EM for Bayes Nets

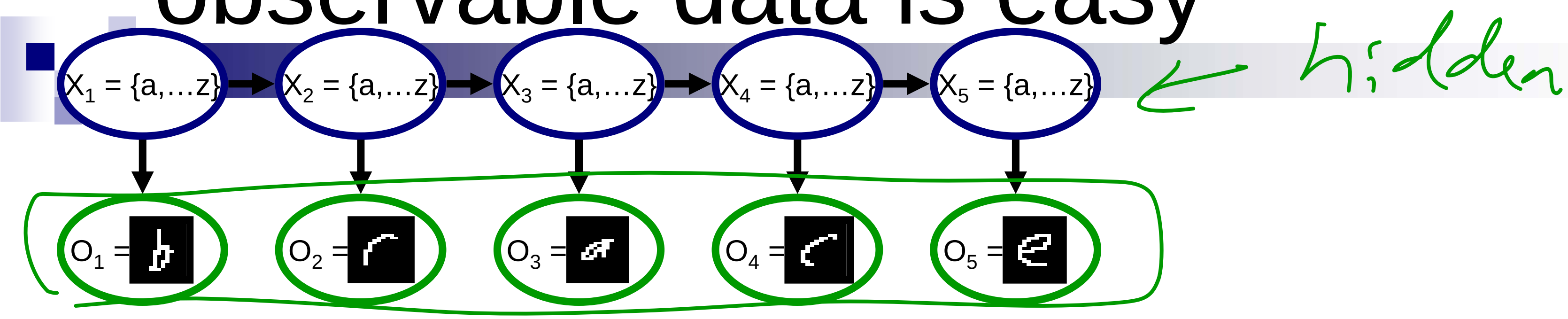
Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

April 16th, 2007

Learning HMMs from fully observable data is easy



Learn 3 distributions:

$$P(X_1 = a) = \frac{\text{Count}(\text{\# first letter was } a)}{N\text{: dataset size}}$$

select training data where letter was a

$$P(O_i = \text{pixel 17 is white} \mid X_i = a) = \frac{\text{Count}(\text{pixel 17 was white}, X_i = a)}{\text{any [illegible] in}}$$

$$P(X_i = a \mid X_{i-} = b$$

)

What if O is observed, but X is hidden

Log likelihood for HMMs when **X** is hidden

$\mathbf{O} = (O_1, \ldots, O_n)$

$X = (x_1, \ldots, x_n)$

for m sequences

$\sum_{j=1}^{m} \log P(o^{(j)} \mid \theta)$

- Marginal likelihood – **O** is observed, **X** is missing
  - □ For simplicity of notation, training data consists of only one sequence:

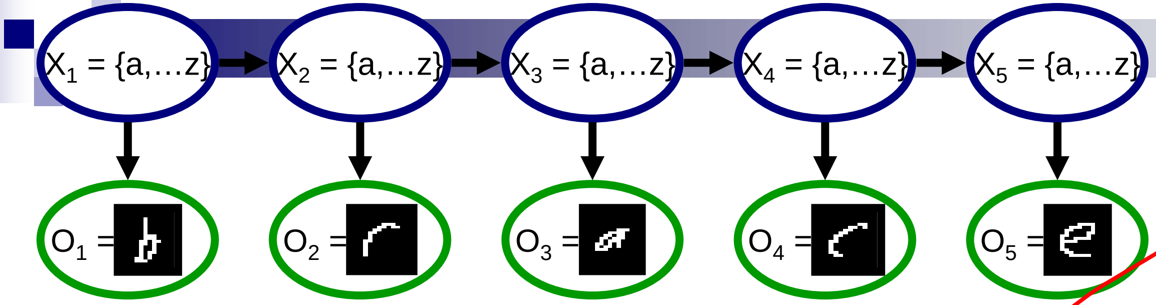
observed

$$
\begin{aligned}
\ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\
&= \underbrace{\log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta)}_{P(X_1 \mid \theta) \cdot P(O_1 \mid X_1, \theta) \cdot \prod_{t=2}^{n} P(X_t \mid X_{t-1}, \theta) \, P(O_t \mid X_t, \theta)}
\end{aligned}
$$

  - □ If there were m sequences:

$$
\underbrace{\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o}^{(j)} \mid \theta)}
$$

Computing Log likelihood for HMMs when **X** is hidden



$$\begin{aligned}\ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\ &= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta)\end{aligned}$$

if X_i takes on K values

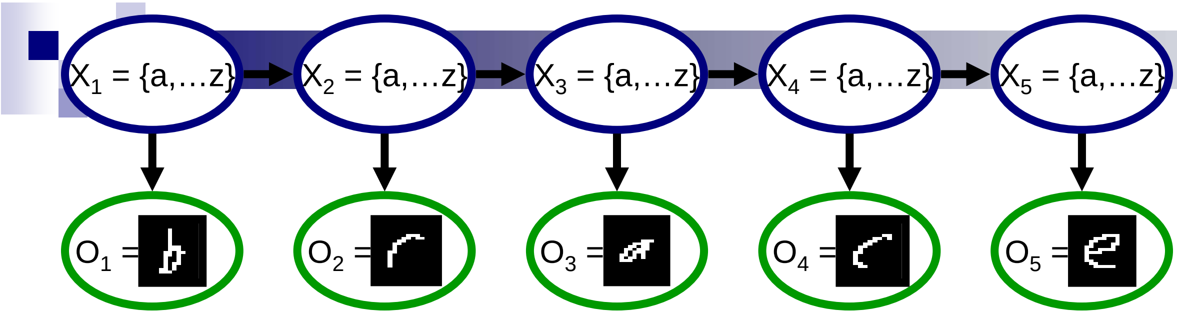
Sum over K^n assignments

$$= \log \sum_{x_1} \sum_{x_2} \dots \sum_{x_n} P(x_1) \cdot P(o_1 | x_1) \cdot \prod_{i=2}^{n} P(x_i | x_{i-1}) \cdot P(O_i | x_i)$$

$$= \log \sum_{x_1} \dots \sum_{x_{n-1}} P(x_1) P(o_1 | x_1) \prod_{i=2}^{n-1} P(x_i | x_{i-1}) P(o_i | x_i) \underbrace{\sum_{x_n} P(x_n | x_{n-1}) \cdot P(O_n | x_n)}_{B_{n-1}(x_{n-1})}$$

use VE. to compute $\ell(\theta : D)$ in $O(n)$ time

The M-step



- Maximization step:

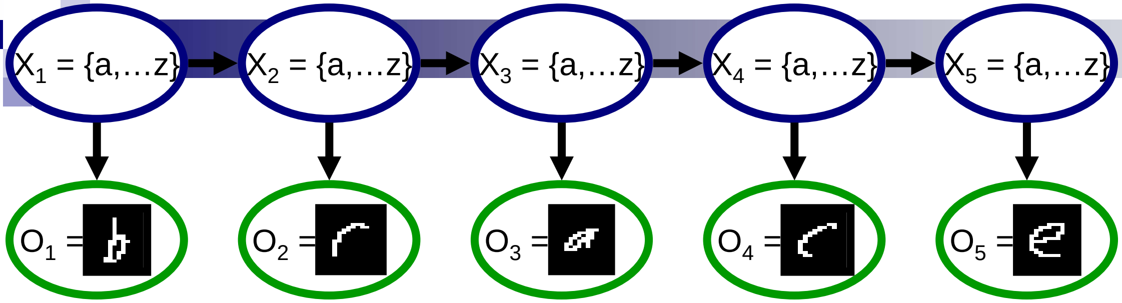
$$\theta^{(t+1)} \leftarrow \underline{\arg\max_\theta} \underbrace{\sum_{\mathbf{x}} Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \log P(\mathbf{x}, \mathbf{o} \mid \theta)}_{\text{weighted log likelihood}}$$

- Use expected counts instead of counts:
  - □ If learning requires Count(**x**,**o**)
  - □ Use $E_{Q(t+1)}$[Count(**x**,**o**)]

$$E_{Q^{(t+1)}}\left[Count\left(X=\underline{\{a,b,c\}},\ \underline{O=[\boxed{a}\ \boxed{b},\ \boxed{d}]}\right)\right] = \sum_{j=1}^{m} \underline{Q^{(t+1)}\left(x=(a,b,c) \mid \boxed{a}\boxed{b}\boxed{d}\right)^{(j)}} = \ \vdots$$

E-step revisited

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) = P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$



- E-step computes probability of hidden vars **x** given **o**
- Must compute:
  - □ $Q(x_t=a|\mathbf{o})$ – marginal probability of each position
    - Just forwards-backwards!
  - □ $Q(x_{t+1}=a,x_t=b|\mathbf{o})$ – joint distribution between pairs of positions

→ see reading [simple eqn.] [may be homework]

Exploiting unlabeled data in clustering

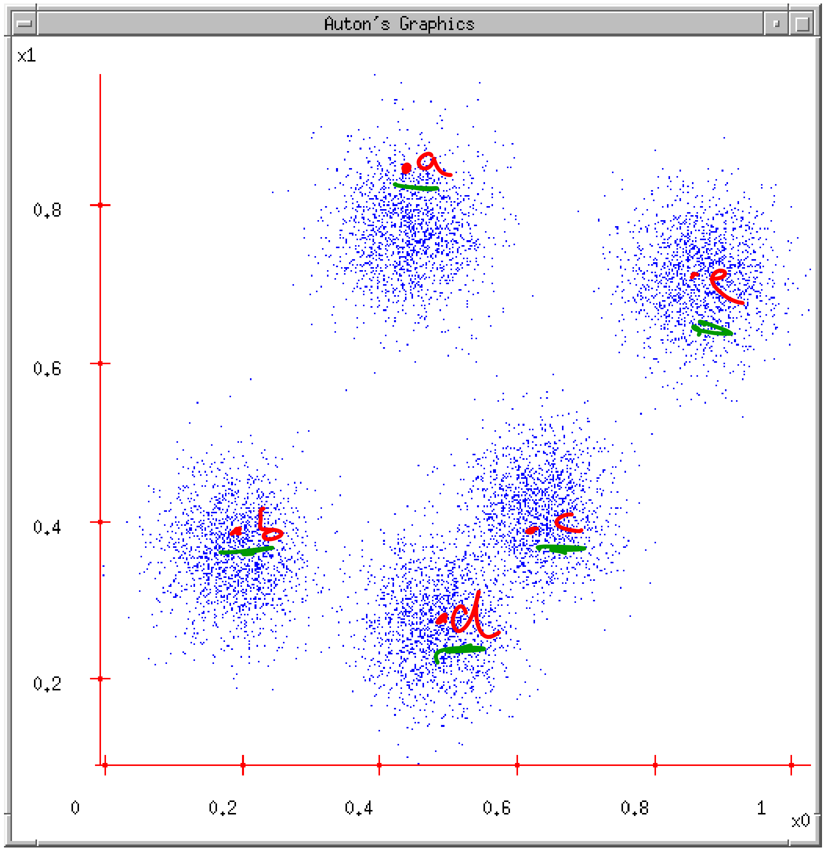
- A few data points are labeled
  - □ <x,o>
- Most points are unlabeled
  - □ <?,o>
- In the E-step of EM:
  - □ If i'th point is unlabeled:
    - compute $Q(X|o_i)$ as usual
  - □ If i'th point is labeled:
    - set $Q(X=x|o_i)=1$ and $Q(X\neq x|o_i)=0$

correct label

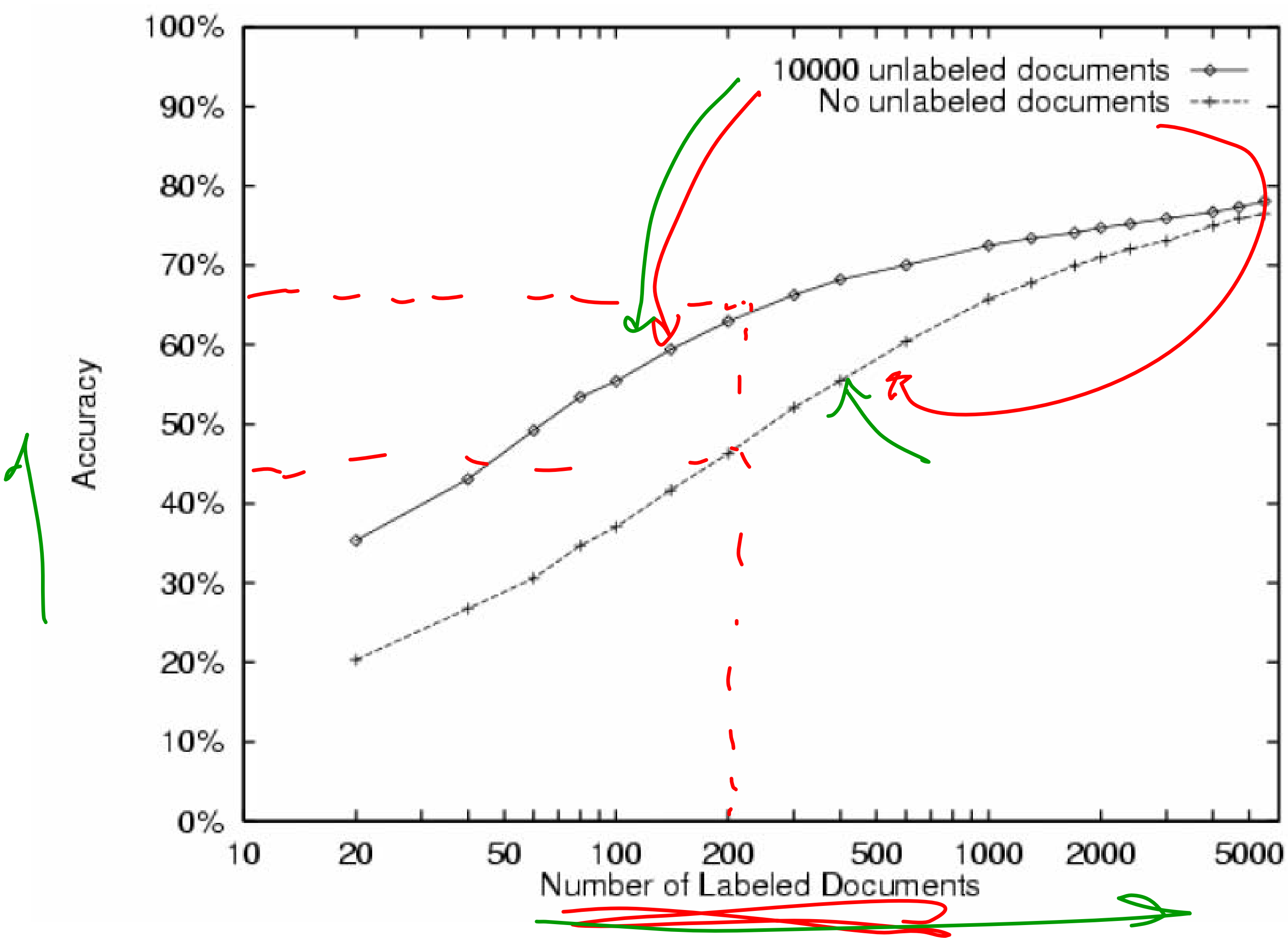
label

not label

- M-step as usual

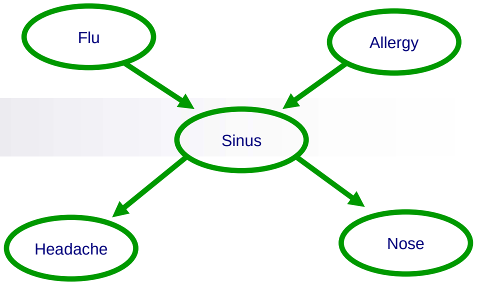


20 Newsgroups data – advantage of adding unlabeled data



$\log a \cdot b = \log a + \log b$

Data likelihood for BNs



- Given structure, log likelihood of fully observed data:

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G}) =$$

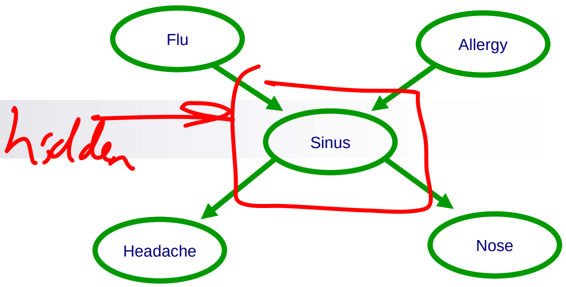
$$\log \prod_{j=1}^{m} P(f^{(j)} | \theta_F) \cdot P(a^{(j)} | \theta_A) \cdot P(s^{(j)} / a^{(j)}, f^{(j)}, \theta_{S|F,A}) \cdot P(h^{(j)} | s^{(j)}, \theta_{H|S}) \cdot P(n^{(j)} | s^{(j)}, \theta_{N|S})$$

$$= \overbrace{\left[\sum_{j=1}^{m} \log P(f^{(j)} | \theta_F)\right]}^{\text{learning Flu CPT}} + \overbrace{\left[\sum_{j=1}^{m} \log P(a^{(j)} | \theta_A)\right]}^{A\ \text{CPT}} + \overbrace{\left[\sum_{j=1}^{m} \log P(s^{(j)} / f^{(j)}, a^{(j)}, \theta_{S|FA})\right]}^{S|FA\ \text{CPT}}$$

$+ \ldots$

become independent learning problems

Marginal likelihood



- What if S is hidden?

$$\log P(\mathcal{D} \mid \theta_{\mathcal{G}}, \mathcal{G})$$

$$= \sum_{j=1}^{m} \log \sum_{s} P(a^{(j)} \mid \theta_A)\, P(f^{(j)} \mid \theta_F) \cdot P(s \mid f^{(j)}, a^{(j)}, \theta_{S|FA})$$

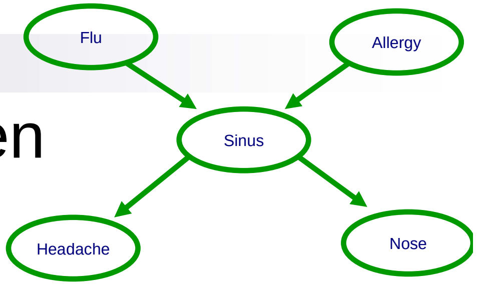
$$\cdot P(h^{(j)} \mid s, \theta_{H|S}) \cdot P(n^{(j)} \mid s, \theta_{N|S})$$

$\log \sum$ doesn't decompose

EM for BNs

same derivation (Jensen's, etc)

Log likelihood for BNs with hidden data

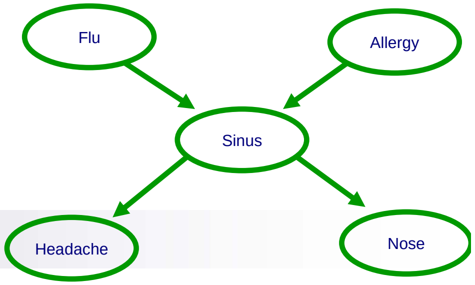


- Marginal likelihood – **O** is observed, **H** is hidden

$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log P(\mathbf{o}^{(j)} \mid \theta)$$

$$= \sum_{j=1}^{m} \log \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{o}^{(j)} \mid \theta)$$

E-step for BNs



- E-step computes probability of hidden vars **h** given **o**

$$Q^{(t+1)}(\underline{\mathbf{h} \mid \mathbf{o}}) \; = \; \underbrace{P(\mathbf{h} \mid \mathbf{o}, \theta^{(t)})}$$

if $|H| = 100$

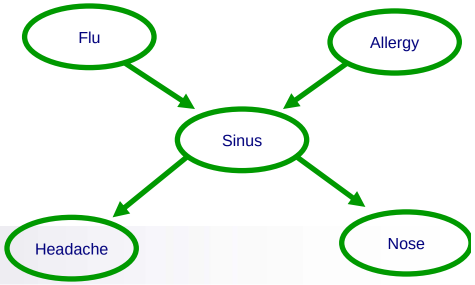
$K^{100} - 1$ params

(very large)

- Corresponds to inference in BN

VE.

The M-step for BNs

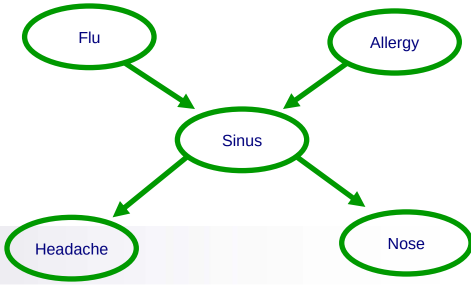


- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j} \sum_{\cancel{\mathbf{x}}\, h} Q^{(t+1)}(\mathbf{h} \mid \mathbf{o}^{(j)}) \log P(\mathbf{h}, \mathbf{o}^{(j)} \mid \theta)$$

- Use expected counts instead of counts:
  - □ If learning requires Count(**h**,**o**)
  - □ Use $E_{Q(t+1)}[\text{Count}(\mathbf{h},\mathbf{o})] = \sum_{j=1}^{m} \delta(o^{(j)} = o) \cdot Q^{(t+1)}(h \mid o^{(j)})$

M-step for each CPT



MLE CPT

$$\hat{p}(S=t \mid A=f, F=t) = \frac{\text{Count}(S=t, A=f, F=t)}{\text{count}(A=f, F=t)}$$

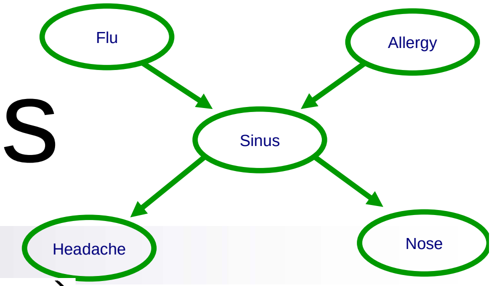
- M-step decomposes per CPT
  - □ Standard MLE:

$$\hat{P}(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{Count}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{Count}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

  - □ M-step uses expected counts:

$$P(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{ExCount}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{ExCount}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

Computing expected counts

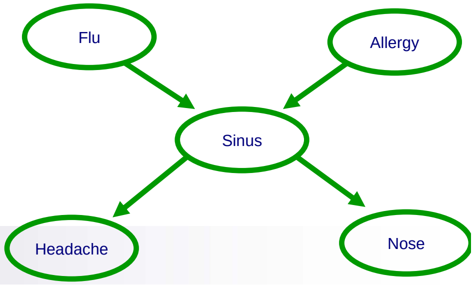


$$P(X_i = x_i \mid \mathbf{Pa}_{X_i} = \mathbf{z}) = \frac{\text{ExCount}(X_i = x_i, \mathbf{Pa}_{X_i} = \mathbf{z})}{\text{ExCount}(\mathbf{Pa}_{X_i} = \mathbf{z})}$$

- M-step requires expected counts:
  - □ For a set of vars **A**, must compute ExCount(**A**=**a**)
  - □ Some of **A** in example *j* will be observed
    - denote by $\mathbf{A}_O = \mathbf{a}_O^{\cancel{(j)}}$
  - □ Some of **A** will be hidden
    - denote by $\mathbf{A}_H = a_H$
- Use inference (E-step computes expected counts):
  - □ $\text{ExCount}^{(t+1)}(\mathbf{A}_O = \mathbf{a}_O^{\cancel{(j)}}, \mathbf{A}_H = \mathbf{a}_H) \leftarrow P(\cancel{\mathbf{A}_H = \mathbf{a}_H, \mathbf{A}_O = \mathbf{a}_O^{(j)} \mid \theta^{(t)}})$

$$= \sum_{j=1}^{m} \delta(A_O^{(j)} = a_O) \cdot \underbrace{P(A_H = a_H \mid o^{(j)}, \theta^{(t)})}_{\text{inference (VE)}}$$

Data need not be hidden in the same way



- When data is fully observed
  - □ A data point is <F=t, A=f, S=t, H=t, N=f>
- When data is partially observed
  - □ A data point is <F=t, A=?, S=?, H=t, N=f>
- But unobserved variables can be different for different data points
  - □ e.g., <F=t, A=t, S=t, H=?, N=?>,
    <F=?, A=f, S=t, H=?, N=f>
- Same framework, just change definition of expected counts
  - □ ExCount$^{(t+1)}$($\mathbf{A_O = a_O}^{\cancel{(j)}}$, $\mathbf{A_H = a_H}$) ← ~~$P(\mathbf{A_H = a_H}, \mathbf{A_O = a_O}^{(j)}|\theta^{(t)})$~~

    set of hidden vars are a function of j..

What you need to know

- EM for Bayes Nets
- E-step: inference computes expected counts
  - □ Only need expected counts over X_i and $\mathbf{Pa}_{x_i}$
- M-step: expected counts used to estimate parameters
- Hidden variables can change per datapoint

special case

- Use labeled and unlabeled data → some data points are complete, some include hidden variables

Announcements

- No recitation this week

  Spring Carnival

- On Wednesday, Special lecture on learning with text data by Prof. Noah Smith (LTI)

Co-Training for Semi-supervised learning

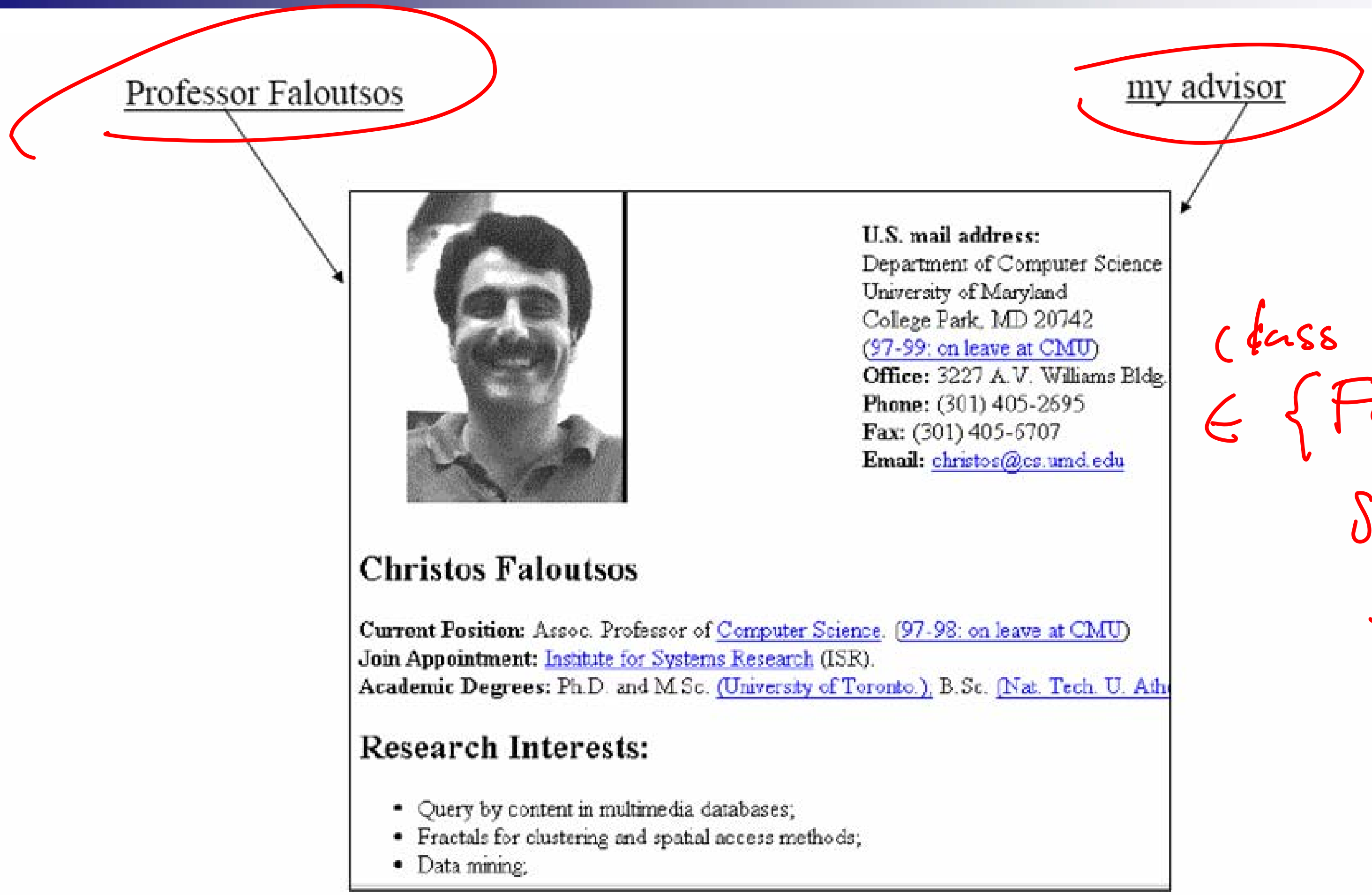
Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

April 16th, 2007

Redundant information



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

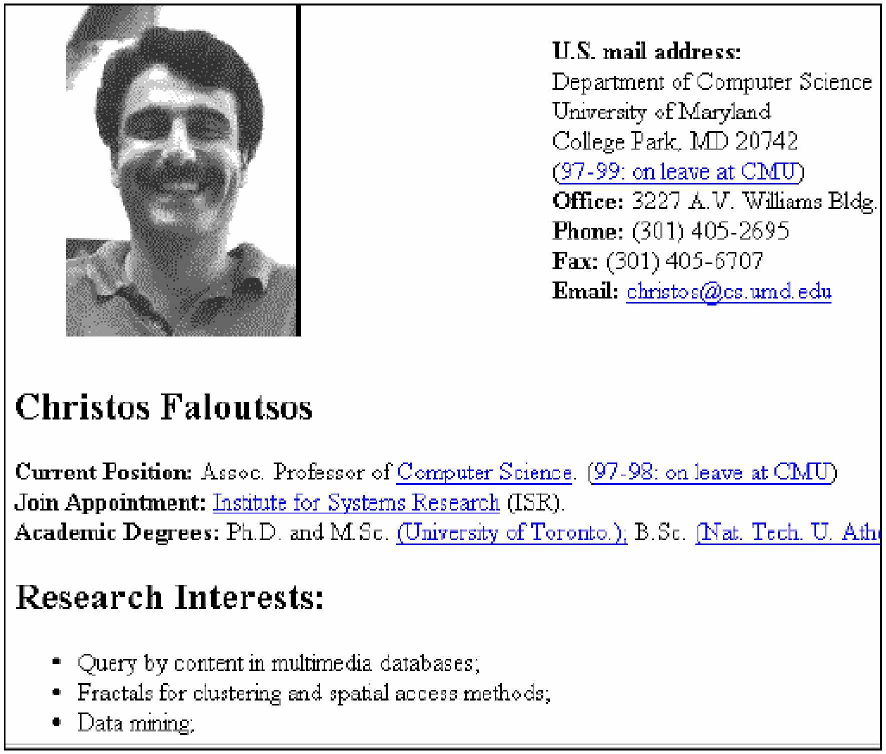
Current Position: Assoc. Professor of Computer Science. (97-98: on leave at CMU)
Join Appointment: Institute for Systems Research (ISR).
Academic Degrees: Ph.D. and M.Sc. (University of Toronto.), B.Sc. (Nat. Tech. U. Ath

Research Interests:

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

(class ∈ {Faculty, Student, project, ... }

Redundant information – webpage text



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

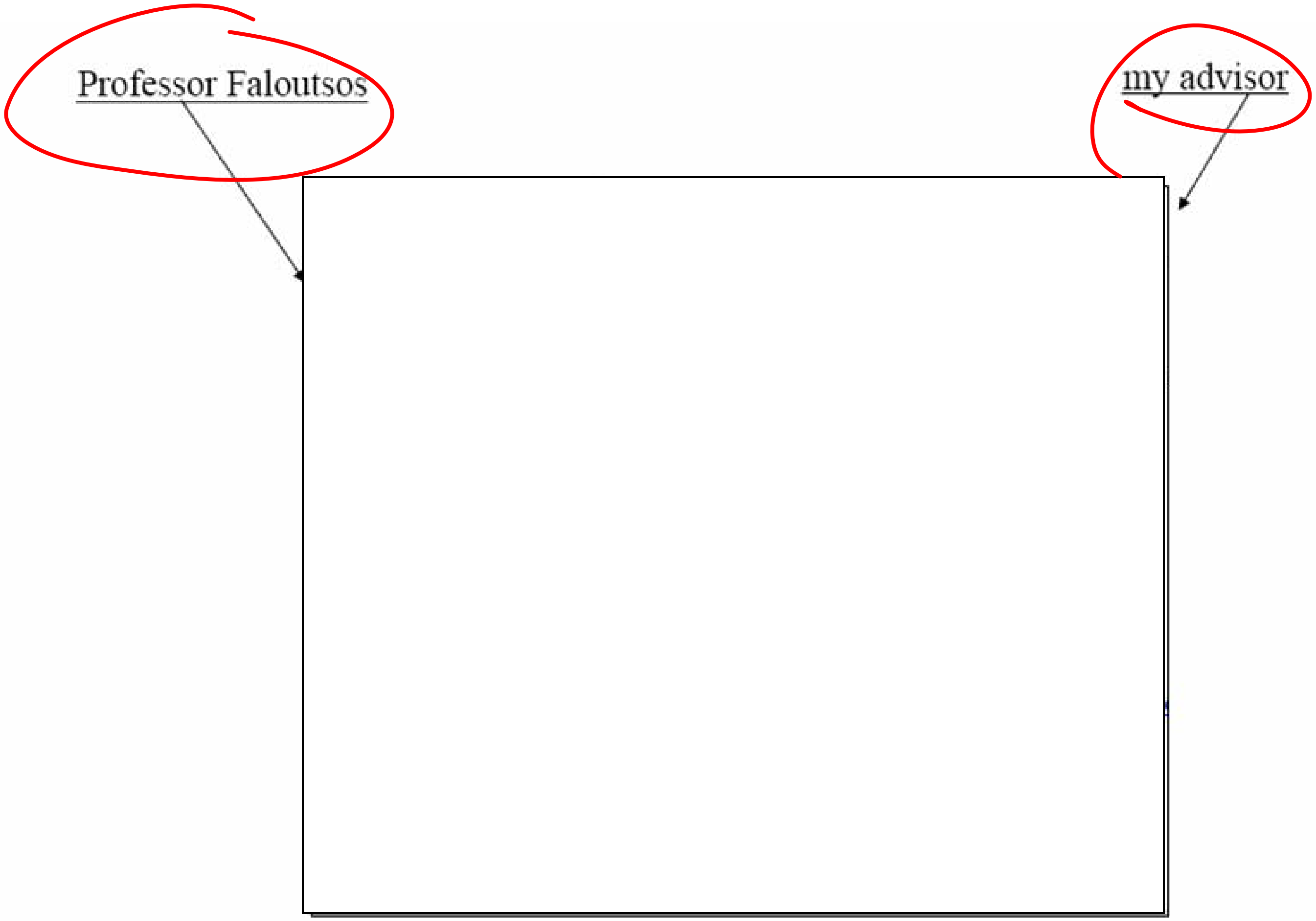
Christos Faloutsos

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:

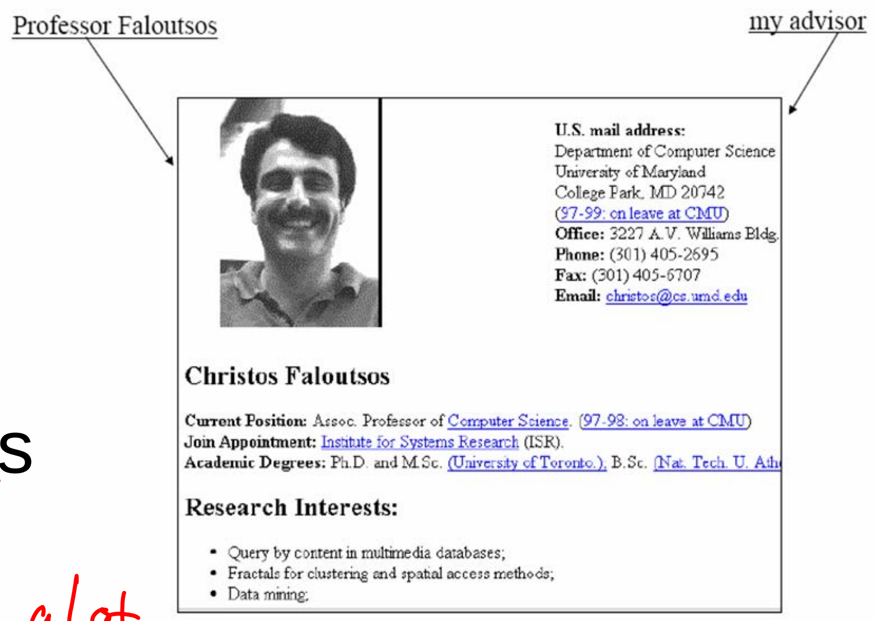
- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

Redundant information – anchor text for hyperlinks



Exploiting redundant information in semi-supervised learning

- Want to predict Y from features **X**
  - □ $f(\mathbf{X}) \mapsto Y$
  - □ have some labeled data **L**
  - □ lots of unlabeled data **U**
- Co-training assumption: **X** is very expressive
  - □ $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$
  - □ can learn
    - $g_1(\mathbf{X}_1) \mapsto Y$
    - $g_2(\mathbf{X}_2) \mapsto Y$



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;

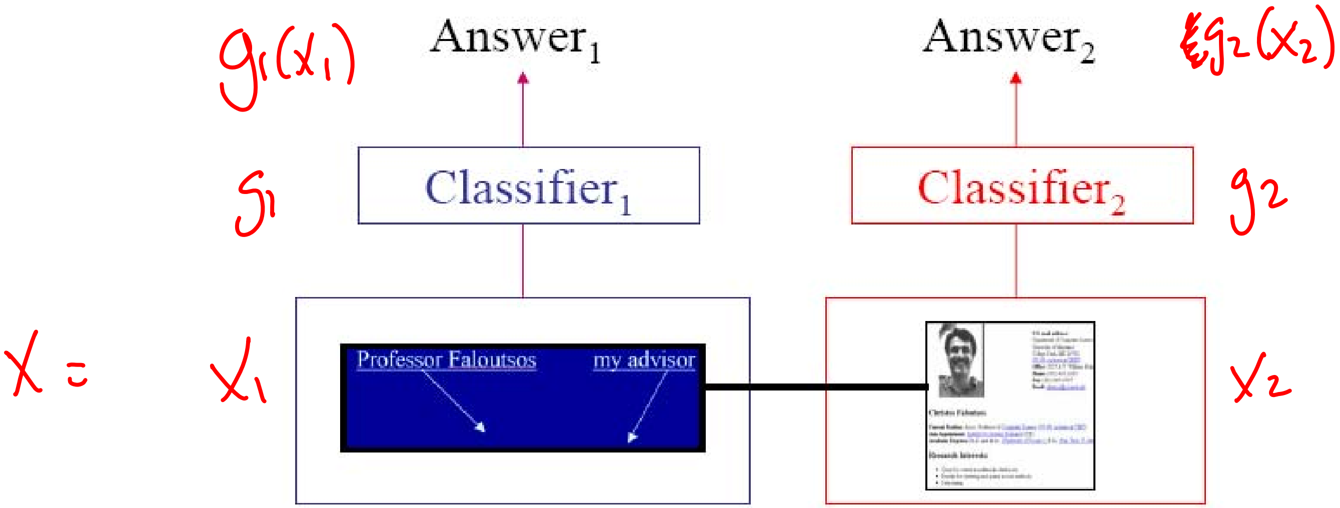
can do alot with unlabeled data, especially if $X_1 \perp X_2 \mid Y$

Co-Training

- Key idea: $\overset{g_1}{\text{Classifier}_1}$ and $\overset{g_2}{\text{Classifier}_2}$ must:
  - □ Correctly classify labeled data
  - □ **Agree** on unlabeled data

if x is labeled as y, I want $g_1(x_1)=y$ & $g_2(x_2)=y$

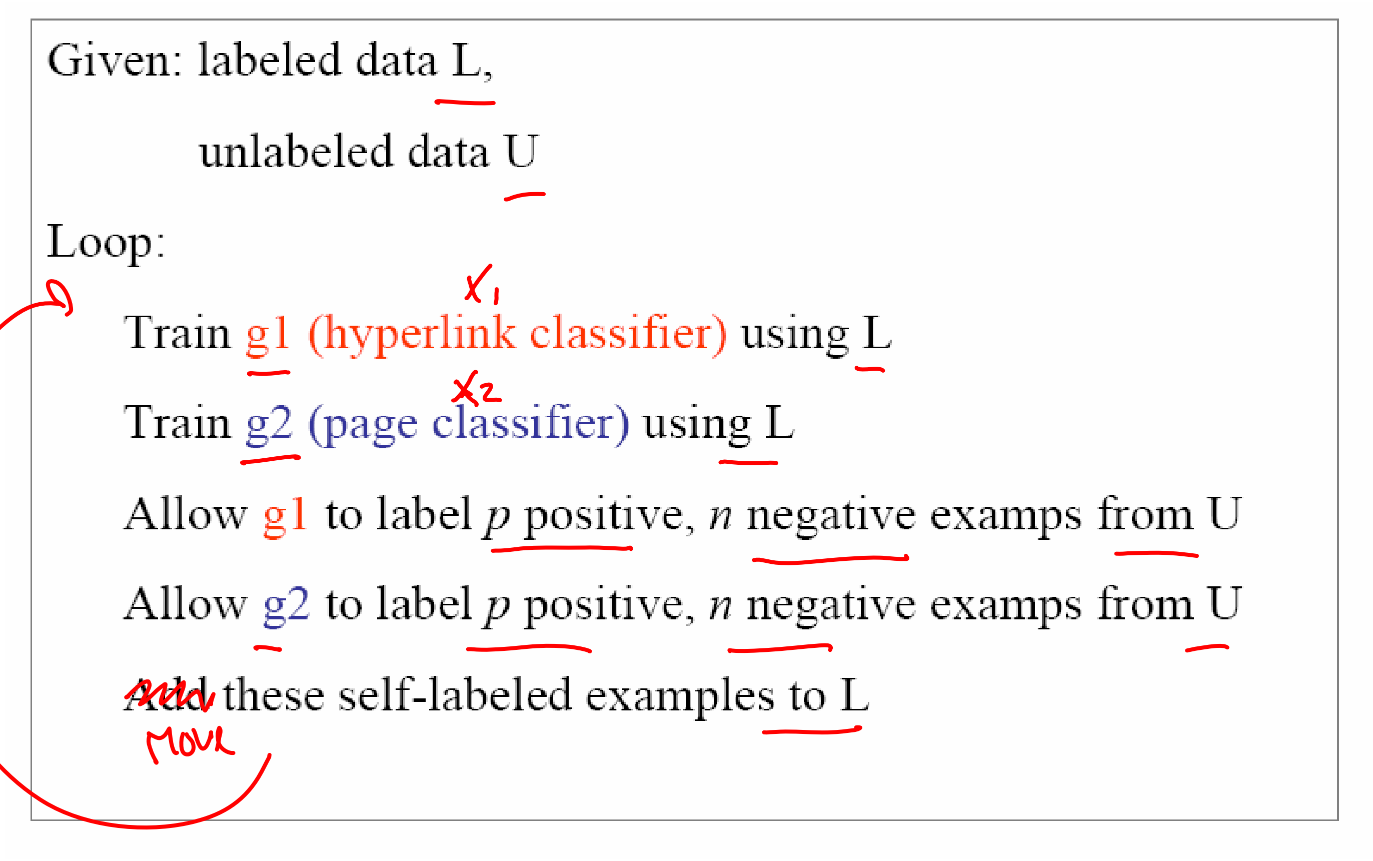
if x is unlabeled want $g_1(x_1) = g_2(x_2)$



Co-Training Algorithm

[Blum & Mitchell '99]

(example of the co-training principle)

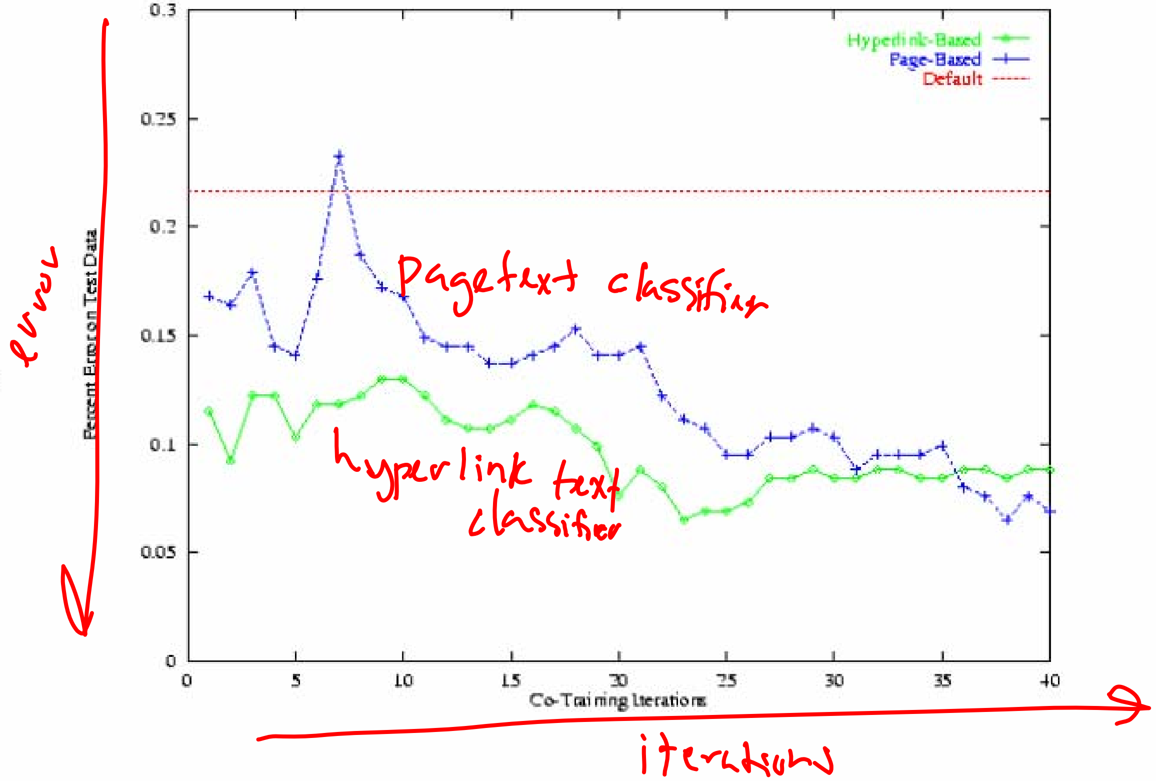


```
Given: labeled data L,
        unlabeled data U
Loop:
    Train g1 (hyperlink $x_1$ classifier) using L
    Train g2 (page $x_2$ classifier) using L
    Allow g1 to label p positive, n negative examps from U
    Allow g2 to label p positive, n negative examps from U
    ~~Add~~ move these self-labeled examples to L
```

Co-Training 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:



Co-Training theory

- Want to predict Y from features **X**
  - □ $f(\mathbf{X}) \mapsto Y$
- Co-training assumption: **X** is very expressive
  - □ $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$
  - □ want to learn $g_1(\mathbf{X}_1) \mapsto Y$ and $g_2(\mathbf{X}_2) \mapsto Y$
- *Assumption*: $\exists\ g_1, g_2, \forall\ \mathbf{x}\ g_1(\mathbf{x}_1) = f(\mathbf{x}), g_2(\mathbf{x}_2) = f(\mathbf{x})$
- Questions:
  - □ Does unlabeled data always help?
  - □ How many labeled examples do I need?
  - □ How many unlabeled examples do I need?

Understanding Co-Training: A simple setting

- Suppose **X**$_1$ and **X**$_2$ are discrete
  - □ |**X**$_1$| = |**X**$_2$| = N

possible values

if X_1 is described by ~~an~~ n ~~bigs~~ binary features, $N=2^n$

- No label noise
- Without unlabeled data, how hard is it to learn g$_1$ (or g$_2$)?

$|H| = 2^N$

hypothesis space

training examples is dependent on

$g_1 \in H$

$\ln|H| = N \cdot \ln 2$

X_1	
1	$\{+,-\}$
2	$\{+,-\}$
⋮	⋮
n	$\{+,-\}$

Co-Training in simple setting – Iteration 0

you get a web page with $X_1 = 12$ & $X_2 = 18$

X_1

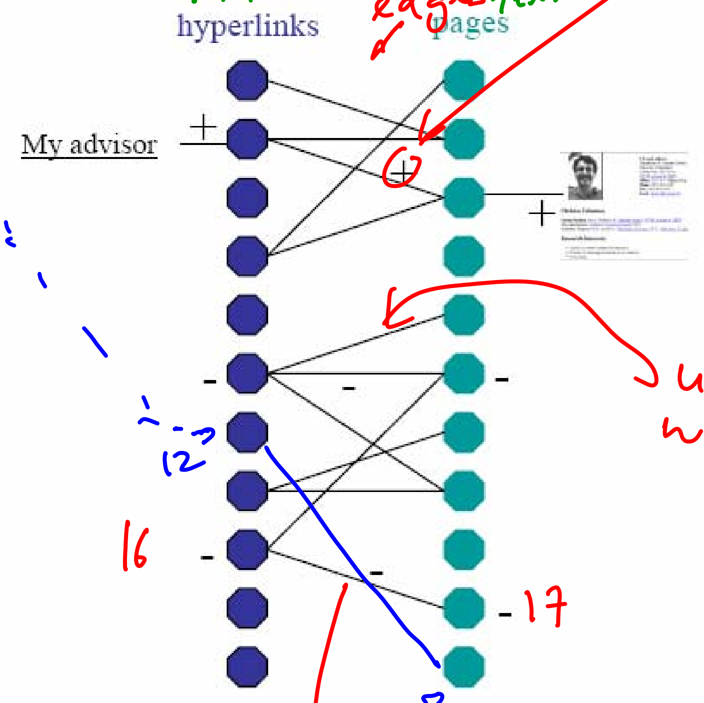
text of hyperlinks

set of webpages form edges text on pages

labeled data

X_2

edge $X_1 = x_1$ to $X_2 = x_2$ means x_1 & x_2 cooccurred on a webpage



unlabeled webpage

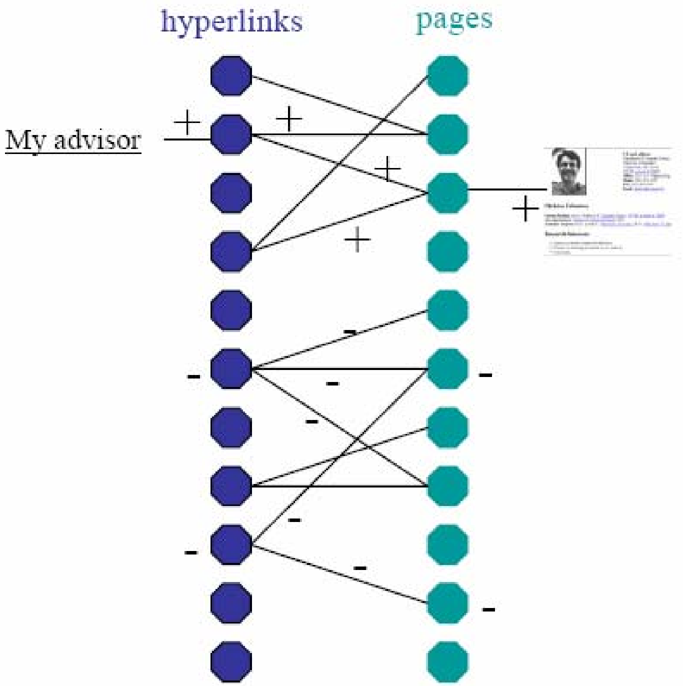
N

N

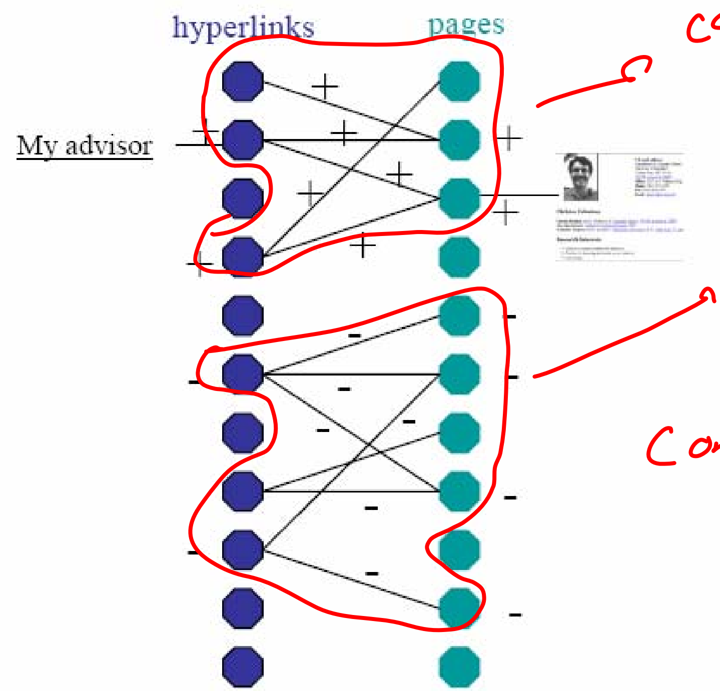
NO LABEL NOISE

one webpage $X_1 = 16$ & $X_2 = 17$

Co-Training in simple setting – Iteration 1



Co-Training in simple setting – after convergence

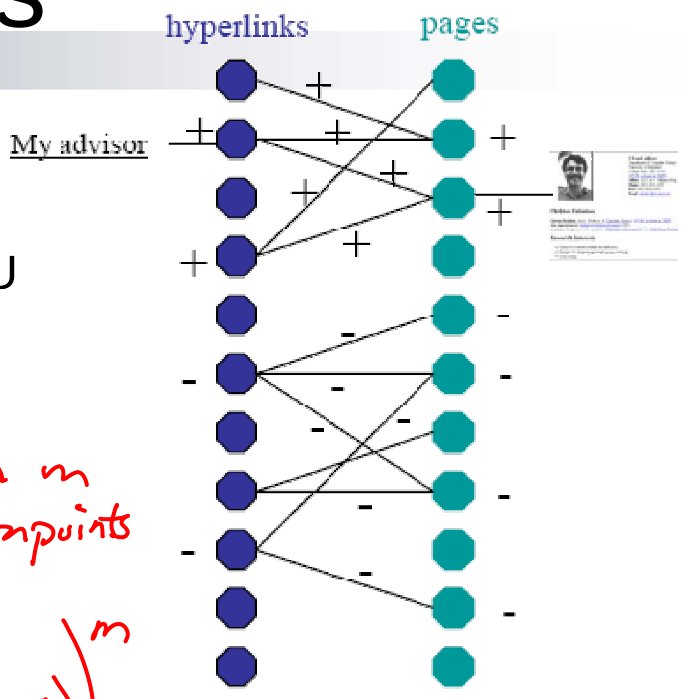


Connected component
+

no edges between them, because no label noise

component
−

Co-Training in simple setting – Connected components



- Suppose infinite **unlabeled** data
  - □ Co-training must have at least one labeled example in each connected component of L+U graph

component g_j

- What's probability of making an error?

with m datapoints

∃ connected component, where no data was labeled

test point x

$$E[error] = \sum_{g_j \in \text{components}} P(x \in g_j) \underbrace{\left(1 - P(x \in g_j)\right)^m}_{\text{no training data in } g_j}$$

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

Where g_j is the jth connected component of graph of L+U, m is number of labeled examples

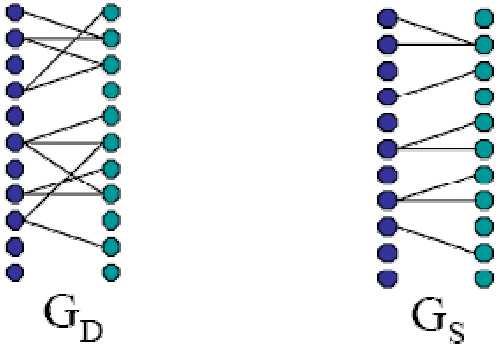
- For k Connected components, how much labeled data?

about K data points

instead of N

How much unlabeled data?

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



O(log(N)/α) 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

Co-Training theory

- Want to predict Y from features **X**
  - □ $f(\mathbf{X})$ Y
- Co-training assumption: **X** is very expressive
  - □ $\mathbf{X} = (\mathbf{X}_1,\mathbf{X}_2)$
  - □ want to learn $g_1(\mathbf{X}_1)$ Y and $g_2(\mathbf{X}_2)$ Y
- *Assumption*: $\exists\ g_1, g_2, \forall\ \mathbf{x}\ g_1(\mathbf{x}_1) = f(\mathbf{x}), g_2(\mathbf{x}_2) = f(\mathbf{x})$
- One co-training result [Blum & Mitchell ’99]
  - □ If
    - $(\mathbf{X}_1 \perp \mathbf{X}_2 \mid Y)$
    - g_1 & g_2 are PAC learnable from noisy data (and thus f)
  - □ Then
    - f is PAC learnable from weak initial classifier plus unlabeled data

What you need to know about co-training

- Unlabeled data can help supervised learning (a lot) when there are (mostly) independent redundant features
- One theoretical result:
  - If ($\mathbf{X}_1 \perp \mathbf{X}_2$ | Y) and g_1 & g_2 are PAC learnable from noisy data (and thus f)
  - Then f is PAC learnable from weak initial classifier plus unlabeled data
  - Disagreement between g_1 and g_2 provides bound on error of final classifier
- Applied in many real-world settings:
  - Semantic lexicon generation [Riloff, Jones 99] [Collins, Singer 99], [Jones 05]
  - Web page classification [Blum, Mitchell 99]
  - Word sense disambiguation [Yarowsky 95]
  - Speech recognition [de Sa, Ballard 98]
  - Visual classification of cars [Levin, Viola, Freund 03]

Acknowledgement

- I would like to thank Tom Mitchell for some of the material used in this presentation of co-training