# Co-EM: EM with feature split

- Repeat until converge
  - Train a A-feature-set classifier using the labeled data and the unlabeled data with labels provided by classifier B
  - Use classifier A to probabilistically label all the unlabeled data
  - Train B-feature-set classifier using the labeled data and the unlabeled data with A's labels.
  - Use classifier B to probabilistically label all the unlabeled data

# Four SSL methods

Method	Uses Feature Split?	
	Yes	No
Incremental	co-training	self-training
Iterative	co-EM	EM

Method	Uses Feature Split?	
	Yes	No
Incremental	3.7%	5.8%
Iterative	3.3%	8.9%

Results on the News 2\*2 dataset

# Random feature split

Method	Uses Random Feature Split?	
	Yes	No
Incremental	5.5%	5.8%
Iterative	5.1%	8.9%

Co-training: 3.7% → 5.5%

Co-EM: 3.3% → 5.1%

- When the conditional independence assumption does not hold, but there is sufficient redundancy among the features, co-training may still work well by using random feature splitting.

# Comments

- Unlabeled data can significantly help improve classification accuracy
- Co-training assumes that there are two redundant and conditionally independent feature sets
  - In practice there is often no natural split of features
  - Random splits can help as well
- Combining generative probabilistic models and EM leads to natural use of unlabeled data
  - Unlabeled data don't always lead to performance gain
  - Depend on whether the generative model is violated by unlabeled data
- Labeled data can also be used to help identifying clusters
  - Semi-supervised clustering