10601
Machine Learning

Semi supervised learning

Can Unlabeled Data improve supervised learning?

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

- Medical outcomes (x=<patient,treatment>, y=outcome)
- Text classification (x=document, y=relevance)
- Customer modeling (x=user actions, y=user intent)
- …

When can Unlabeled Data help supervised learning?

Consider 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 $U = \{x_{m+1}, \ldots x_{m+n}\}$

Determine:

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

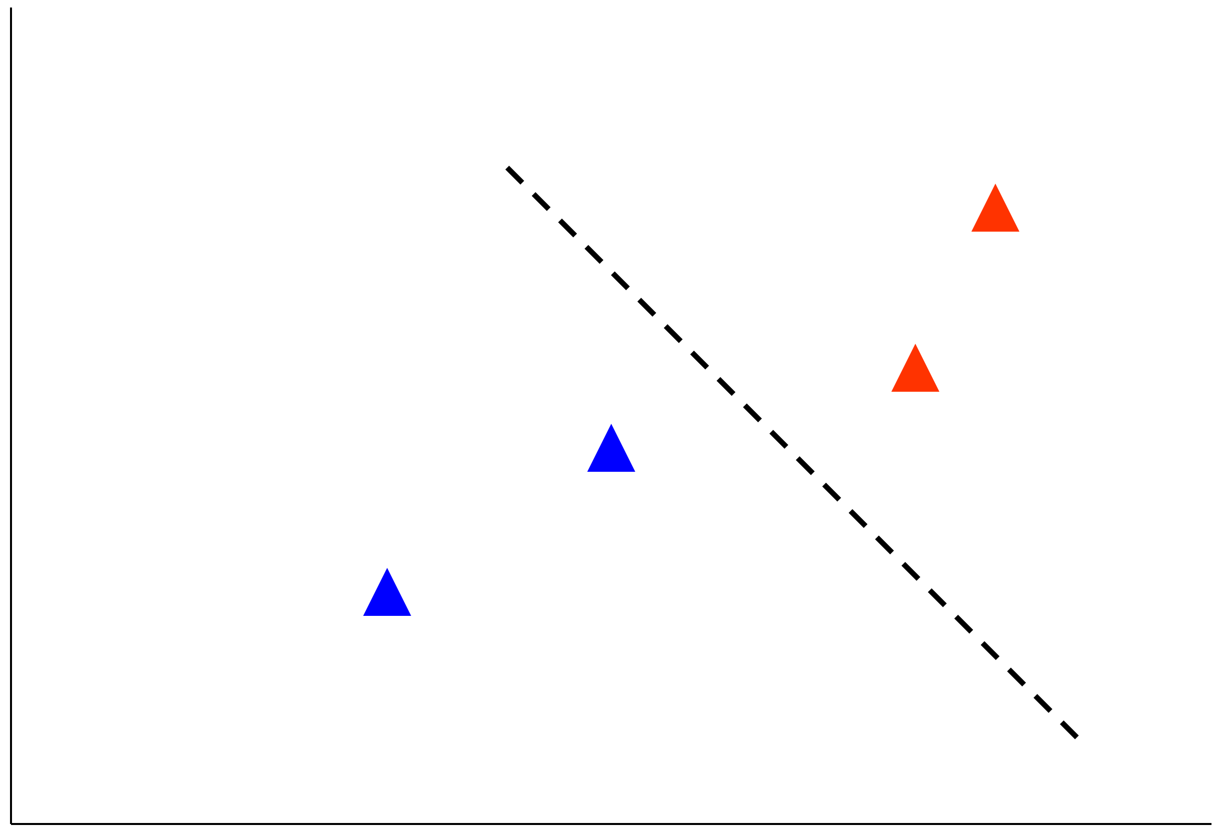
Four Ways to Use Unlabeled Data for Supervised Learning

1. Use to re-weight labeled examples
2. Use to help EM learn class-specific generative models
3. If problem has redundantly sufficient features, use CoTraining
4. Use to detect/preempt overfitting

1. Use unlabeled data to reweight labeled examples

- Most machine learning algorithms (neural nets, decision trees) attempt to *minimize errors over labeled examples*
- But our ultimate goal is to *minimize error over future examples* drawn from the same underlying distribution
- If we know the underlying distribution, we should weight each training example by its probability according to this distribution
- Unlabeled data allows us to estimate this distribution more accurately, and to reweight our labeled examples accordingly

Example



1. reweight labeled examples

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

- Wish to find

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

- Often approximate as

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

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

1. 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

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

# of times we have x in the labeled set

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

1. 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

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

# of times we have x in the labeled set

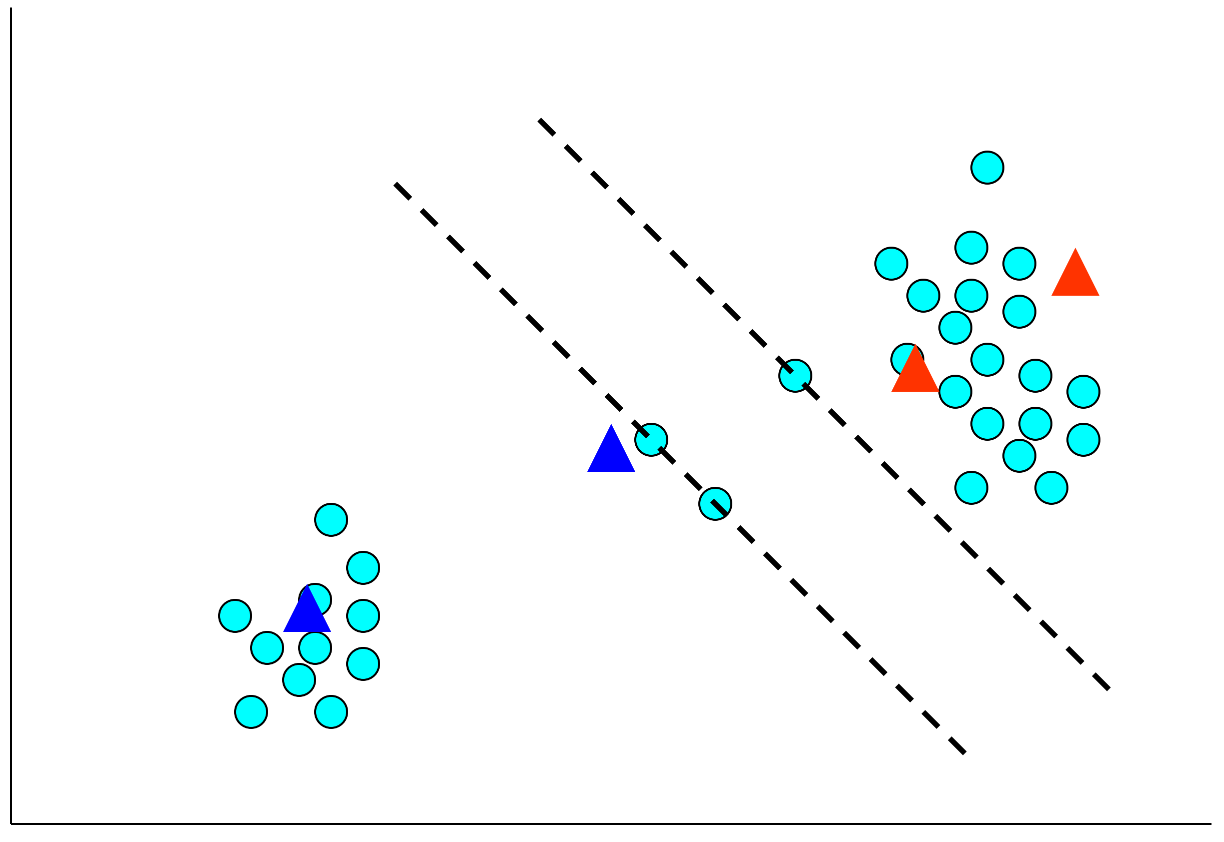
$$\hat{f} \leftarrow \underset{h \in H}{\operatorname{argmin}} \sum_{x \in X} \delta(h(x) \neq f(x)) \frac{n(x, L)}{|L|}$$

- Can use U for improved approximation:

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

# of times we have x in the unlabeled set

Example



2. Improve EM clustering algorithms

- Consider completely unsupervised clustering, where we assume data X is generated by a mixture of probability distributions, one for each cluster
  - For example, Gaussian mixtures
- Some classifier learning algorithms such as Gaussian Bayes classifiers also assumes the data X is generated by a mixture of distributions, one for each class Y
- Supervised learning: estimate P(X|Y) from labeled data
- Opportunity: estimate P(X|Y) from labeled and unlabeled data, using EM as in clustering

Bag of Words Text Classification

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

Baseline: Naïve Bayes Learner

Train:

For each class c_j of documents

1. Estimate $P(c_j)$

2. For each word w_i estimate $P(w_i \mid c_j)$

Classify (doc):

Assign *doc* to most probable class

$$\arg\max_j P(c_j) \prod_{w_i \in doc} P(w_i \mid c_j)$$

Naïve Bayes assumption: words are conditionally independent, given class

Faculty

associate	0.00417
chair	0.00303
member	0.00288
ph	0.00287
director	0.00282
fax	0.00279
journal	0.00271
recent	0.00260
received	0.00258
award	0.00250

Students

resume	0.00516
advisor	0.00456
student	0.00387
working	0.00361
stuff	0.00359
links	0.00355
homepage	0.00345
interests	0.00332
personal	0.00332
favorite	0.00310

Courses

homework	0.00413
syllabus	0.00399
assignments	0.00388
exam	0.00385
grading	0.00381
midterm	0.00374
pm	0.00371
instructor	0.00370
due	0.00364
final	0.00355

Departments

departmental	0.01246
colloquia	0.01076
epartment	0.01045
seminars	0.00997
schedules	0.00879
webmaster	0.00879
events	0.00826
facilities	0.00807
eople	0.00772
postgraduate	0.00764

Research Projects

investigators	0.00256
group	0.00250
members	0.00242
researchers	0.00241
laboratory	0.00238
develop	0.00201
related	0.00200
arpa	0.00187
affiliated	0.00184
project	0.00183

Others

type	0.00164
jan	0.00148
enter	0.00145
random	0.00142
program	0.00136
net	0.00128
time	0.00128
format	0.00124
access	0.00117
begin	0.00116

Expectation Maximization (EM) Algorithm

- Use labeled data *L* to learn initial classifier *h*

Loop:

- E Step:
  - Assign probabilistic labels to *U*, based on *h*
- M Step:
  - Retrain classifier *h* using both *L* (with fixed membership) and assigned labels to *U* (soft membership)

- Under certain conditions, guaranteed to converge to locally maximum likelihood *h*

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.

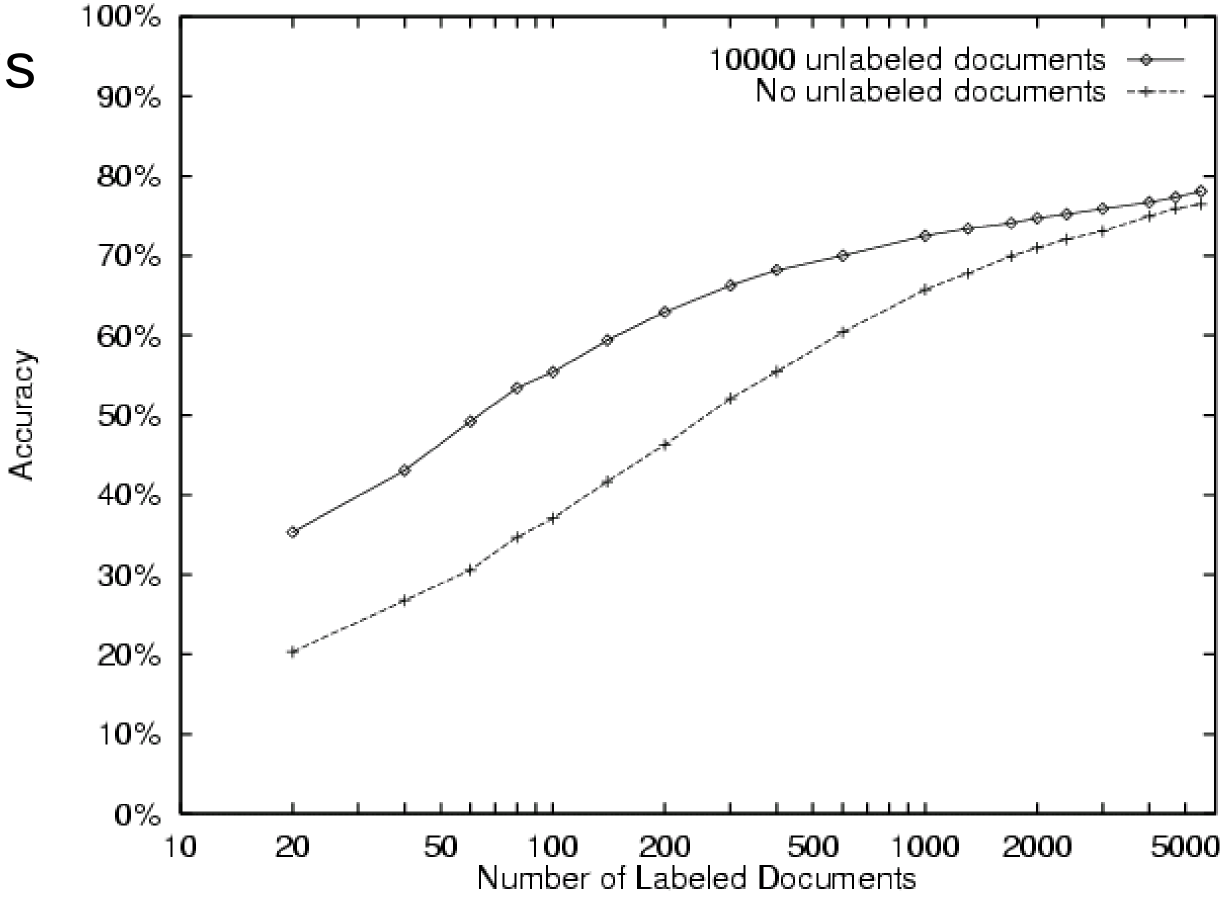
Iteration 0	Iteration 1	Iteration 2
intelligence	DD	D
DD	D	DD
artificial	lecture	lecture
understanding	cc	cc
DDw	$D^\star$	$DD{:}DD$
dist	$DD{:}DD$	due
identical	handout	$D^\star$
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	DDam	hw
cognitive	yurttas	exam
logic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

Newsgrop postings

– 20 newsgroups, 1000/group



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

- In some settings, available data features are so redundant that we can train two classifiers using different 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 training of both classifiers, forcing them to agree

CoTraining

$$learn \quad f : X \rightarrow Y$$

$$where \quad X = X_1 \times X_2$$

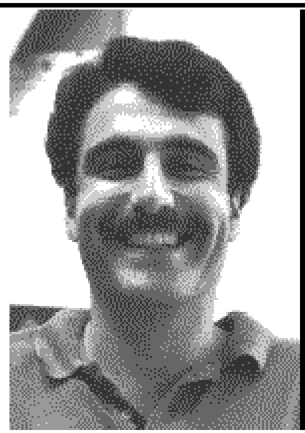
$$where \quad x \quad drawn \quad from \quad unknown \quad distribution$$

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

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

[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

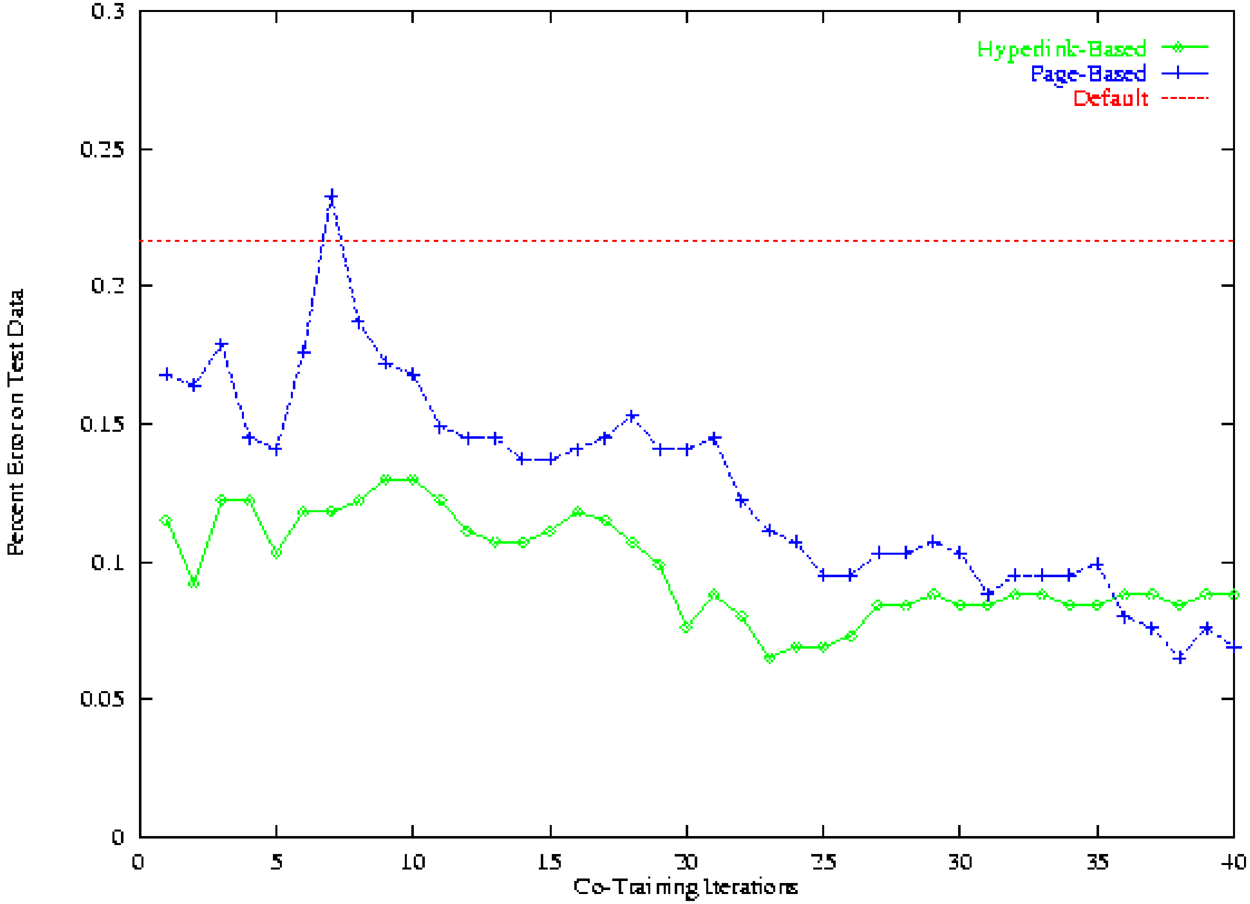
Allow g2 to label *p* positive, *n* negative examps from U

Add the intersection of the self-labeled examples to L

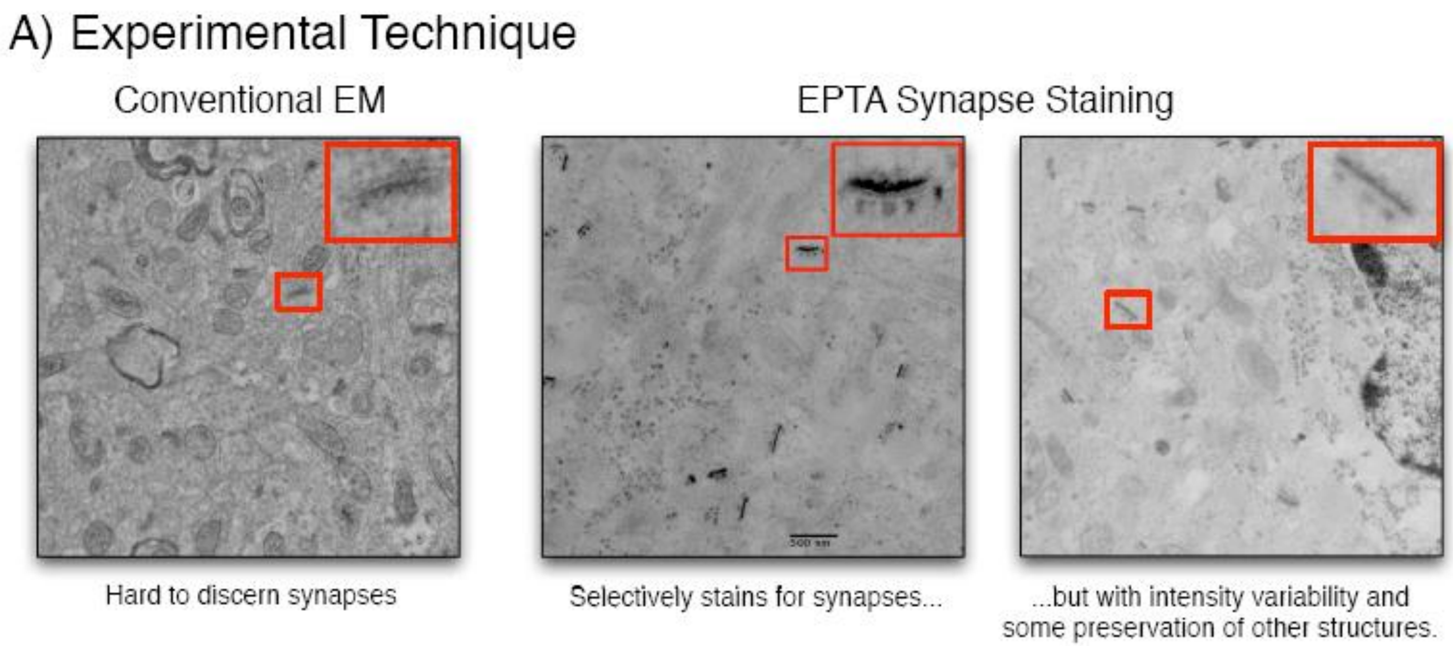
CoTraining: Experimental Results

- 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% (when both agree)

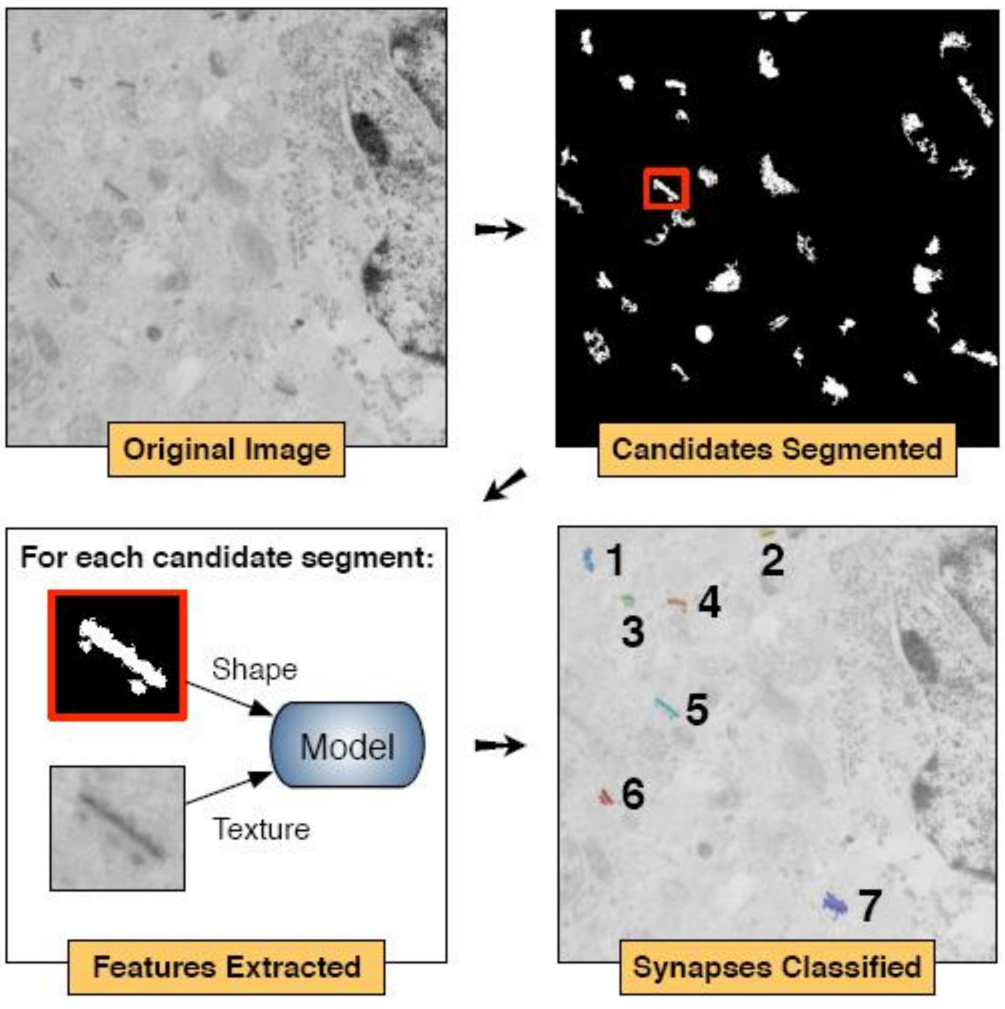
Typical run:



Classifying images: Neural networks



Co-Training



Original Image
Candidates Segmented
For each candidate segment:
Shape
Texture
Model
Features Extracted
1
2
3
4
5
6
7
Synapses Classified

Accuracy

		—Accuracy—		—AUC—	
Training / Test	**Co-training**	**Positive**	**Negative**	**Prec-Recall**	**ROC**
Train P75 / Test P14	No	66.36%	98.20%	73.65%	96.91%
Train P75 / Test P14	Yes (0.5%)	72.90%	**98.60%**	**75.75%**	**97.14%**
Train P75 / Test P14	Yes (1.5%)	**74.77%**	96.91%	73.06%	96.65%
Train P14 / Test P75	No	48.78%	**98.96%**	60.50%	90.38%
Train P14 / Test P75	Yes (0.5%)	60.16%	98.21%	**64.23%**	**92.89%**
Train P14 / Test P75	Yes (1.5%)	**60.98%**	97.55%	63.80%	92.83%

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

Defining a distance metric

- Definition of distance metric
  - Non-negative *d(f,g)≥0;*
  - symmetric *d(f,g)=d(g,f)*;
  - triangle inequality *d(f,g) · 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:

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

Using the distance metric

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

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

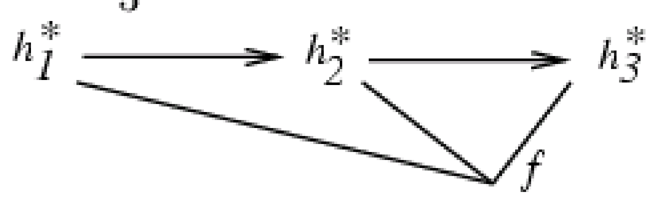
$$\hat{d}(h_1, f) = \frac{1}{|L|} \sum_{x_i \in L} \delta(h_1(x_i) \neq y_i)$$

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

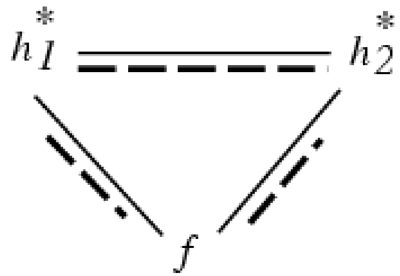
Organize H into complexity classes

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

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



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!

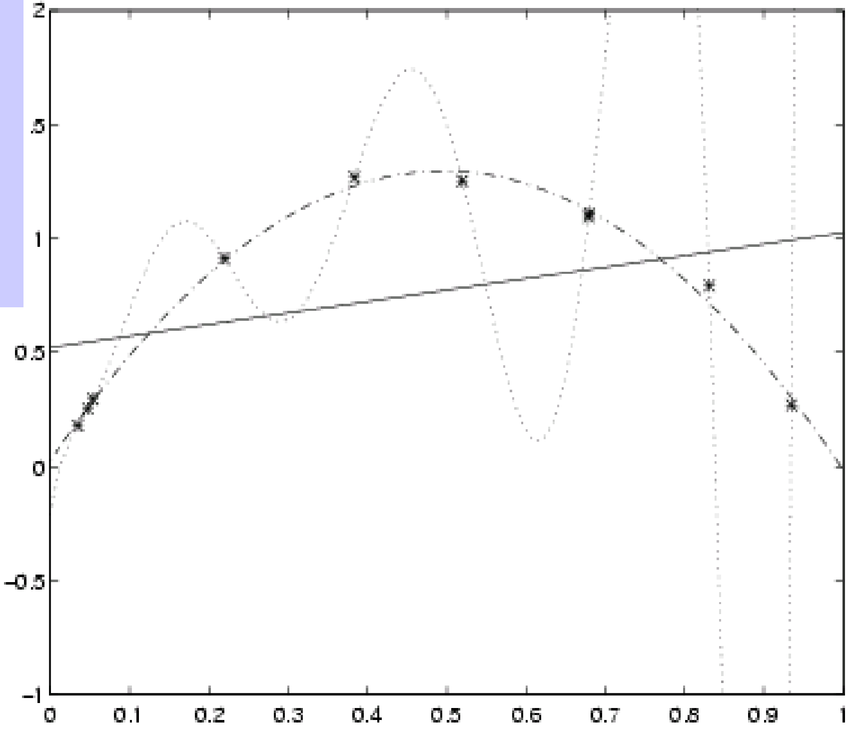
$$d(h_1, h_2) \leq d(h_1, f) + d(f, h_2)$$

→ Heuristic:

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

Generated y values contain zero mean Gaussian noise ε

$Y=f(x)+\varepsilon$



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.

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, …

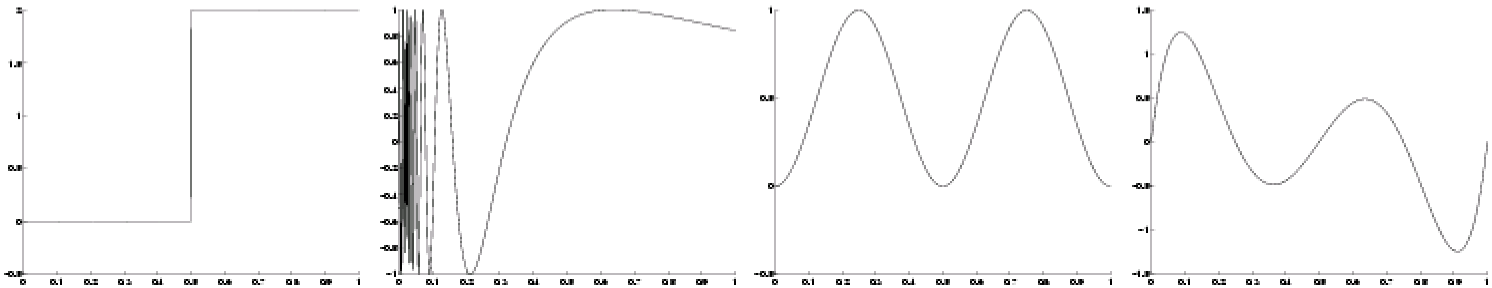


Figure 5: Target functions used in the polynomial curve fitting experiments (in order): step($x \geq 0.5$), $\sin(1/x)$, $\sin^2(2\pi x)$, and a fifth degree polynomial.

Summary

Several ways to use unlabeled data in supervised learning

1. Use to reweight labeled examples
2. Use to help EM learn class-specific generative models
3. If problem has redundantly sufficient features, use CoTraining
4. Use to detect/preempt overfitting

Ongoing research area

Acknowledgment

Some of these slides are based in on slides from Tom Mitchell.