Function Approximation from Partly Unlabeled Data

Machine Learning 10-701
April 2005

Tom M. Mitchell
Carnegie Mellon University

When can Unlabeled Data improve supervised learning?

Important question! In many cases, unlabeled data is plentiful, labeled data expensive

- Medical outcomes (x=<symptoms,treatment>, y=outcome)
- Text classification (x=document, y=relevance)
- Customer modeling (x=user actions, y=user intent)
- Sensor interpretation (x=<video,audio>, y=who's there)

When can Unlabeled Data help supervised learning?

Problem setting:

- Set *X* of instances drawn from unknown distribution *P(X)*
- Wish to learn target function *f: X* → *Y* (or, P(Y|X))
- Given a set *H* of possible hypotheses for *f*

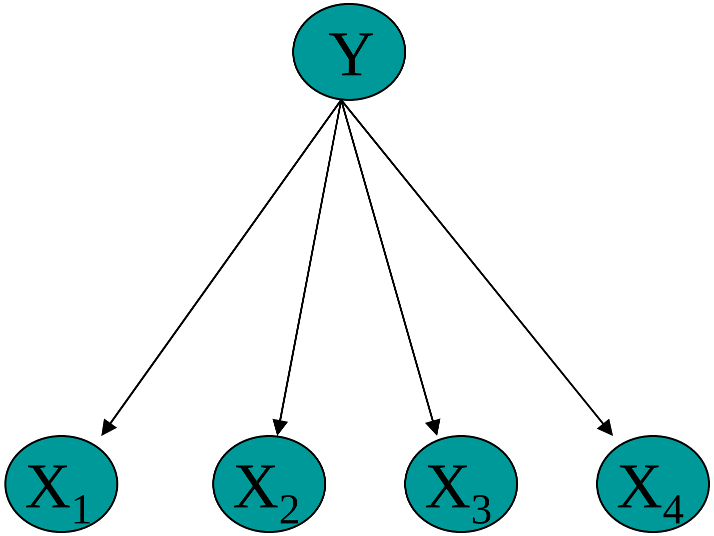
Given:

- iid labeled examples $L = \{\langle x_1, y_1 \rangle \ldots \langle x_m, y_m \rangle\}$
- iid unlabeled examples $U = \{x_{m+1}, \ldots x_{m+n}\}$

Wish to determine:

$$\hat{f} \leftarrow \arg\min_{h \in H} \Pr_{x \in P(X)} [h(x) \neq f(x)]$$

Idea 1: Use Labeled and Unlabeled Data to Train Bayes Net for P(X,Y)



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

Learn Bayes net for P(X1, X2, X3, X4,Y), then use this to infer P(Y|X1, X2, X3, X4)

E Step:

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

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

M Step:

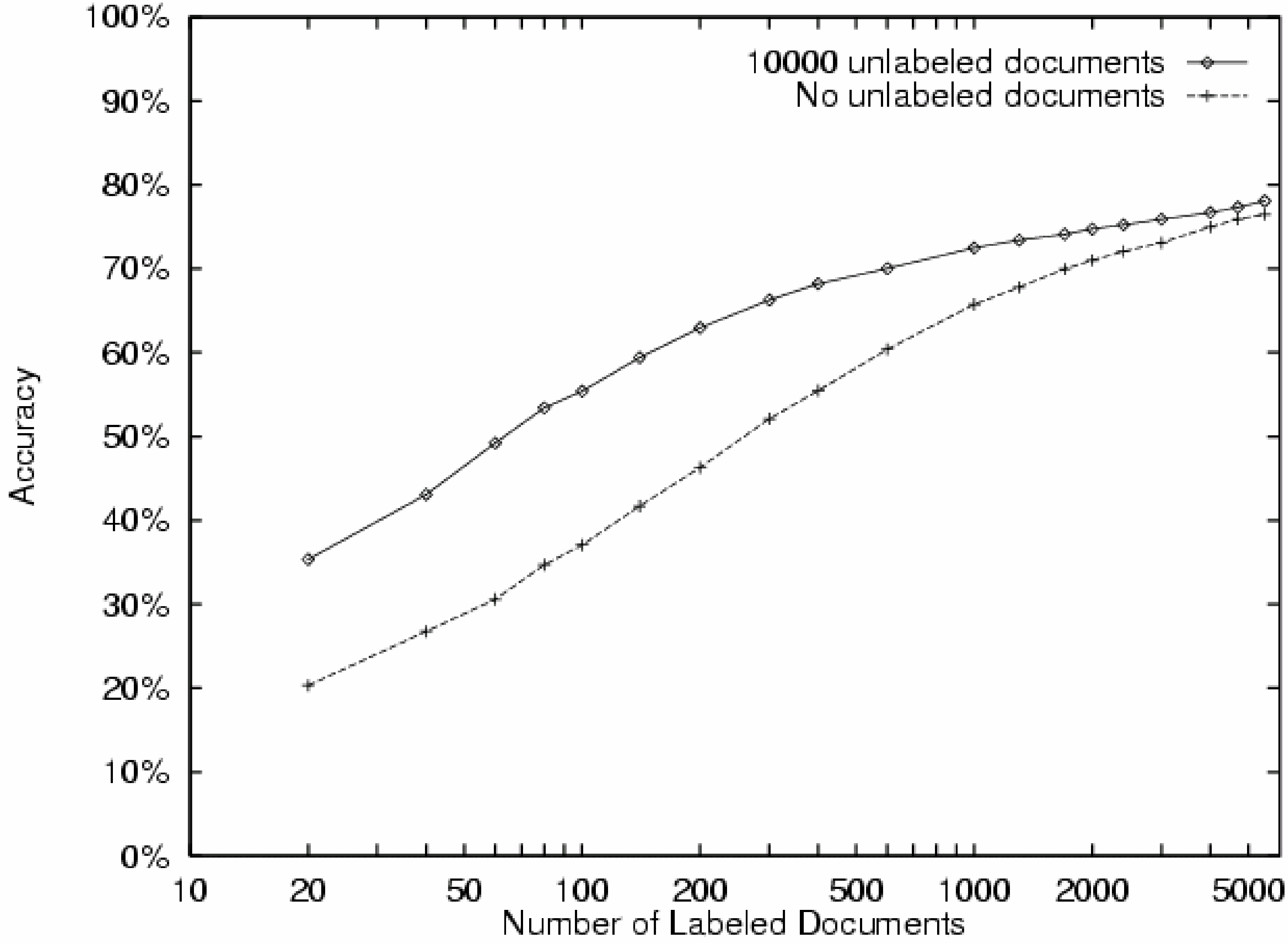
w_t is t-th word in vocabulary

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

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

20 Newsgroups

[Nigam, et al., 2000]



Idea 2: Use U to reweight labeled examples

- Most learning algorithms *minimize errors over labeled examples*
- But we really want to *minimize error over future examples* drawn from the same underlying distribution
- If we know the underlying distribution, we could weight each training example by its probability according to this distribution
- Unlabeled data allows us to estimate this underlying distribution

Idea 2: Use U to reweight labeled examples L

Use $U \rightarrow \widehat{P}(X)$ to alter the loss function

- Wish to find:

$$\widehat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) P(x)$$

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

- Usually approximate this as:

$$\widehat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x,y \rangle \in L} \delta(h(x) \neq y)$$

Which equals:

$$\widehat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq y) \left[\frac{n(x,L)}{|L|} \right]$$

n(x,L) = number of times x occurs in L

- Can produce a better approximation by incorporating U:

$$\widehat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) \left[\frac{n(x,L) + n(x,U)}{|L| + |U|} \delta(n(x,L) > 0) \right]$$

Reweighting Labeled Examples

- Wish to find

$$\hat{f} \leftarrow \arg\min_{h \in H} \sum_{x \in X} \delta(h(x) \neq f(x)) \left[\delta(n(x,L) > 0) \frac{n(x,L) + n(x,U)}{|L| + |U|} \right]$$

- Already have algorithm (e.g., decision tree learner) to find

$$\hat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x,y \rangle \in L} \delta(h(x) \neq y)$$

- Just reweight examples in L, and have algorithm minimize

$$\hat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x,y \rangle \in L} \delta(h(x) \neq y) \; \frac{n(x,L) + n(x,U)}{|L| + |U|}$$

- Or if X is continuous, use L+U to estimate p(X), and minimize

$$\hat{f} \leftarrow \arg\min_{h \in H} \frac{1}{L} \sum_{\langle x,y \rangle \in L} \delta(h(x) \neq y) \; \hat{p}(x)$$

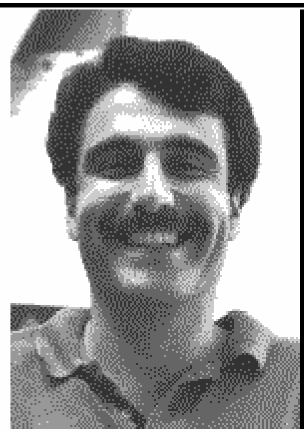
Idea 3: CoTraining

- In some settings, available data features are redundant and we can train two classifiers based on disjoint features
- In this case, the two classifiers should agree on the classification for each unlabeled example
- Therefore, we can use the unlabeled data to constrain joint training of both classifiers

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:

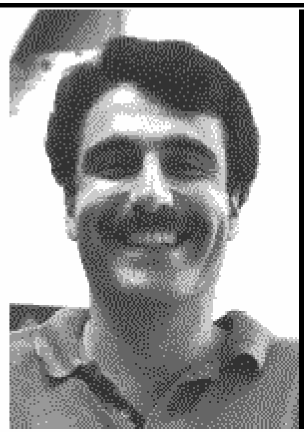
- Query by content in multimedia databases;
- Fractals for clustering and spatial access methods;
- Data mining;

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

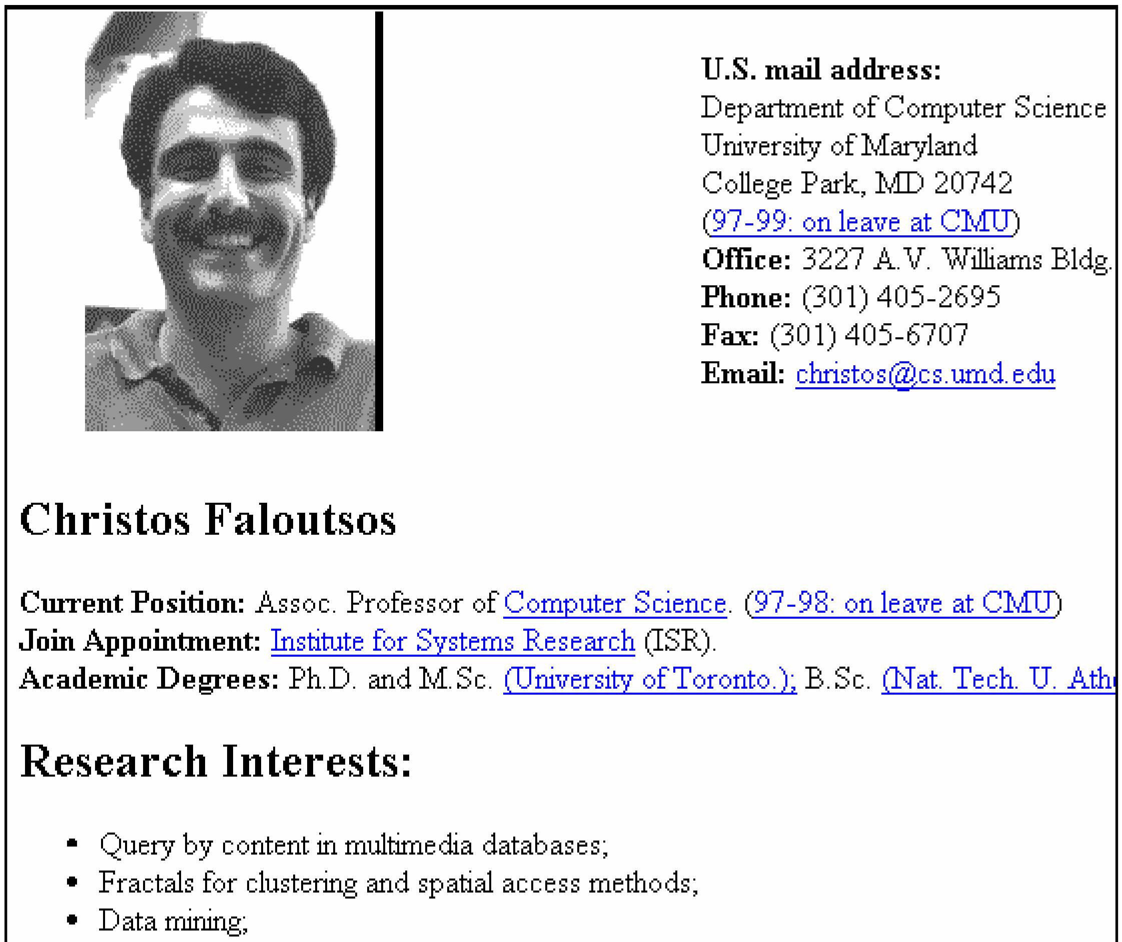
Research Interests:

- 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

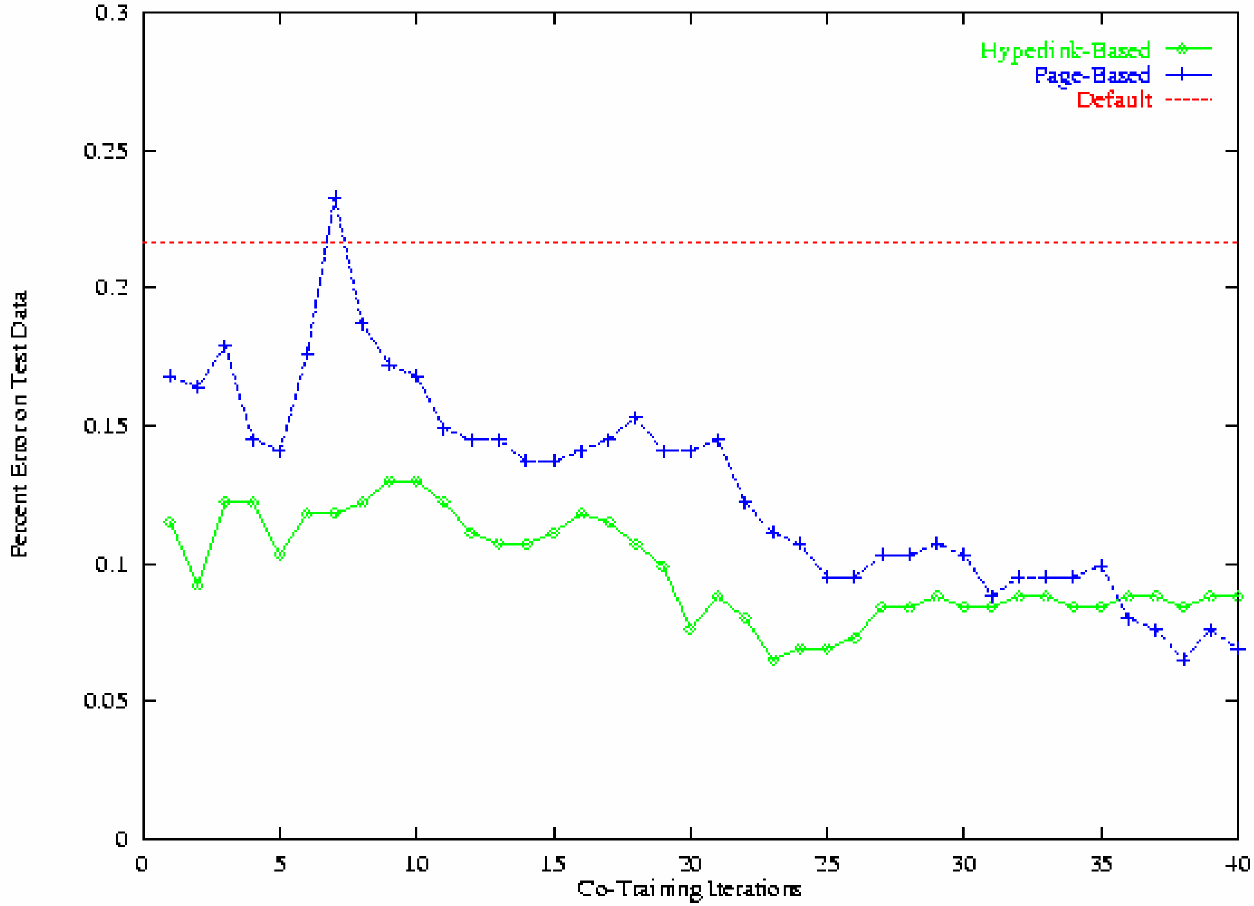
Allow g2 to label *p* positive, *n* negative examps from U

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:

- wish to learn f: X → Y, given L and U drawn from P(X)
- features describing X can be partitioned (X = X1 x X2)

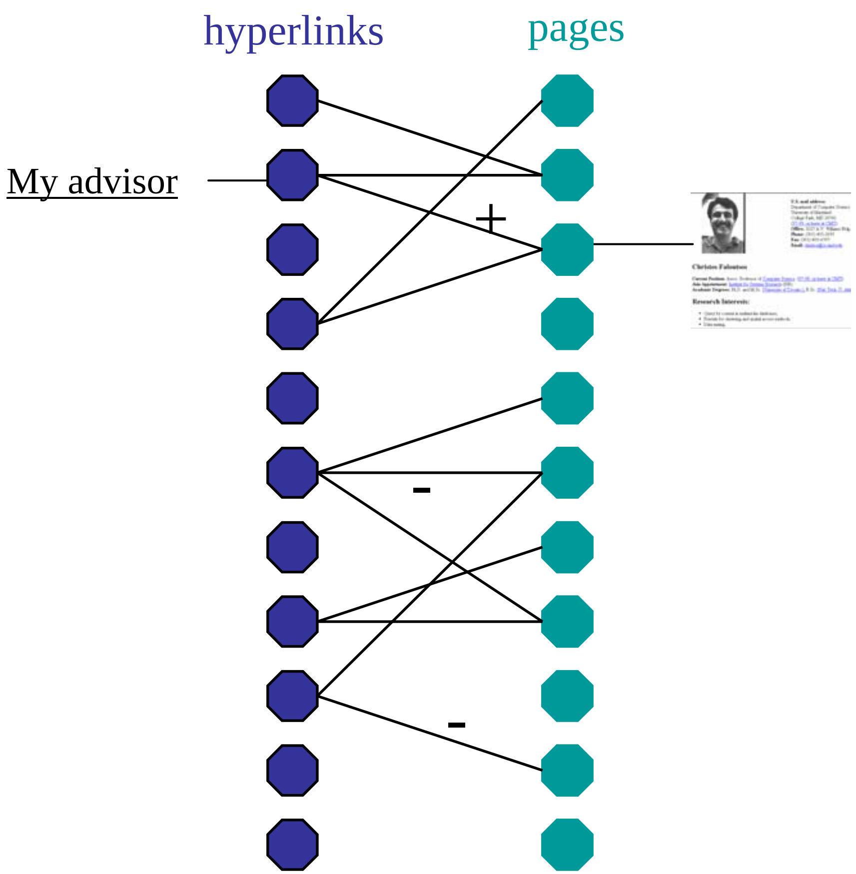
  such that f can be computed from either X1 or X2

$$(\exists g_1, g_2)(\forall x \in X) \quad g_1(x_1) = f(x) = g_2(x_2)$$

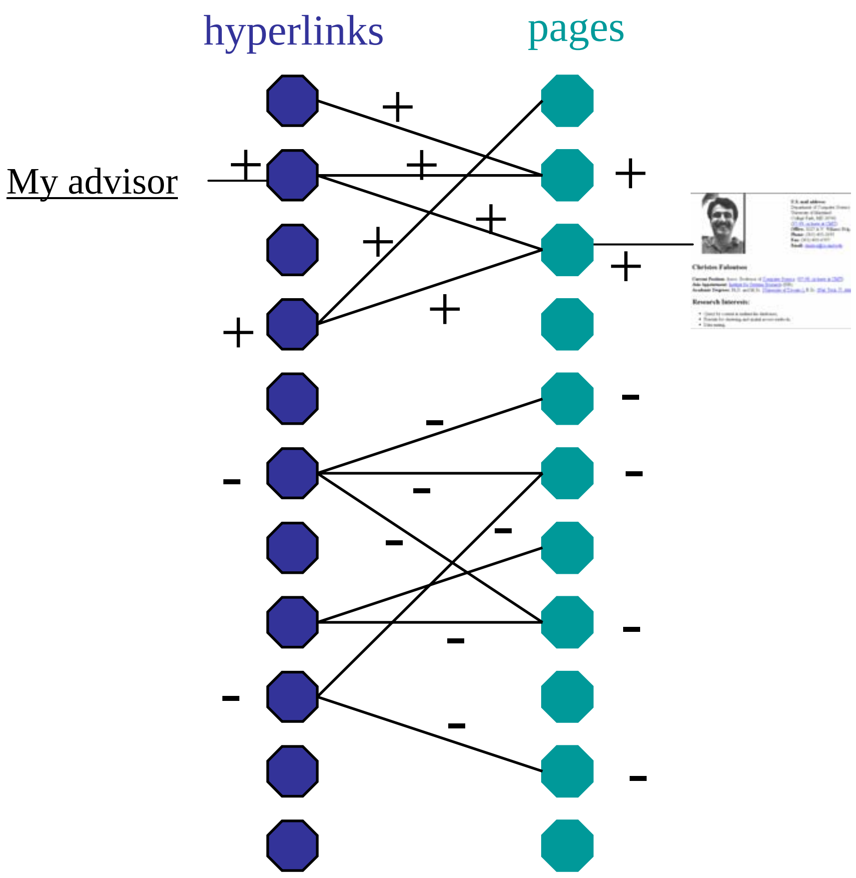
One result [Blum&Mitchell 1998]:

- If
  - X1 and X2 are 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



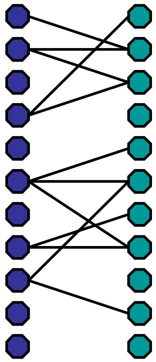
Co-Training Rote Learner



Expected Rote CoTraining error given *m* examples

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

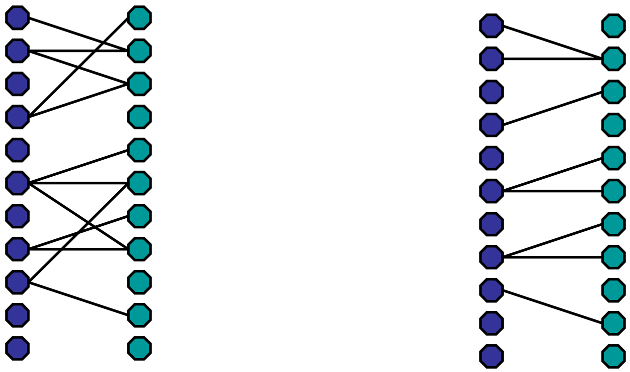
$$E[error] = \sum_j P(x \in g_j)(1 - P(x \in g_j))^m$$



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

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]

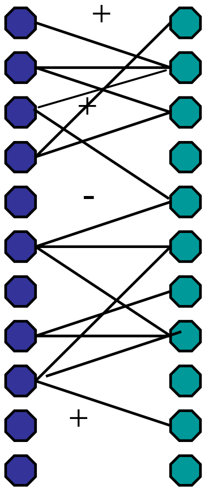
This theorem assumes X1 and X2 are conditionally independent given Y

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* $1 \le i \le k$ *then (a)* f *is a permutation and (b) for all* $1 \le i \le k$*,*

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

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

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

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

Disagreement over unlabeled

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

Misfit to estimated class priors

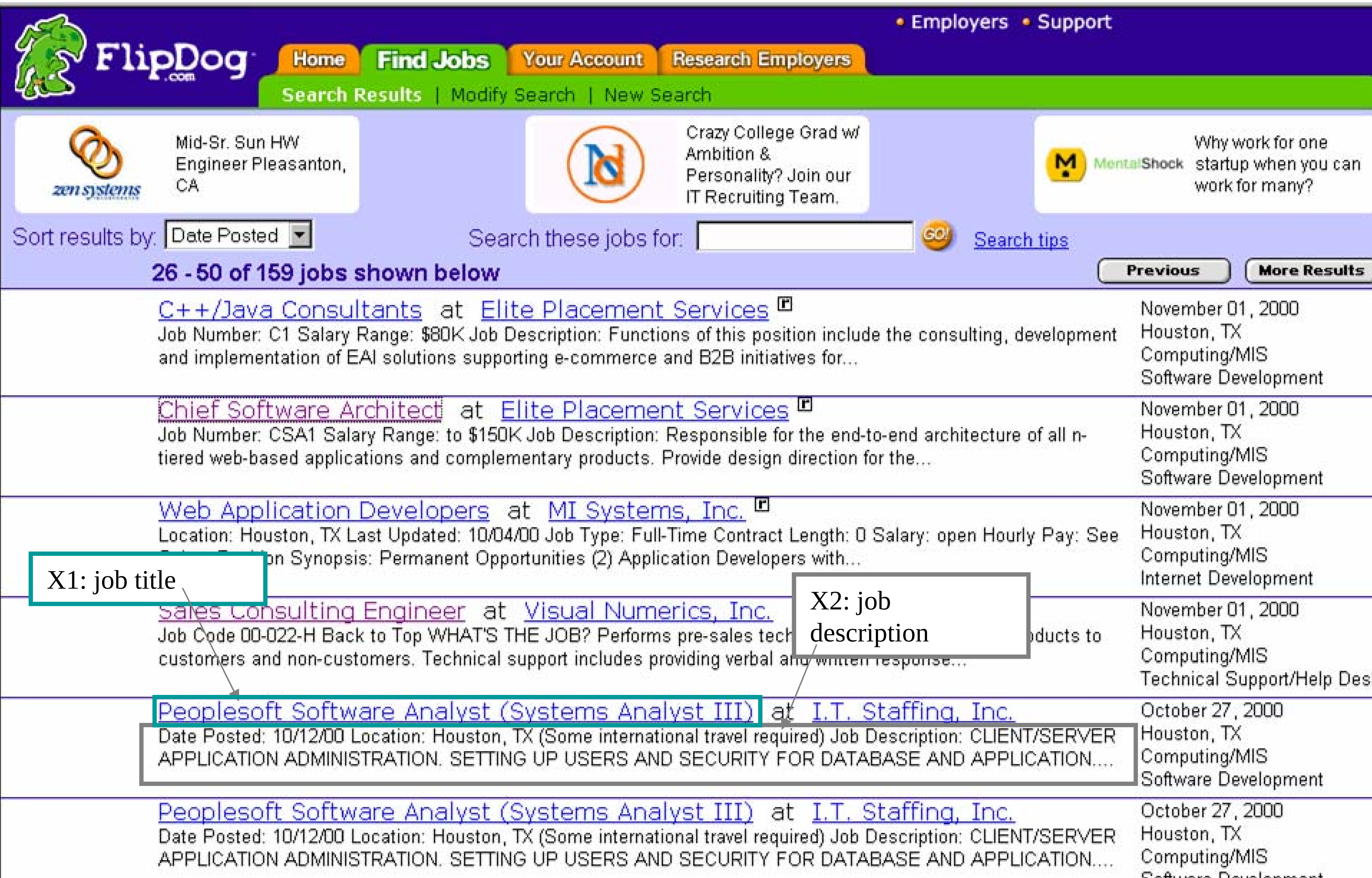
$$E4 = \left(\left(\frac{1}{|L|} \sum_{<x,y> \in L} y \right) - \left(\frac{1}{|L| + |U|} \sum_{x \in L \cup U} \frac{\hat{g}_1(x_1) + \hat{g}_2(x_2)}{2} \right) \right)^2$$

What Function Approximators?

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

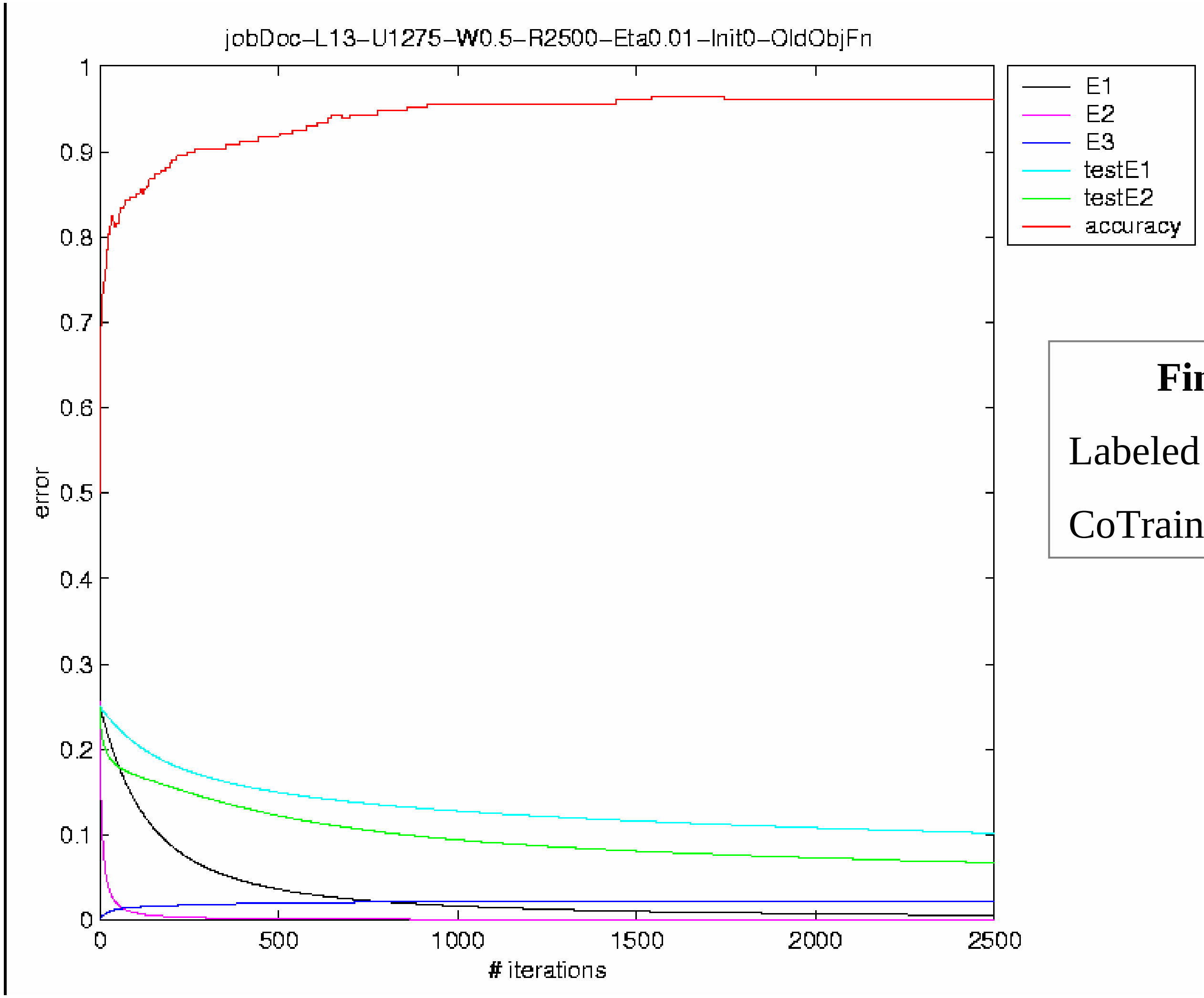
- Same functional form as logistic regression
- Use gradient descent to simultaneously learn g1 and g2, directly minimizing E = E1 + E2 + E3 + E4
- No word independence assumption, use both labeled and unlabeled data

Classifying Jobs for FlipDog



Gradient CoTraining

Classifying FlipDog job descriptions: SysAdmin vs. WebProgrammer



Final Accuracy

Labeled data alone: 86%

CoTraining: 96%

Gradient CoTraining

Classifying Capitalized sequences as Person Names

Eg., "Company president Mary Smith said today…"

x1 x2 x1

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]
  - Visual classification of cars [Levin, Viola, Freund 03]

4. Use U to Detect/Preempt Overfitting

- Overfitting is a problem for many learning algorithms (e.g., decision trees, neural networks)

- The symptom of overfitting: complex hypothesis h2 performs better on training data than simpler hypothesis h1, but worse on test data

- Unlabeled data can help detect overfitting, by comparing predictions of h1 and h2 over the unlabeled examples
  - The rate at which h1 and h2 disagree on U should be the same as the rate on L, unless overfitting is occuring

4. Use U to Detect/Preempt Overfitting

Define *metric* over $H \cup \{f\}$

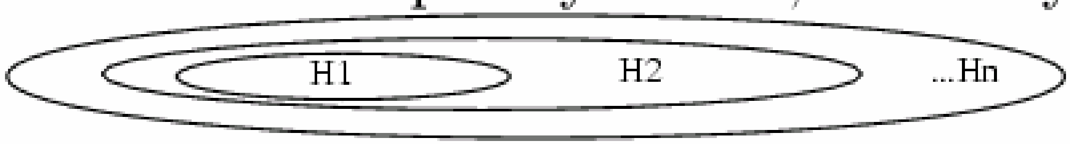
definition → $d(h_1, h_2) \equiv \int \delta(h_1(x) \neq h_2(x)) p(x) dx$

estimates →

$$\hat{d}(h_1, f) = \frac{1}{|L|} \sum_{x_i \in L} \delta(h_1(x_i) \neq y_i)$$

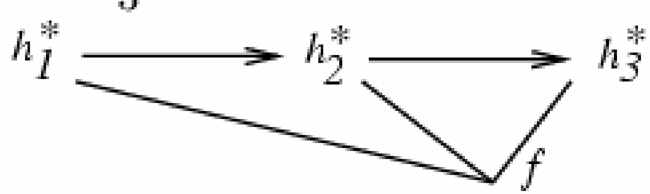
$$\hat{d}(h_1, h_2) = \frac{1}{|U|} \sum_{x \in U} \delta(h_1(x) \neq h_2(x))$$

Organize H into complexity classes, sorted by $P(h)$



Let h_i^* be hypothesis with lowest $\hat{d}(h, f)$ in H_i

Prefer h_1^*, h_2^*, or h_3^*?



- Definition of distance metric
  - Non-negative $d(f,g) \geq 0;$
  - symmetric $d(f,g)=d(g,f)$;
  - triangle inequality $d(f,g) \leq d(f,h)+d(h,g)$

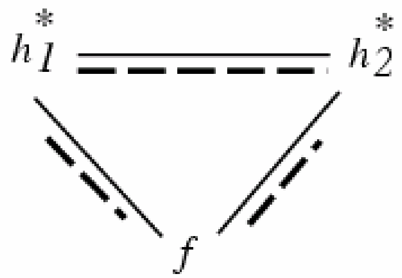
- Classification with zero-one loss:

$$d(h_1, h_2) \equiv \int \delta(h_1(x) \neq h_2(x)) p(x) dx$$

- Regression with squared loss:

$$d(h_1, h_2) \equiv \sqrt{\int (h_1(x) - h_2(x))^2 p(x) dx}$$

Idea: Use U to Avoid Overfitting



Note:

- $\hat{d}(h_i^*, f)$ optimistically biased (too short)
- $\hat{d}(h_i^*, h_j^*)$ unbiased
- Distances must obey triangle inequality!

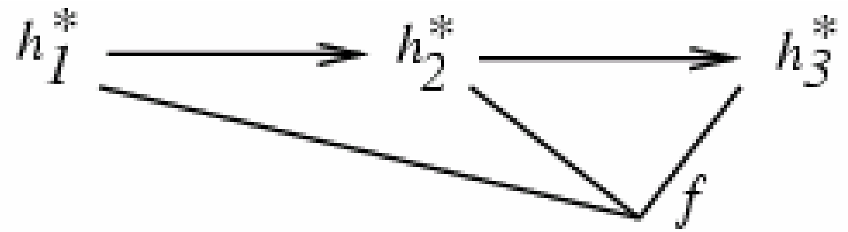
$$d(h_1, h_2) \leq d(h_1, f) + d(f, h_2)$$

→ Heuristic:

- Continue training until $\hat{d}(h_i, h_{i+1})$ fails to satisfy triangle inequality

Procedure TRI

- Given hypothesis sequence $h_0, h_1, \ldots$
- Choose the last hypothesis h_ℓ in the sequence that satisfies the triangle inequality $d(h_k, h_\ell) \leq d(\widehat{h_k, \mathrm{P}}_{Y|X}) + d(\widehat{h_\ell, \mathrm{P}}_{Y|X})$ with every preceding hypothesis h_k, $0 \leq k < \ell$. (Note that the inter-hypothesis distances $d(h_k, h_\ell)$ are measured on the *un*labeled training data.)



Experimental Evaluation of TRI
[Schuurmans & Southey, MLJ 2002]

- Use it to select degree of polynomial for regression
- Compare to alternatives such as cross validation, structural risk minimization, …

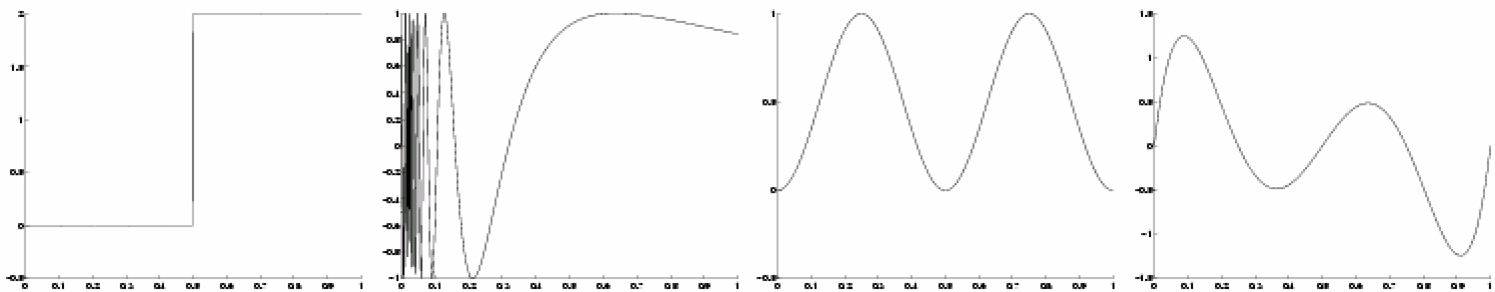


Figure 5: Target functions used in the polynomial curve fitting experiments (in order): step($x \geq 0.5$), $\sin(1/x)$, $\sin^2(2\pi x)$, and a fifth degree polynomial.

Generated y values contain zero mean Gaussian noise ε

Y=f(x)+ε

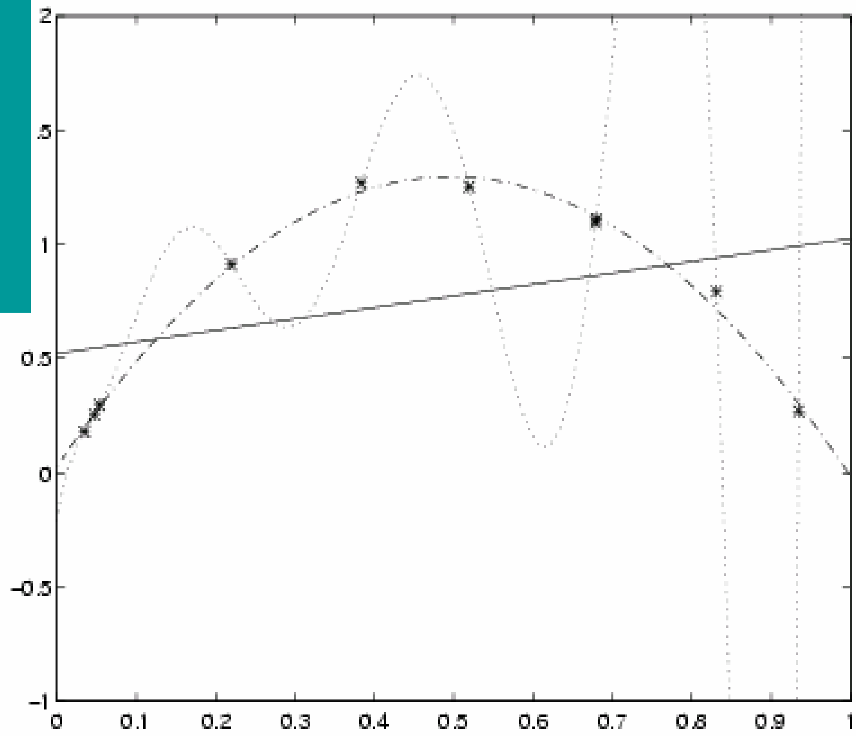


Figure 4: An example of minimum squared error polynomials of degrees 1, 2, and 9 for a set of 10 training points. The large degree polynomial demonstrates erratic behavior off the training set.

Approximation ratio:

$$\frac{\text{true error of selected hypothesis}}{\text{true error of best hypothesis considered}}$$

Results using 200 unlabeled, t labeled

Cross validation (Ten-fold) → CVT

Structural risk minimization → SRM

Worst performance in top .50 of trials → row 50

$t = 20$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.00	1.06	1.14	7.54	5.47	15.2	22.2	25.8	1.02
50	1.06	1.17	1.39	224	118	394	585	590	1.12
75	1.17	1.42	3.62	5.8e3	3.9e3	9.8e3	1.2e4	1.2e4	1.24
95	1.44	6.75	56.1	6.1e5	3.7e5	7.8e5	9.2e5	8.2e5	1.54
100	2.41	1.1e4	2.2e4	1.5e8	6.5e7	1.5e8	1.5e8	8.2e7	3.02

$t = 30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.00	1.08	1.17	4.69	1.51	5.41	5.45	2.72	1.06
50	1.08	1.17	1.54	34.8	9.19	39.6	40.8	19.1	1.14
75	1.19	1.37	9.68	258	91.3	266	266	159	1.25
95	1.45	6.11	419	4.7e3	2.7e3	4.8e3	5.1e3	4.0e3	1.51
100	2.18	643	1.6e7	1.6e7	1.6e7	1.6e7	1.6e7	1.6e7	2.10

Table 1: Fitting $f(x) = \text{step}(x \geq 0.5)$ with $P_X = U(0,1)$ and $\sigma = 0.05$. Tables give distribution of approximation ratios achieved at training sample size $t = 20$ and $t = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

$t = 20$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	2.04	1.03	1.00	1.00	1.06	1.00	1.01	1.58	1.02
50	3.11	1.37	1.33	1.34	1.94	1.35	1.61	18.2	1.32
75	3.87	2.23	2.30	2.13	10.0	2.75	4.14	1.2e3	1.83
95	5.11	9.45	8.84	8.26	5.0e3	11.8	82.9	1.8e5	3.94
100	8.92	105	526	105	2.0e7	2.1e3	2.7e5	2.4e7	6.30

$t = 30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.50	1.00	1.00	1.00	1.00	1.00	1.00	1.02	1.01
50	3.51	1.16	1.03	1.05	1.11	1.02	1.08	1.45	1.27
75	4.15	1.64	1.45	1.48	2.02	1.39	1.88	6.44	1.60
95	5.51	5.21	5.06	4.21	26.4	5.01	19.9	295	3.02
100	9.75	124	1.4e3	20.0	9.1e3	28.4	9.4e3	1.0e4	8.35

Table 4: Fitting $f(x) = \sin^2(2\pi x)$ with $P_X = U(0,1)$ and $\sigma = 0.05$. Tables give distribution of approximation ratios achieved at training sample size $t = 20$ and $t = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

Bound on Error of TRI Relative to Best Hypothesis Considered

Proposition 1 *Let* h_m *be the optimal hypothesis in the sequence* $h_0, h_1, \ldots$ *(that is,* $h_m = \arg\min_{h_k} d(h_k, \mathrm{P}_{Y|X})$*) and let* h_ℓ *be the hypothesis selected by TRI. If (i)* $m \leq \ell$ *and (ii)* $d(h_m, \widehat{\mathrm{P}}_{Y|X}) \leq d(h_m, \mathrm{P}_{Y|X})$ *then*

$$d(h_\ell, \mathrm{P}_{Y|X}) \;\leq\; 3d(h_m, \mathrm{P}_{Y|X}) \tag{6}$$

Extension to TRI:
Adjust for expected bias of training data estimates
[Schuurmans & Southey, MLJ 2002]

Procedure ADJ

- Given hypothesis sequence $h_0, h_1, \ldots$
- For each hypothesis h_ℓ in the sequence
  - multiply its estimated distance to the target $d(\widehat{h_\ell, \mathrm{P}_{Y|X}})$ by the worst ratio of unlabeled and labeled distance to some predecessor h_k to obtain an adjusted distance estimate $d(\widehat{\widehat{h_\ell, \mathrm{P}_{Y|X}}}) = d(\widehat{h_\ell, \mathrm{P}_{Y|X}}) \frac{d(h_k, h_\ell)}{d(\widehat{h_k, h_\ell})}$.
- Choose the hypothesis h_n with the smallest adjusted distance $d(\widehat{\widehat{h_n, \mathrm{P}_{Y|X}}})$.

Experimental results: averaged over multiple target functions, outperforms TRI

What you should know

1. Unlabeled can help EM learn Bayes nets for P(X,Y)

2. Use unlabeled data to reweight labeled examples

3. If problem has redundantly sufficient features, CoTrain multiple classifiers, using unlabeled data as constraints

4. Use unlabeled data to detect/preempt overfitting

Further Reading

- EM for Naïve Bayes classifiers: K.Nigam, et al., 2000. "Text Classification from Labeled and Unlabeled Documents using EM", *Machine Learning,* 39, pp.103—134.

- CoTraining: A. Blum and T. Mitchell, 1998. “Combining Labeled and Unlabeled Data with Co-Training,” *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98).*
- S. Dasgupta, et al., “PAC Generalization Bounds for Co-training”, *NIPS 2001*

- Model selection: D. Schuurmans and F. Southey, 2002. “Metric-Based methods for Adaptive Model Selection and Regularizaiton,” Machine Learning, 48, 51—84.