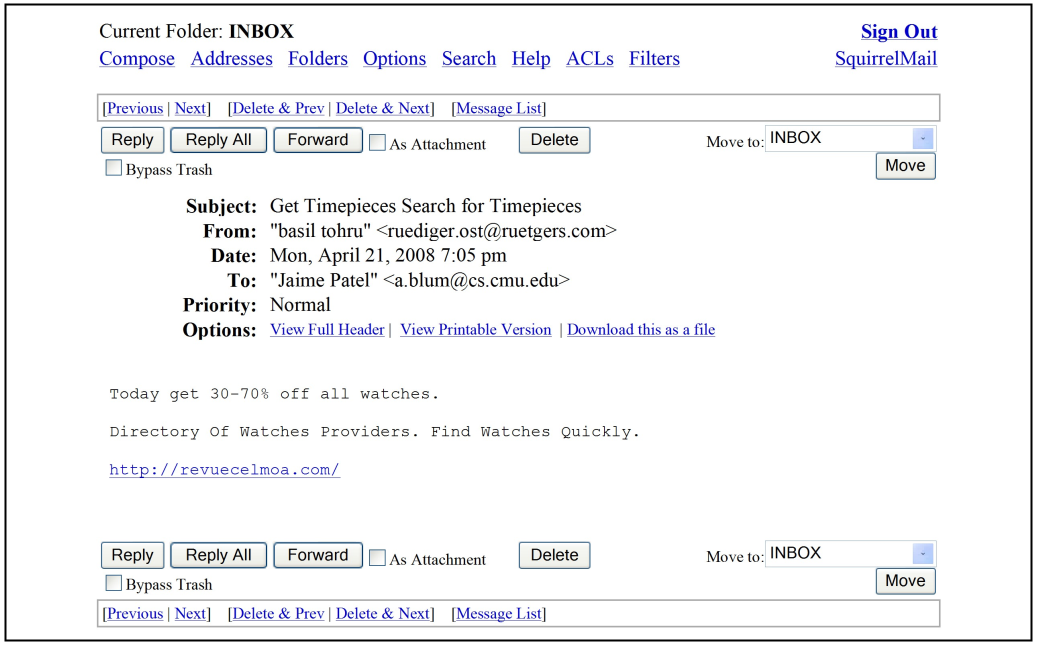
Semi-Supervised Learning

~~Semi-~~Supervised Learning

Supervised Learning = learning from labeled data. Dominant paradigm in Machine Learning.

- E.g, say you want to train an email classifier to distinguish spam from important messages



Current Folder: **INBOX** **Sign Out**
Compose Addresses Folders Options Search Help ACLs Filters SquirrelMail

[Previous | Next] [Delete & Prev | Delete & Next] [Message List]

Reply | Reply All | Forward | As Attachment | Delete | Move to: INBOX | Move
Bypass Trash

Subject: Get Timepieces Search for Timepieces
From: "basil tohru" <ruediger.ost@ruetgers.com>
Date: Mon, April 21, 2008 7:05 pm
To: "Jaime Patel" <a.blum@cs.cmu.edu>
Priority: Normal
Options: View Full Header | View Printable Version | Download this as a file

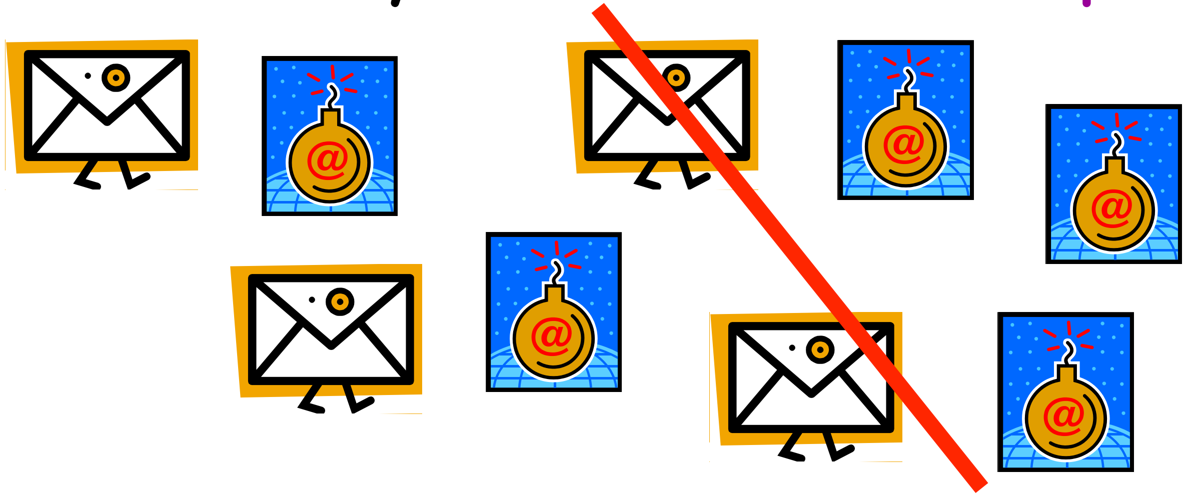
```
Today get 30-70% off all watches.

Directory Of Watches Providers. Find Watches Quickly.

http://revuecelmoa.com/
```

Reply | Reply All | Forward | As Attachment | Delete | Move to: INBOX | Move
Bypass Trash
[Previous | Next] [Delete & Prev | Delete & Next] [Message List]

~~Semi-~~Supervised Learning

Supervised Learning = learning from labeled data. Dominant paradigm in Machine Learning.

- E.g, say you want to train an email classifier to distinguish spam from important messages
- Take sample S of data, labeled according to whether they were/weren't spam.



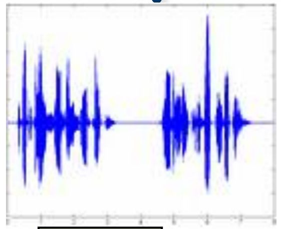
~~Semi~~-Supervised Learning

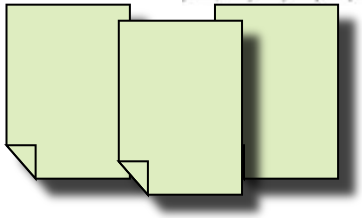
Supervised Learning = learning from labeled data. Dominant paradigm in Machine Learning.

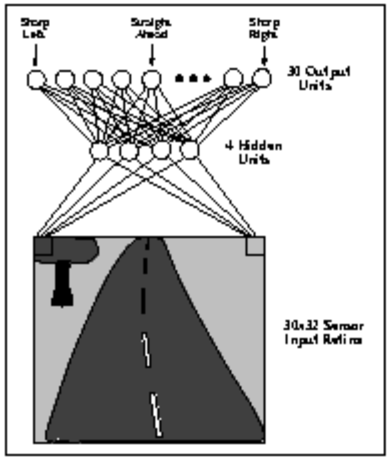
- E.g, say you want to train an email classifier to distinguish spam from important messages
- Take sample S of data, labeled according to whether they were/weren't spam.
- Train a classifier (like SVM, decision tree, etc) on S. Make sure it's not overfitting.
- Use to classify new emails.

Basic paradigm has many successes

- recognize speech,
- steer a car,
- classify documents
- classify proteins
- recognizing faces, objects in images
- ...







However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.

However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

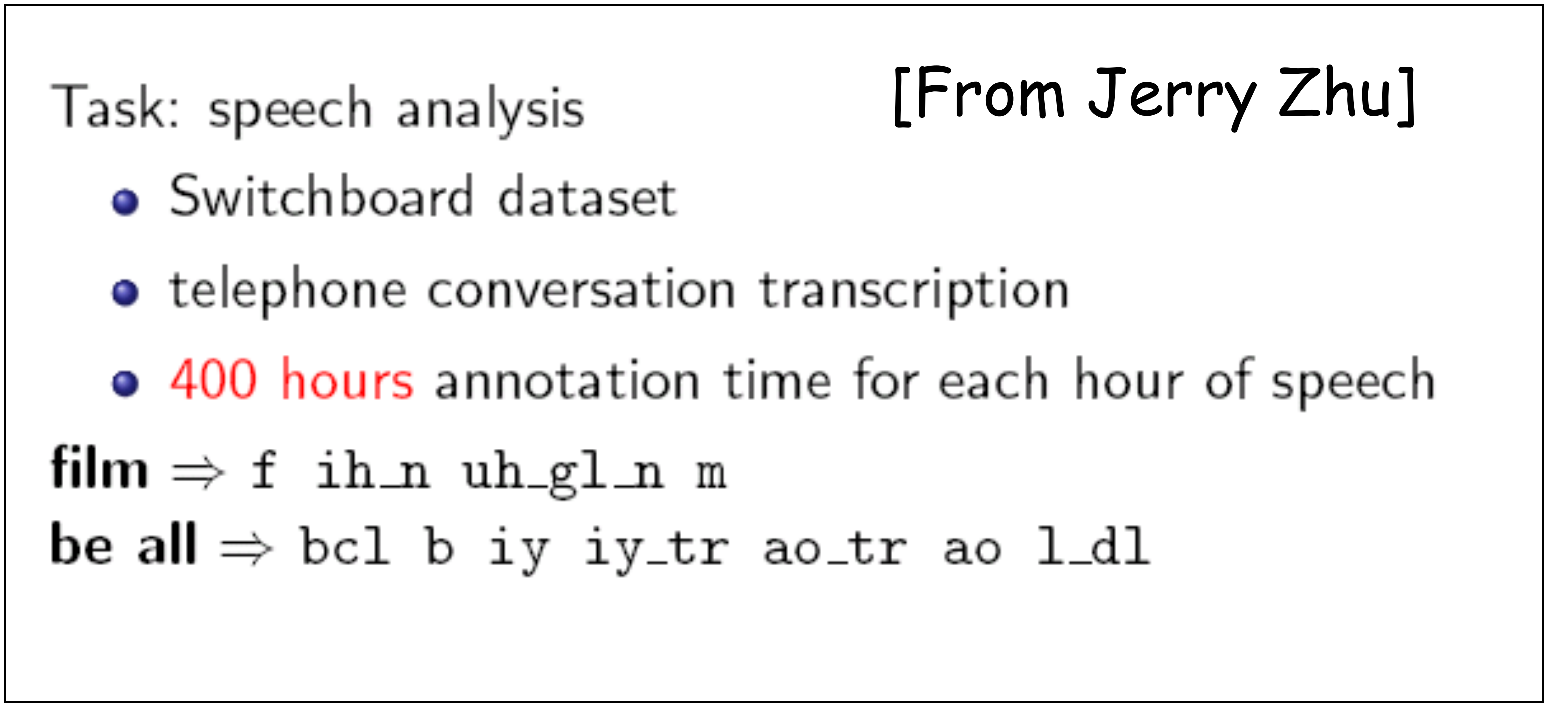
Unlabeled data is much cheaper.

Speech	Customer modeling
Images	Protein sequences
Medical outcomes	Web pages

However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.



[From Jerry Zhu]

Task: speech analysis

- Switchboard dataset
- telephone conversation transcription
- 400 hours annotation time for each hour of speech

film ⇒ f ih_n uh_gl_n m

be all ⇒ bcl b iy iy_tr ao_tr ao l_dl

However, for many problems, labeled data can be rare or expensive.

Need to pay someone to do it, requires special testing,…

Unlabeled data is much cheaper.

Can we make use of cheap unlabeled data?

Semi-Supervised Learning

Can we use unlabeled data to augment a small labeled sample to improve learning?

- But unlabeled data is missing the most important info!!
- But maybe still has useful regularities that we can use.
- But, but, but

Semi-Supervised Learning

Can we use unlabeled data to augment a small labeled sample to improve learning?

But unlabeled data is missing the most important info!!

But maybe still has useful regularities that we can use.

Bu Bu But…

Semi-Supervised Learning

Substantial recent work in ML. A number of interesting methods have been developed.

This talk:

- Discuss several diverse methods for taking advantage of unlabeled data.

Method 1:

Expectation-Maximization

How to use unlabeled data

- One way is to use the EM algorithm
  - EM: Expectation Maximization
- The EM algorithm is a popular iterative algorithm for maximum likelihood estimation in problems with missing data.
- The EM algorithm consists of two steps,
  - *Expectation step*, i.e., filling in the missing data
  - *Maximization step* – calculate a new maximum *a posteriori* estimate for the parameters.

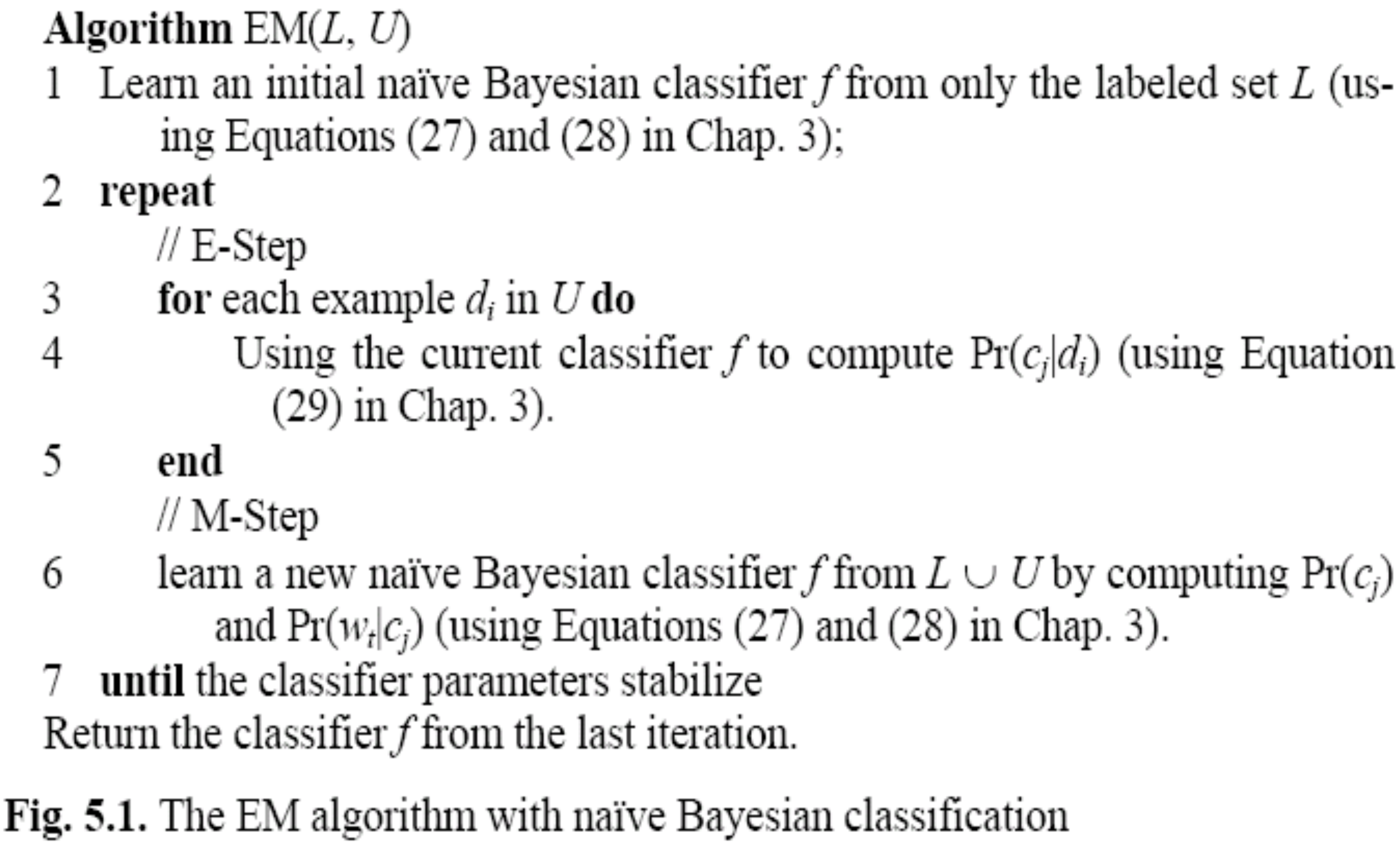
Incorporating unlabeled Data with EM (Nigam et al, 2000)

- Basic EM
- Augmented EM with weighted unlabeled data
- Augmented EM with multiple mixture components per class

Algorithm Outline

1. Train a classifier with only the labeled documents.
2. Use it to probabilistically classify the unlabeled documents.
3. Use ALL the documents to train a new classifier.
4. Iterate steps 2 and 3 to convergence.

Basic Algorithm



```
Algorithm EM($L$, $U$)
1   Learn an initial naïve Bayesian classifier $f$ from only the labeled set $L$ (us-
        ing Equations (27) and (28) in Chap. 3);
2   repeat
        // E-Step
3       for each example $d_i$ in $U$ do
4           Using the current classifier $f$ to compute $\Pr(c_j|d_i)$ (using Equation
              (29) in Chap. 3).
5       end
        // M-Step
6       learn a new naïve Bayesian classifier $f$ from $L \cup U$ by computing $\Pr(c_j)$
          and $\Pr(w_t|c_j)$ (using Equations (27) and (28) in Chap. 3).
7   until the classifier parameters stabilize
Return the classifier $f$ from the last iteration.
```

Fig. 5.1. The EM algorithm with naïve Bayesian classification

Basic EM: E Step & M Step

E Step:

$$\Pr(c_j \mid d_i; \hat{\Theta}) = \frac{\Pr(c_j \mid \hat{\Theta}) \Pr(d_i \mid c_j; \hat{\Theta})}{\Pr(d_i \mid \hat{\Theta})} \qquad (29)$$

$$= \frac{\Pr(c_j \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j; \hat{\Theta})}{\sum_{r=1}^{|C|} \Pr(c_r \mid \hat{\Theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r; \hat{\Theta})},$$

M Step:

$$\Pr(w_t \mid c_j; \hat{\Theta}) = \frac{\lambda + \sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}. \qquad (27)$$

$$\Pr(c_j \mid \hat{\Theta}) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|}. \qquad (28)$$

The problem

- It has been shown that the EM algorithm in Fig. 5.1 works well if the
  - The two mixture model assumptions for a particular data set are true.
- The two mixture model assumptions, however, can cause major problems when they do not hold. In many real-life situations, they may be violated.
- It is often the case that a class (or topic) contains a number of sub-classes (or sub-topics).
  - For example, the class Sports may contain documents about different sub-classes of sports, Baseball, Basketball, Tennis, and Softball.
- Some methods to deal with the problem.

Weighting the influence of unlabeled examples by factor μ

New M step:

$$\Pr(w_t \mid c_j) = \frac{\lambda + \sum_{i=1}^{|D|} \Lambda(i) N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} \Lambda(i) N_{ti} \Pr(c_j \mid d_i)}, \qquad (1)$$

where

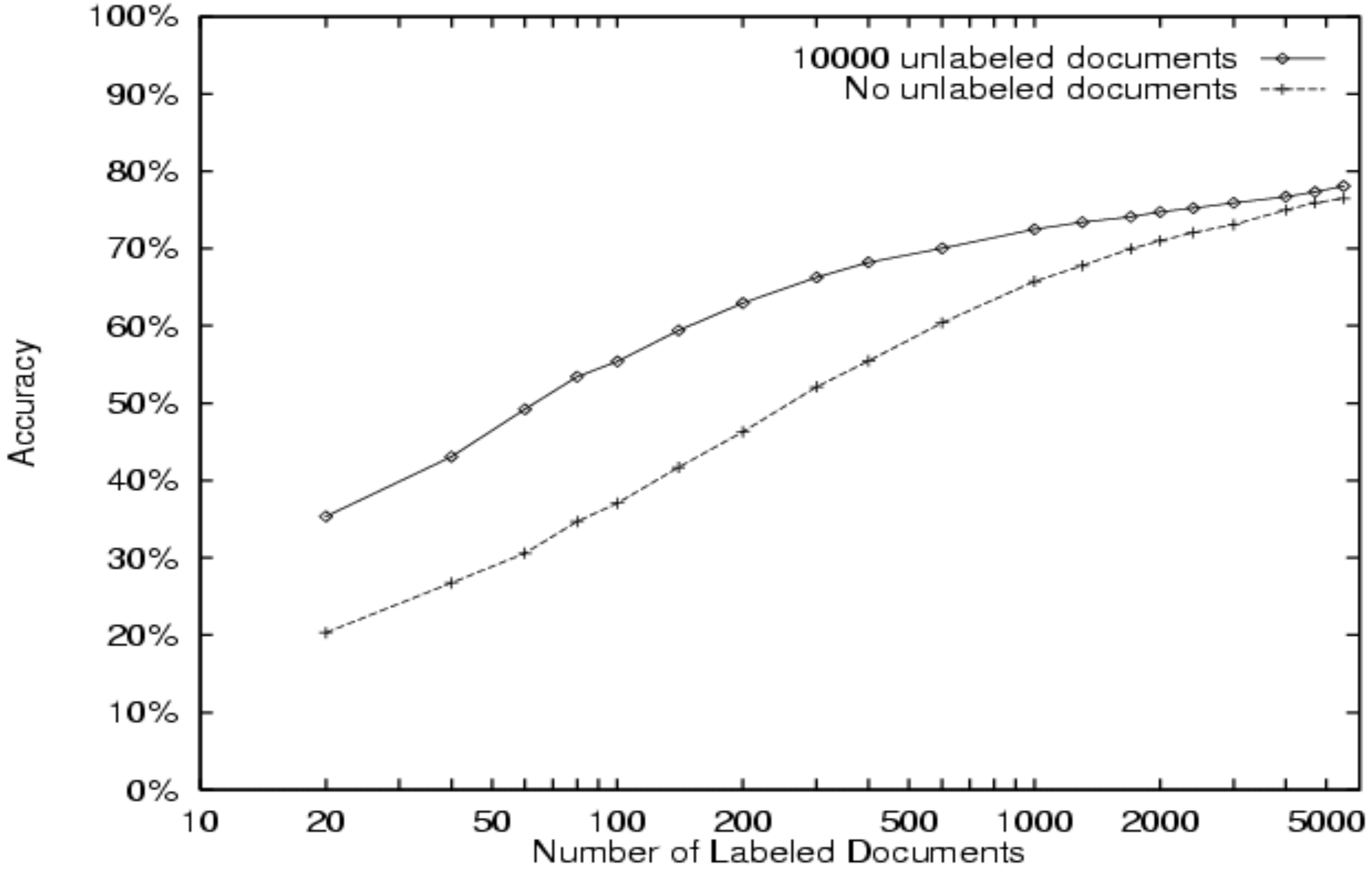
$$\Lambda(i) = \begin{cases} \mu & \text{if } d_i \in U \\ 1 & \text{if } d_i \in L. \end{cases} \qquad (2)$$

The prior probability also needs to be weighted.

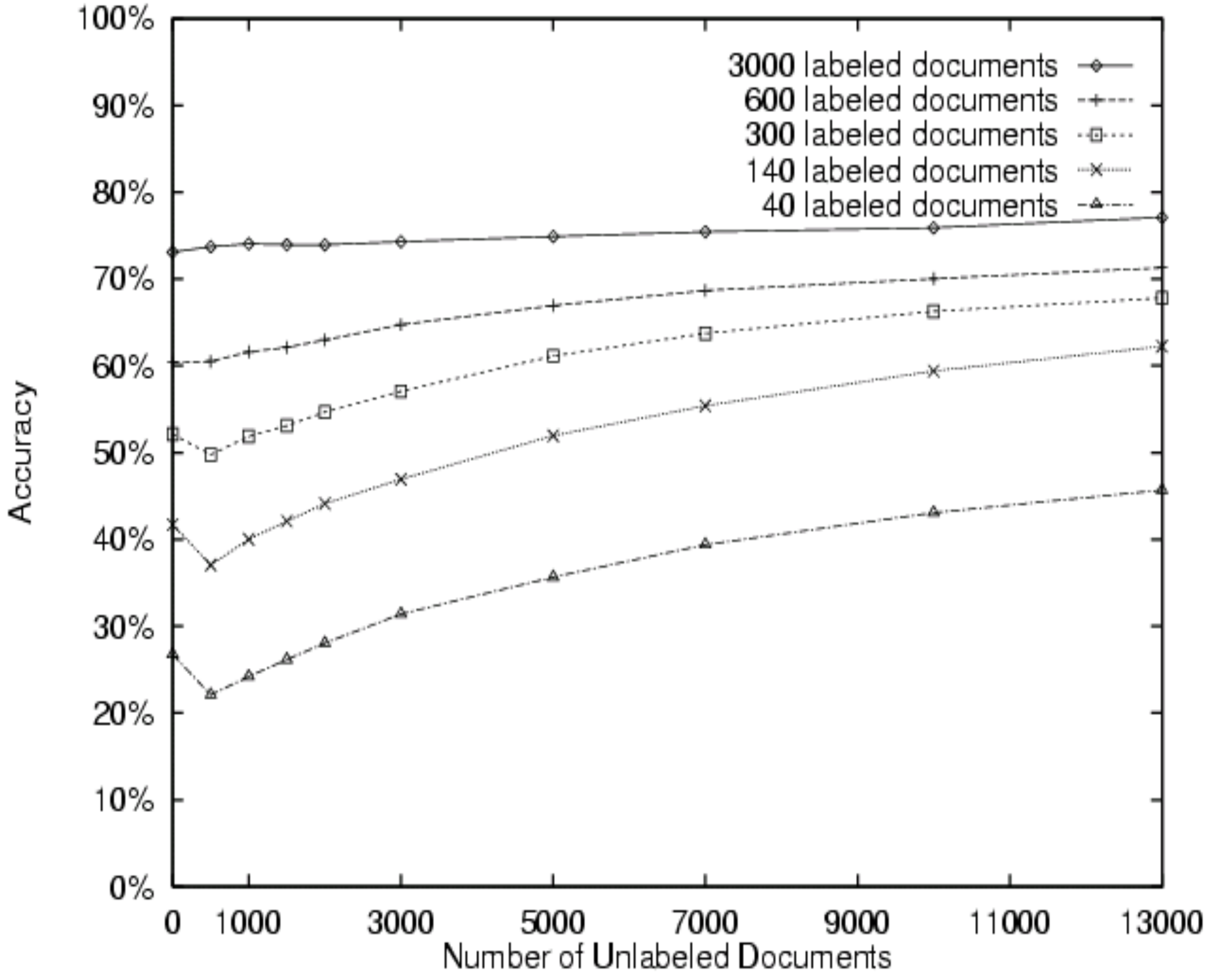
Experimental Evaluation

- Newsgroup postings
  - 20 newsgroups, 1000/group
- Web page classification
  - student, faculty, course, project
  - 4199 web pages
- Reuters newswire articles
  - 12,902 articles
  - 10 main topic categories

20 Newsgroups



20 N



100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Accuracy
3000 labeled documents
600 labeled documents
300 labeled documents
140 labeled documents
40 labeled documents
0
1000
3000
5000
7000
9000
11000
13000
Number of Unlabeled Documents

Method 2:

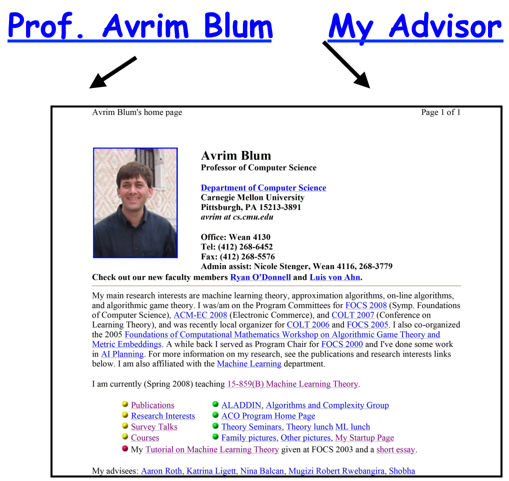
Co-Training

Co-training

[Blum&Mitchell' 98]

Many problems have two different sources of info you can use to determine label.

E.g., classifying webpages: can use words on page or words on links pointing to the page.



Avrim Blum's home page Page 1 of 1

Avrim Blum
Professor of Computer Science

Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
avrim at cs.cmu.edu

Office: Wean 4130
Tel: (412) 268-6452
Fax: (412) 268-5576
Admin assist: Nicole Stenger, Wean 4116, 268-3779
Check out our new faculty members Ryan O'Donnell and Luis von Ahn.

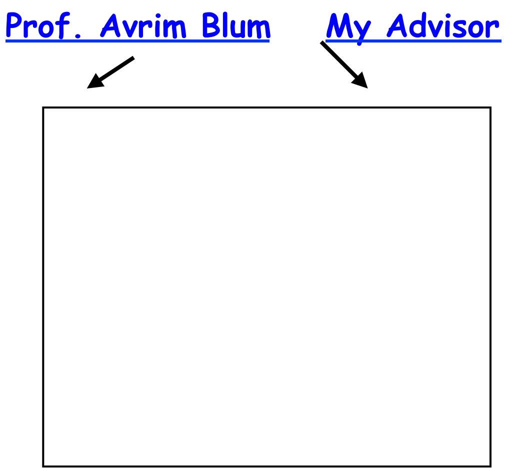
My main research interests are machine learning theory, approximation algorithms, on-line algorithms, and algorithmic game theory. I was/am on the Program Committees for FOCS 2008 (Symp. Foundations of Computer Science), ACM-EC 2008 (Electronic Commerce), and COLT 2007 (Conference on Learning Theory), and was recently local organizer for COLT 2006 and FOCS 2005. I also co-organized the 2005 Foundations of Computational Mathematics Workshop on Algorithmic Game Theory and Metric Embeddings. A while back I served as Program Chair for FOCS 2000 and I've done some work in AI Planning. For more information on my research, see the publications and research interests links below. I am also affiliated with the Machine Learning department.

I am currently (Spring 2008) teaching 15-859(B) Machine Learning Theory.

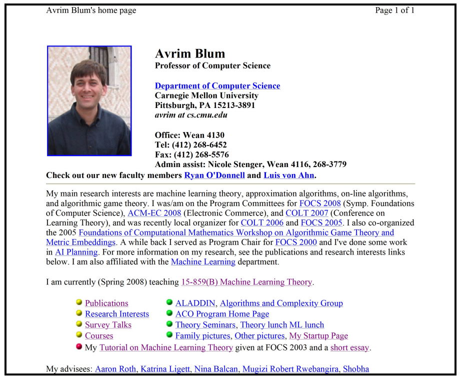
- Publications
- Research Interests
- Survey Talks
- Courses
- ALADDIN, Algorithms and Complexity Group
- ACO Program Home Page
- Theory Seminars, Theory lunch ML lunch
- Family pictures, Other pictures, My Startup Page
- My Tutorial on Machine Learning Theory given at FOCS 2003 and a short essay.

My advisees: Aaron Roth, Katrina Ligett, Nina Balcan, Mugizi Robert Rwebangira, Shobha

x - Link info & Text info



x_1- Link info


Avrim Blum's home page Page 1 of 1

Avrim Blum
Professor of Computer Science

Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
avrim at cs.cmu.edu

Office: Wean 4130
Tel: (412) 268-6452
Fax: (412) 268-5576
Admin assist: Nicole Stenger, Wean 4116, 268-3779
Check out our new faculty members Ryan O'Donnell and Luis von Ahn.

My main research interests are machine learning theory, approximation algorithms, on-line algorithms, and algorithmic game theory. I was/am on the Program Committees for FOCS 2008 (Symp. Foundations of Computer Science), ACM-EC 2008 (Electronic Commerce), and COLT 2007 (Conference on Learning Theory), and was recently local organizer for COLT 2006 and FOCS 2005. I also co-organized the 2005 Foundations of Computational Mathematics Workshop on Algorithmic Game Theory and Metric Embeddings. A while back I served as Program Chair for FOCS 2000 and I've done some work in AI Planning. For more information on my research, see the publications and research interests links below. I am also affiliated with the Machine Learning department.

I am currently (Spring 2008) teaching 15-859(B) Machine Learning Theory.

- Publications
- Research Interests
- Survey Talks
- Courses
- ALADDIN, Algorithms and Complexity Group
- ACO Program Home Page
- Theory Seminars, Theory lunch ML lunch
- Family pictures, Other pictures, My Startup Page
- My Tutorial on Machine Learning Theory given at FOCS 2003 and a short essay.

My advisees: Aaron Roth, Katrina Ligett, Nina Balcan, Mugizi Robert Rwebangira, Shobha

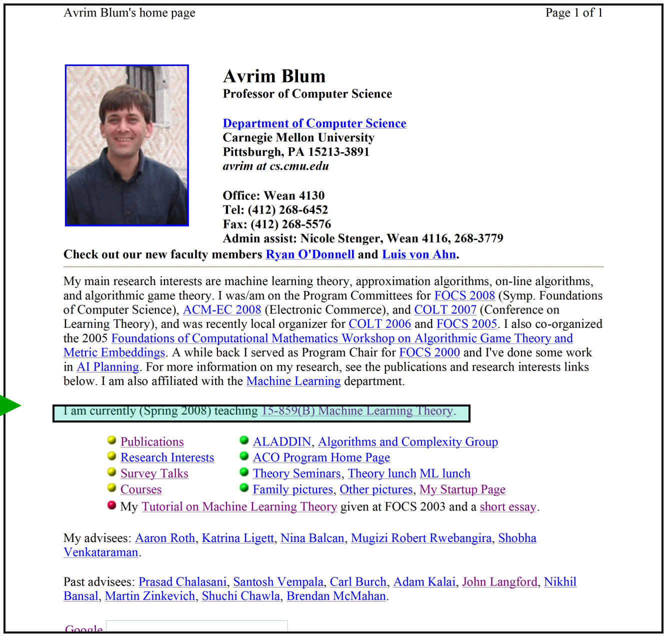
x_2- Text info

Co-training

Idea: Use small labeled sample to learn initial rules.

- E.g., “my advisor” pointing to a page is a good indicator it is a faculty home page.
- E.g., “I am teaching” on a page is a good indicator it is a faculty home page.

my advisor



Avrim Blum's home page — Page 1 of 1

Avrim Blum
Professor of Computer Science

Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
avrim at cs.cmu.edu

Office: Wean 4130
Tel: (412) 268-6452
Fax: (412) 268-5576
Admin assist: Nicole Stenger, Wean 4116, 268-3779

Check out our new faculty members Ryan O'Donnell and Luis von Ahn.

My main research interests are machine learning theory, approximation algorithms, on-line algorithms, and algorithmic game theory. I was/am on the Program Committees for FOCS 2008 (Symp. Foundations of Computer Science), ACM-EC 2008 (Electronic Commerce), and COLT 2007 (Conference on Learning Theory), and was recently local organizer for COLT 2006 and FOCS 2005. I also co-organized the 2005 Foundations of Computational Mathematics Workshop on Algorithmic Game Theory and Metric Embeddings. A while back I served as Program Chair for FOCS 2000 and I've done some work in AI Planning. For more information on my research, see the publications and research interests links below. I am also affiliated with the Machine Learning department.

I am currently (Spring 2008) teaching 15-859(B) Machine Learning Theory.

- Publications
- Research Interests
- Survey Talks
- Courses
- ALADDIN, Algorithms and Complexity Group
- ACO Program Home Page
- Theory Seminars, Theory lunch ML lunch
- Family pictures, Other pictures, My Startup Page
- My Tutorial on Machine Learning Theory given at FOCS 2003 and a short essay.

My advisees: Aaron Roth, Katrina Ligett, Nina Balcan, Mugizi Robert Rwebangira, Shobha Venkataraman.

Past advisees: Prasad Chalasani, Santosh Vempala, Carl Burch, Adam Kalai, John Langford, Nikhil Bansal, Martin Zinkevich, Shuchi Chawla, Brendan McMahan.

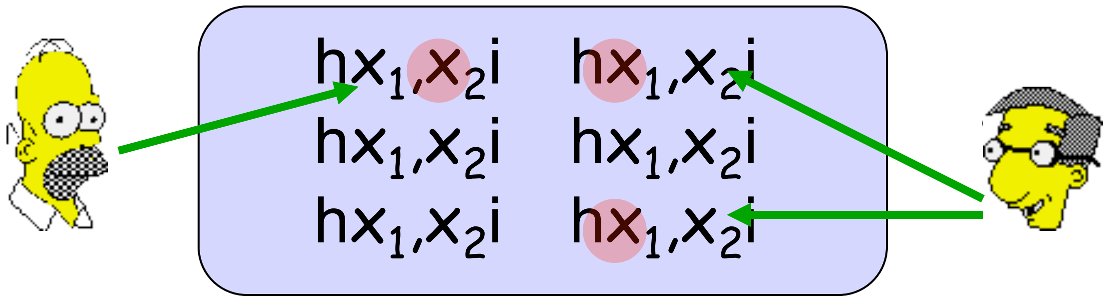
Google

Co-training

Idea: Use small labeled sample to learn initial rules.

- E.g., "my advisor" pointing to a page is a good indicator it is a faculty home page.
- E.g., "I am teaching" on a page is a good indicator it is a faculty home page.

Then look for unlabeled examples where one rule is confident and the other is not. Have it label the example for the other.

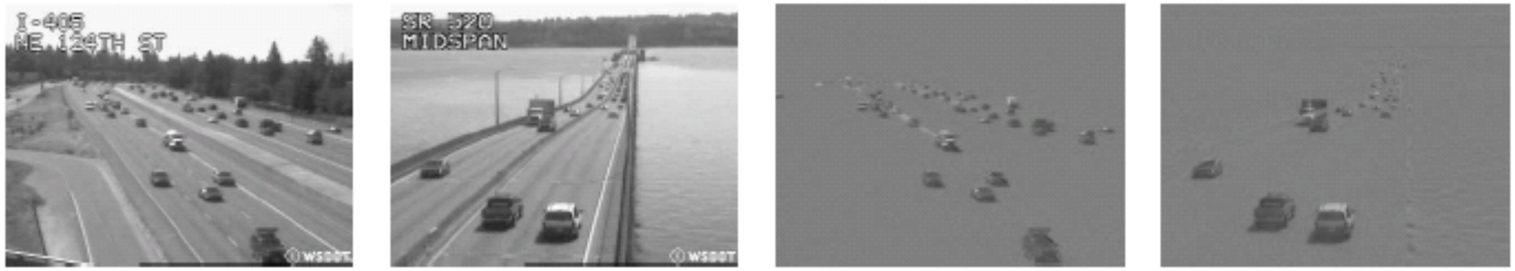


Training 2 classifiers, one on each type of info. Using each to help train the other.

Co-training

Turns out a number of problems can be set up this way.

E.g., [Levin-Viola-Freund03] identifying objects in images. Two different kinds of preprocessing.



E.g., [Collins&Singer99] named-entity extraction.

- "I arrived in London yesterday"

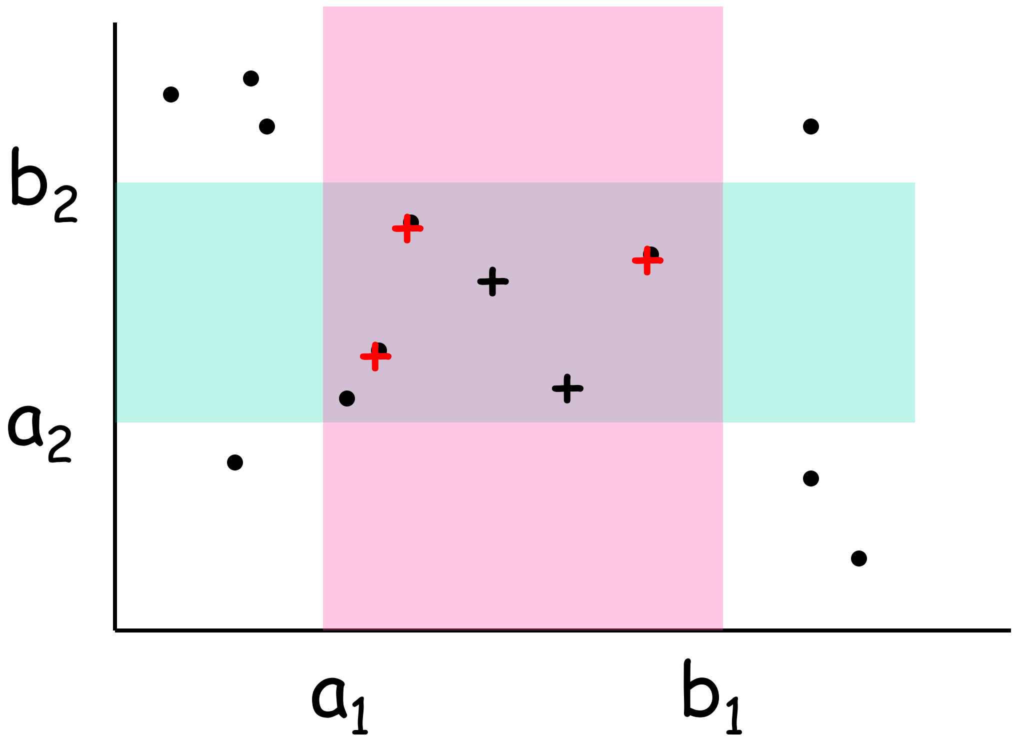
- ...

Co-training

- Setting is each example x = hx_1,x_2i, where x_1, x_2 are two “views” of the data.
- Have separate algorithms running on each view. Use each to help train the other.
- Basic hope is that two views are consistent. Using agreement as proxy for labeled data.

Toy example: intervals

As a simple example, suppose x_1, x_2 2 R. Target function is some interval [a,b].

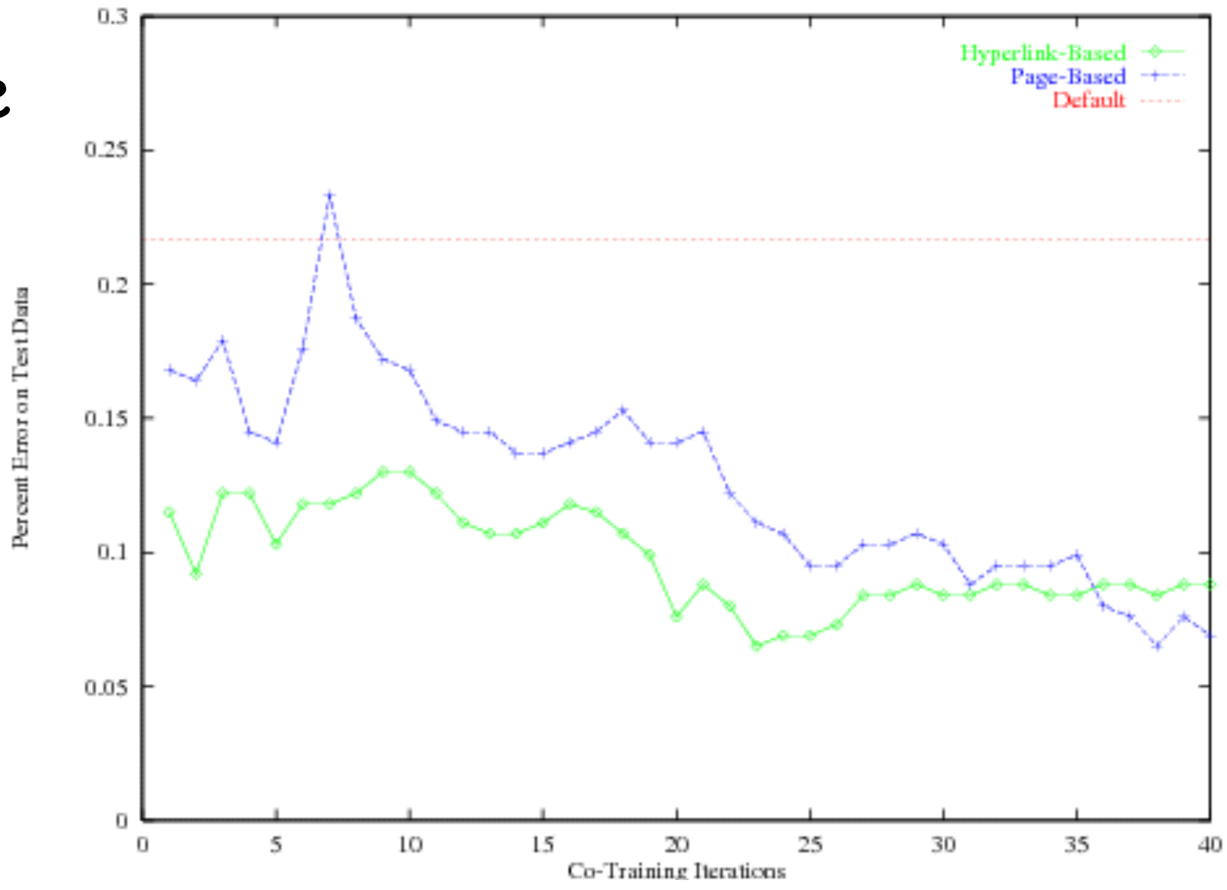


Results: webpages

12 labeled examples, 1000 unlabeled

	Page-based	Hyperlink-based	Combined
Std. Supervised	12.9	12.4	11.1
Co-training	6.2	11.6	5.0
Just say neg	22	22	22

(sample run)



Results: images [Levin-Viola-Freund '03]:

- Visual detectors with different kinds of processing

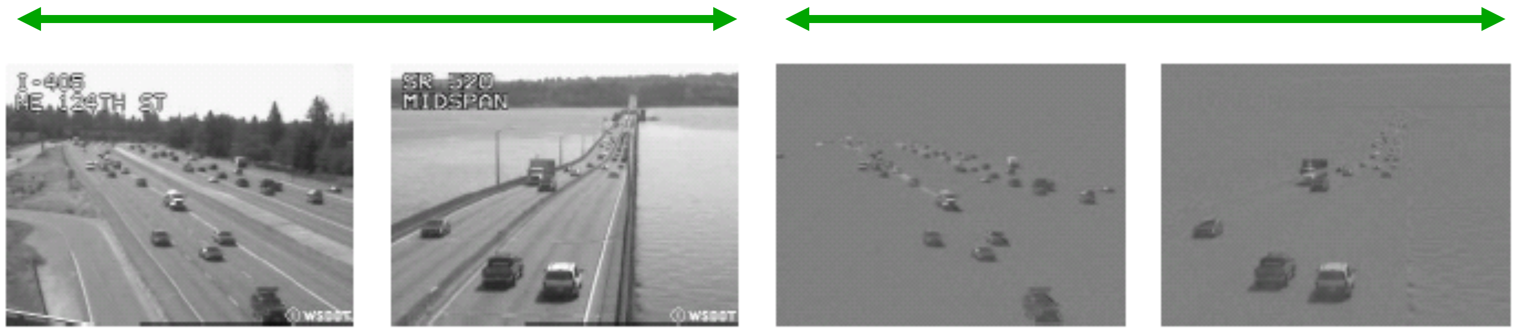
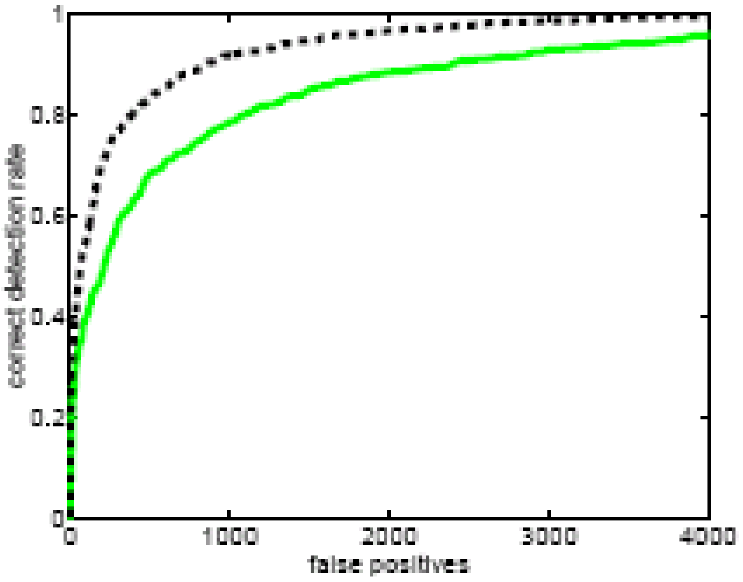


Figure 1: Example images used to test and train the car detection system. On the left are the original images. On the right are background subtracted images.

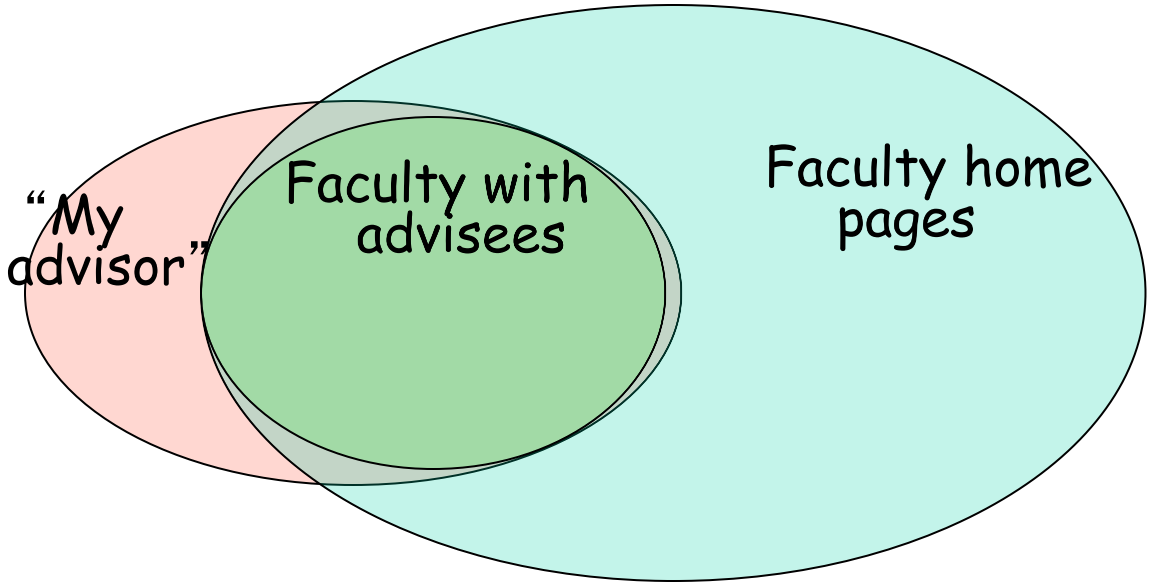
- Images with 50 labeled cars. 22,000 unlabeled images.
- Factor 2-3+ improvement.



From [LVF03]

Co-Training Theorems

- [BM98] if x_1, x_2 are independent given the label, and if have alg that is robust to noise, then can learn from an initial "weakly-useful" rule plus unlabeled data.



Co-Training Theorems

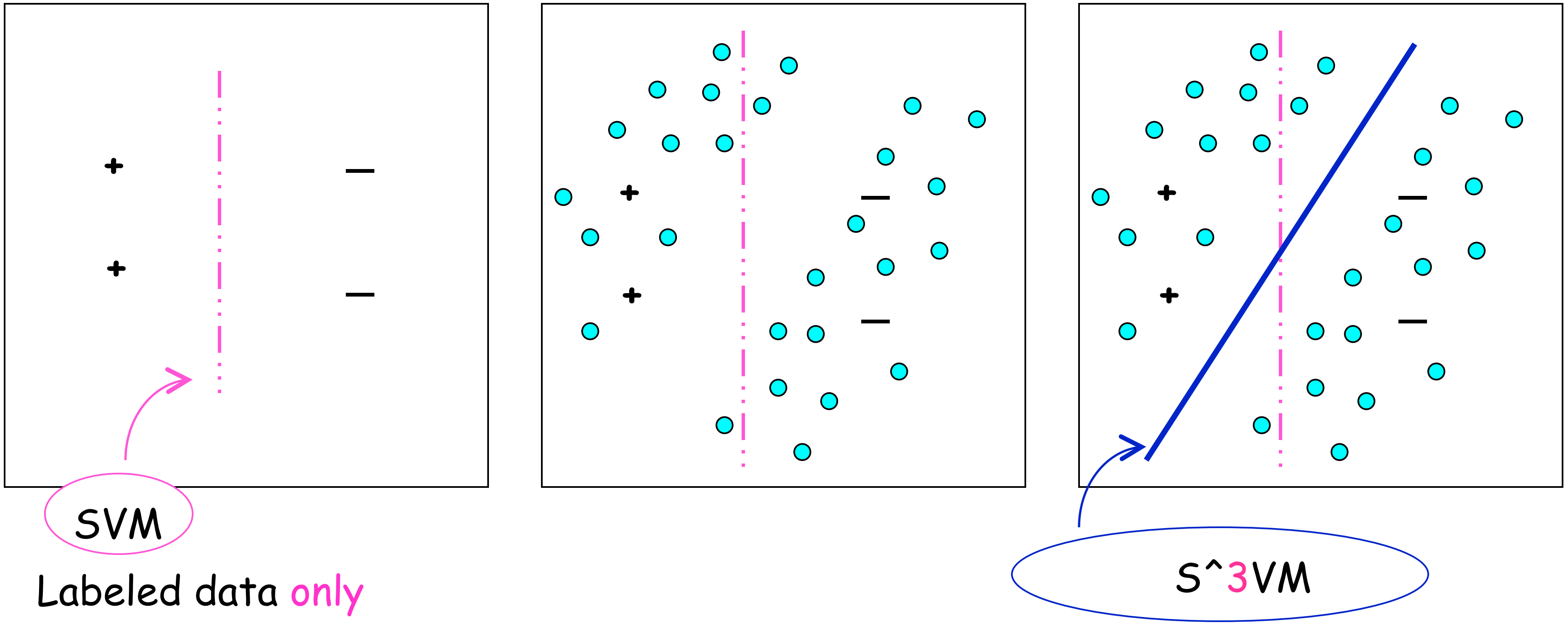
- [BM98] if x_1, x_2 are independent given the label, and if have alg that is robust to noise, then can learn from an initial "weakly-useful" rule plus unlabeled data.
- [BB05] in some cases (e.g., LTFs), you can use this to learn from a single labeled example!
- [BBY04] if algs are correct when they are confident, then suffices for distrib to have expansion.
- …

Method 2:

Semi-Supervised (Transductive) SVM

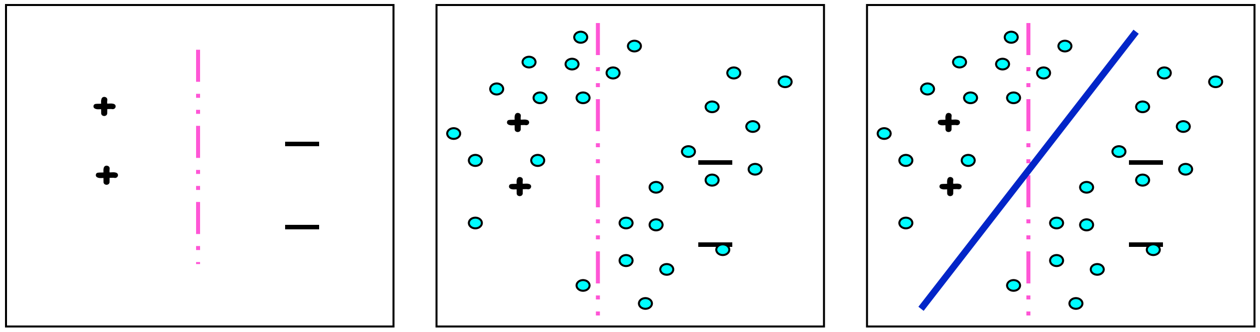
S^3VM [Joachims98]

- Suppose we believe target separator goes through low density regions of the space/large margin.
- Aim for separator with large margin wrt labeled and unlabeled data. (L+U)



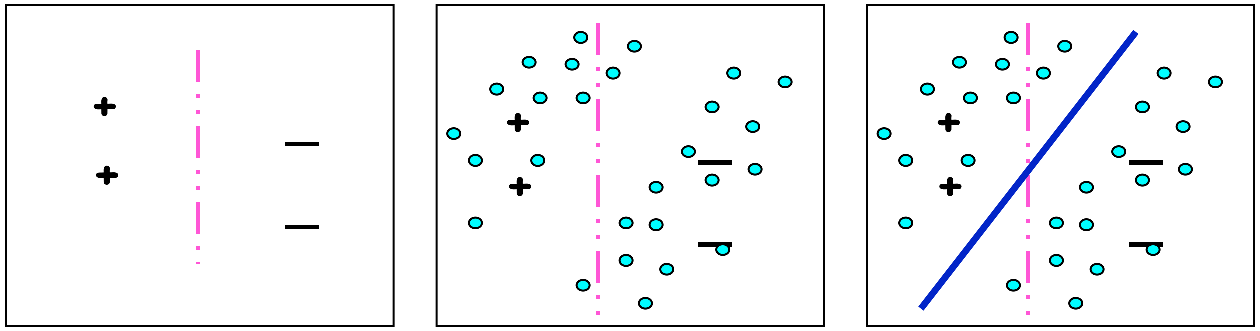
S^3VM [Joachims98]

- Suppose we believe target separator goes through low density regions of the space/large margin.
- Aim for separator with large margin wrt labeled and unlabeled data. (L+U)
- Unfortunately, optimization problem is now NP-hard. Algorithm instead does local optimization.
  - Start with large margin over labeled data. Induces labels on U.
  - Then try flipping labels in greedy fashion.



S^3VM [Joachims98]

- Suppose we believe target separator goes through low density regions of the space/large margin.
- Aim for separator with large margin wrt labeled and unlabeled data. (L+U)
- Unfortunately, optimization problem is now NP-hard. Algorithm instead does local optimization.
  - Or, branch-and-bound, other methods (Chapelle etal06)
- Quite successful on text data.



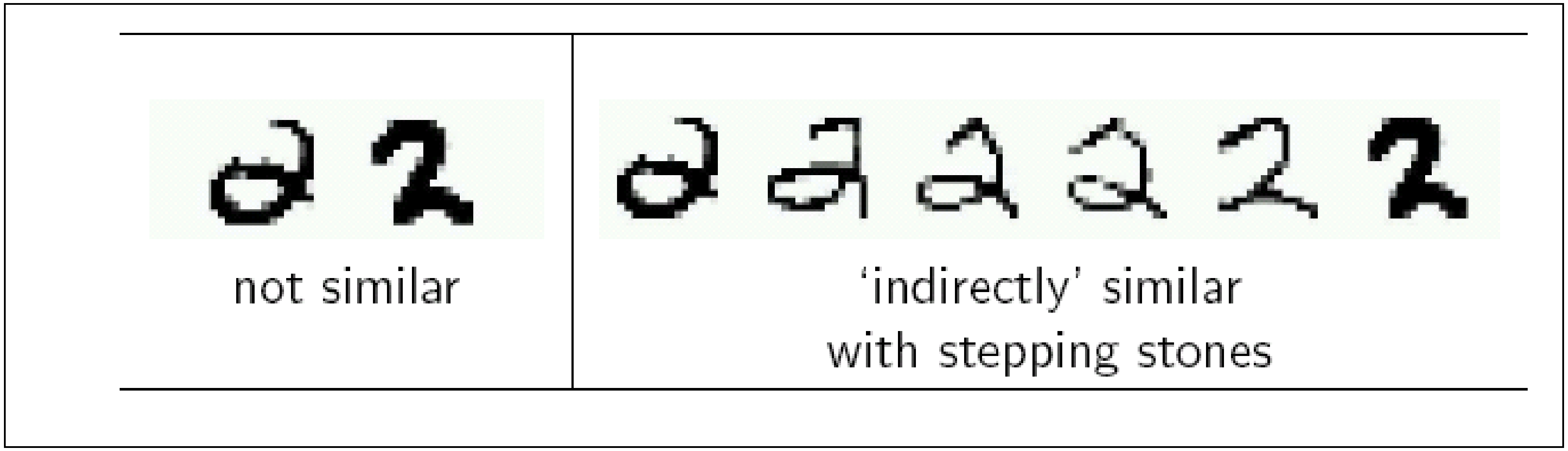
Method 4:

Graph-based methods

Graph-based methods

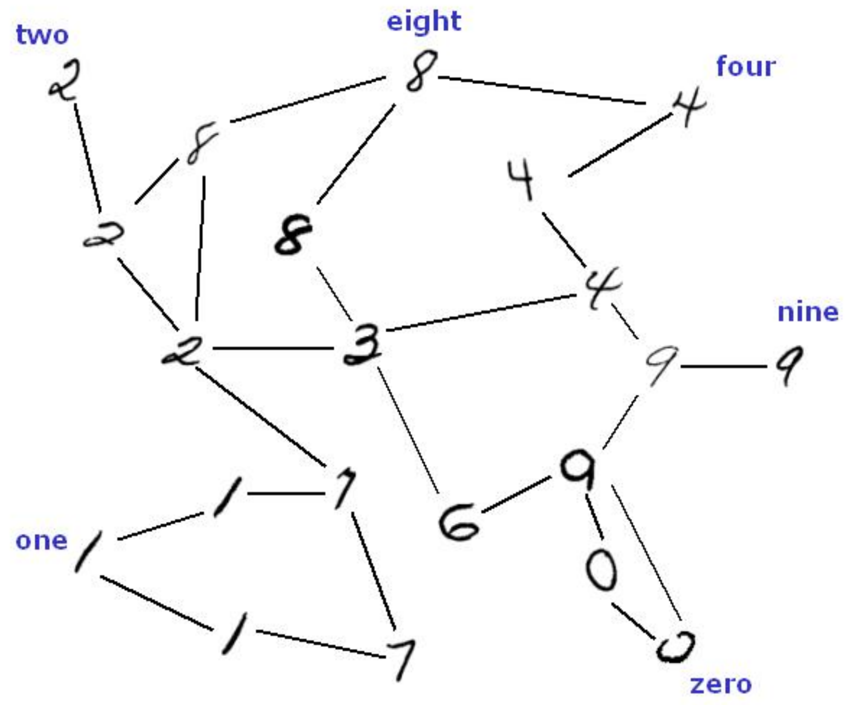
- Suppose we believe that very similar examples probably have the same label.
- If you have a lot of labeled data, this suggests a Nearest-Neighbor type of alg.
- If you have a lot of unlabeled data, perhaps can use them as "stepping stones"

E.g., handwritten digits [Zhu07]:



Graph-based methods

- Idea: construct a graph with edges between very similar examples.
- Unlabeled data can help "glue" the objects of the same class together.



Graph-based methods

- Idea: construct a graph with edges between very similar examples.
- Unlabeled data can help "glue" the objects of the same class together.

image 4005

neighbor 1: time edge

neighbor 2: color edge

neighbor 3: color edge

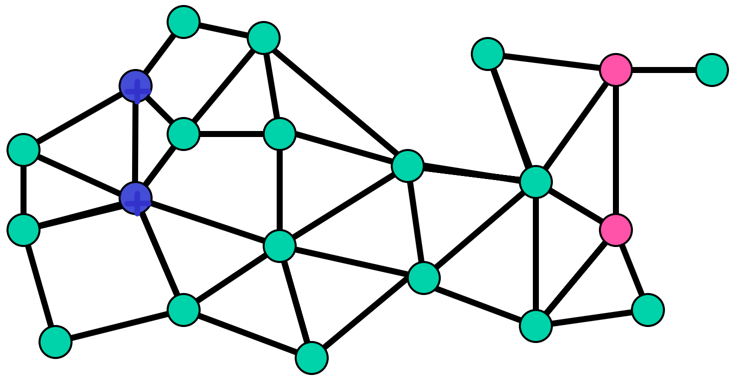
neighbor 4: color edge

neighbor 5: face edge

Graph-based methods

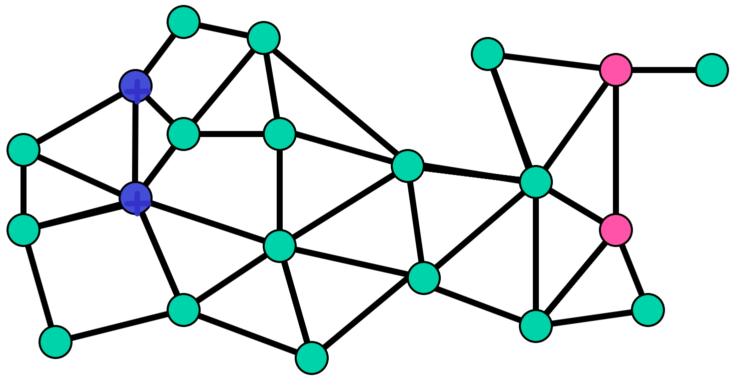
- Idea: construct a graph with edges between very similar examples.
- Unlabeled data can help "glue" the objects of the same class together.
- Solve for:
  - Minimum cut [BC,BLRR]
  - Minimum "soft-cut" [ZGL]

    $\sum_{e=(u,v)}(f(u)-f(v))^2$
  - Spectral partitioning [J]
  - …



Graph-based methods

- Suppose just two labels: 0 & 1.
- Solve for labels 0 · f(x) · 1 for unlabeled examples x to minimize:
  - $\sum_{e=(u,v)} |f(u)-f(v)|$ [soln = minimum cut]
  - $\sum_{e=(u,v)} (f(u)-f(v))^2$ [soln = electric potentials]
  - …



Learning from Positive and Unlabeled Examples

PU learning

Learning from Positive & Unlabeled data

- Positive examples: One has a set of examples of a class *P*, and
- Unlabeled set: also has a set *U* of unlabeled (or mixed) examples with instances from *P* and also not from *P* (*negative examples*).
- Build a classifier: Build a classifier to classify the examples in *U* and/or future (test) data.
- Key feature of the problem: no labeled negative training data.
- We call this problem, PU-learning.

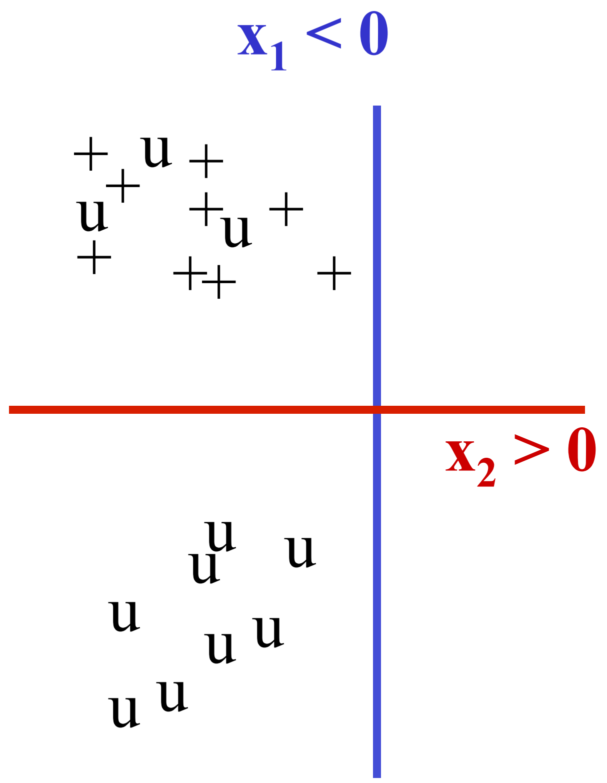
Applications of the problem

- With the growing volume of online texts available through the Web and digital libraries, one often wants to find those documents that are related to one's work or one's interest.
- For example, given a ICML proceedings,
  - find all machine learning papers from AAAI, IJCAI, KDD
  - No labeling of negative examples from each of these collections.
- Similarly, given one's bookmarks (positive documents), identify those documents that are of interest to him/her from Web sources.

Direct Marketing

- Company has database with details of its customer – positive examples, but no information on those who are not their customers, i.e., no negative examples.
- Want to find people who are similar to their customers for marketing
- Buy a database consisting of details of people, some of whom may be potential customers – unlabeled examples.

Are Unlabeled Examples Helpful?



- Function known to be either $x_1 < 0$ or $x_2 > 0$
- Which one is it?

“Not learnable” with only positive examples. However, addition of unlabeled examples makes it learnable.

Theoretical foundations

- (X, Y): X - input vector, $Y \in \{1, -1\}$ - class label.
- f : classification function
- We rewrite the probability of error

$$\Pr[f(X) \neq Y] = \Pr[f(X) = 1 \text{ and } Y = -1] + \Pr[f(X) = -1 \text{ and } Y = 1] \qquad (1)$$

We have $\Pr[f(X) = 1 \text{ and } Y = -1]$

$$= \Pr[f(X) = 1] - \Pr[f(X) = 1 \text{ and } Y = 1]$$
$$= \Pr[f(X) = 1] - (\Pr[Y = 1] - \Pr[f(X) = -1 \text{ and } Y = 1]).$$

Plug this into (1), we obtain

$$\Pr[f(X) \neq Y] = \Pr[f(X) = 1] - \Pr[Y = 1] + 2\Pr[f(X) = -1 \mid Y = 1]\Pr[Y = 1] \qquad (2)$$

Theoretical foundations (cont)

- $\Pr[f(X) \neq Y] = \Pr[f(X) = 1] - \Pr[Y = 1]$ (2)
  $+ 2\Pr[f(X) = -1 | Y = 1] \Pr[Y = 1]$
- Note that $\Pr[Y = 1]$ is constant.
- If we can hold $\Pr[f(X) = -1 | Y = 1]$ small, then learning is approximately the same as minimizing $\Pr[f(X) = 1]$.
- Holding $\Pr[f(X) = -1 | Y = 1]$ small while minimizing $\Pr[f(X) = 1]$ is approximately the same as
  - minimizing $\Pr_U[f(X) = 1]$
  - while holding $\Pr_P[f(X) = 1] \geq r$ (where r is recall Pr[f(X)=1| Y=1]) which is the same as ($\Pr_p[f(X) = -1] \leq 1 - r$)

  if the set of positive examples P and the set of unlabeled examples U are large enough.
- Theorem 1 and Theorem 2 in [Liu et al 2002] state these formally in the noiseless case and in the noisy case.

Put it simply

- A constrained optimization problem.
- A reasonably good generalization (learning) result can be achieved
  - If the algorithm tries to minimize the number of unlabeled examples labeled as positive
  - subject to the constraint that the fraction of errors on the positive examples is no more than 1-*r*.

An illustration

- Assume a linear classifier. Line 3 is the best solution.

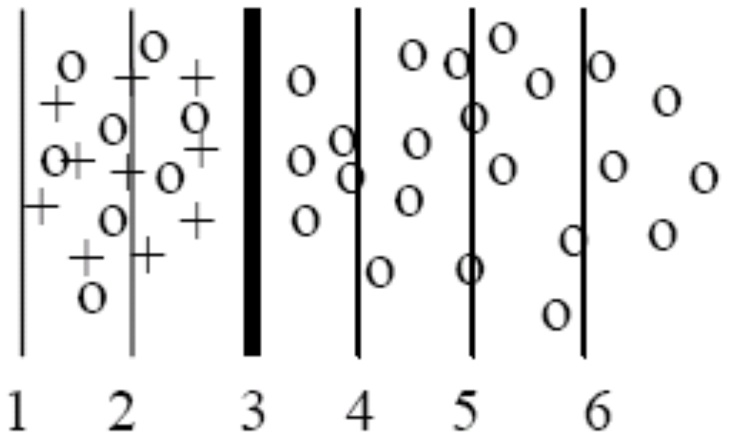
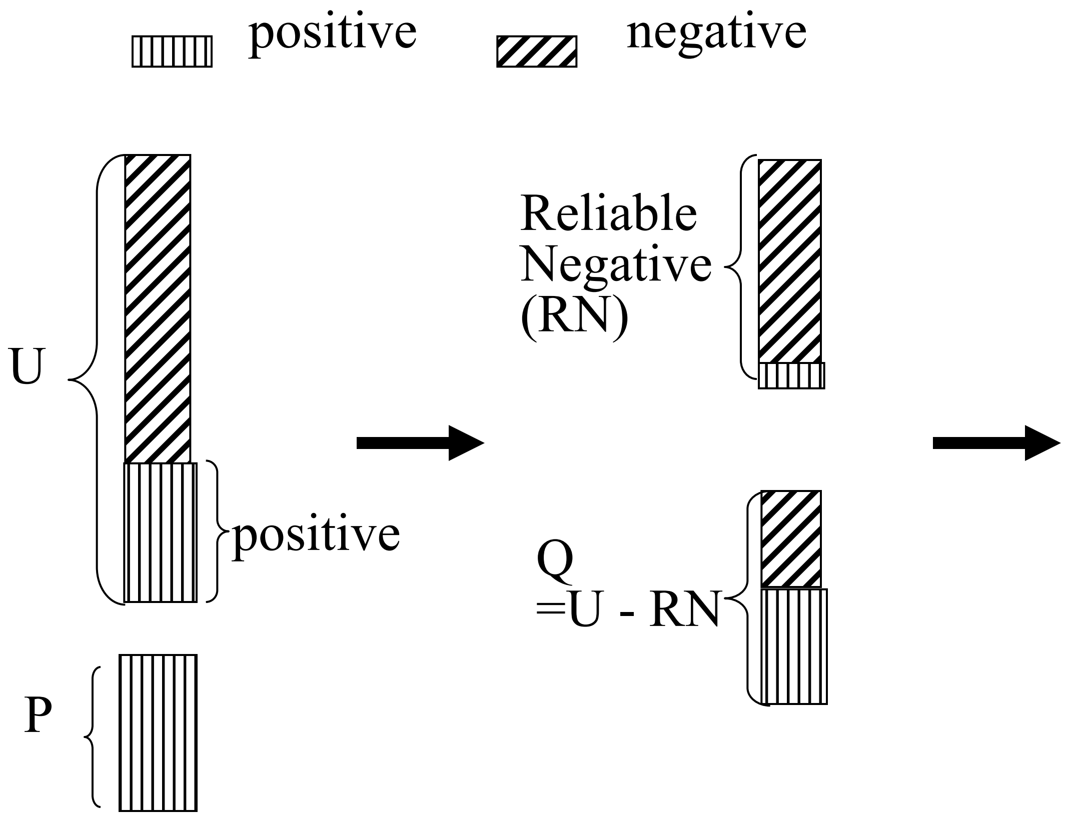


Fig. 5.6. An illustration of the constrained optimization problem

Existing 2-step strategy

- Step 1: Identifying a set of reliable negative documents from the unlabeled set.
  - S-EM [Liu et al, 2002] uses a Spy technique,
  - PEBL [Yu et al, 2002] uses a 1-DNF technique
  - Roc-SVM [Li & Liu, 2003] uses the Rocchio algorithm.
  - ...
- Step 2: Building a sequence of classifiers by iteratively applying a classification algorithm and then selecting a good classifier.
  - S-EM uses the Expectation Maximization (EM) algorithm, with an error based classifier selection mechanism
  - PEBL uses SVM, and gives the classifier at convergence. I.e., no classifier selection.
  - Roc-SVM uses SVM with a heuristic method for selecting the final classifier.

Step 1 Step 2



Using P, RN and Q to build the final classifier iteratively

or

Using only P and RN to build a classifier

Step 1: The Spy technique

- Sample a certain % of positive examples and put them into unlabeled set to act as "spies".
- Run a classification algorithm assuming all unlabeled examples are negative,
  - we will know the behavior of those actual positive examples in the unlabeled set through the "spies".
- We can then extract reliable negative examples from the unlabeled set more accurately.

Step 1: Other methods

- 1-DNF method:
  - Find the set of words W that occur in the positive documents more frequently than in the unlabeled set.
  - Extract those documents from unlabeled set that do not contain any word in W. These documents form the reliable negative documents.
- Rocchio method from information retrieval.
- Naïve Bayesian method.

Step 2: Running EM or SVM iteratively

(1) Running a classification algorithm iteratively
- Run EM using P, RN and Q until it converges, or
- Run SVM iteratively using P, RN and Q until this no document from Q can be classified as negative. RN and Q are updated in each iteration, or
- …

(2) Classifier selection.

Do they follow the theory?

- Yes, heuristic methods because
  - Step 1 tries to find some initial reliable negative examples from the unlabeled set.
  - Step 2 tried to identify more and more negative examples iteratively.
- The two steps together form an iterative strategy of increasing the number of unlabeled examples that are classified as negative while maintaining the positive examples correctly classified.

Can SVM be applied directly?

- Can we use SVM to directly deal with the problem of learning with positive and unlabeled examples, without using two steps?
- Yes, with a little re-formulation.

Support Vector Machines

- Support vector machines (SVM) are linear functions of the form $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, where **w** is the weight vector and **x** is the input vector.
- Let the set of training examples be $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ is an input vector and y_i is its class label, $y_i \in \{1, -1\}$.
- To find the linear function:

Minimize: $$\frac{1}{2}\mathbf{w}^T\mathbf{w}$$

Subject to: $$y_i(\mathbf{w}^T\mathbf{x}_i + b) \quad 1, \quad i = 1, 2, \ldots, n$$

Soft margin SVM

- To deal with cases where there may be no separating hyperplane due to noisy labels of both positive and negative training examples, the soft margin SVM is proposed:

Minimize: $$\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C \overset{n}{\underset{i=1}{}} \quad \xi_i$$

Subject to: $$y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \quad 1 - \xi_i, \quad i = 1, 2, \ldots, n$$

where $C \geq 0$ is a parameter that controls the amount of training errors allowed.

Biased SVM (noiseless case)

- Assume that the first *k*-1 examples are positive examples (labeled 1), while the rest are unlabeled examples, which we label negative (-1).

Minimize: $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C \overset{n}{\underset{i=k}{\quad}} \xi_i$

Subject to: $\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \quad 1, \quad i = 1, 2, ..., k-1$

$-1(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \quad 1 - \xi_i, \quad i = k, k+1, ..., n$

$\xi_i \geq 0$, i = k, k+1…, n

Biased SVM (noisy case)

- If we also allow positive set to have some noisy negative examples, then we have:

Minimize: $$\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C_{+} \mathop{}_{i=1}^{k-1} \xi_i + C_{-} \mathop{}_{i=k}^{n} \xi_i$$

Subject to: $$y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \quad 1 - \xi_i, \quad i = 1,2..., n$$

$$\xi_i \geq 0,\ i = 1,\ 2,\ \ldots,\ n.$$

- This turns out to be the same as the asymmetric cost SVM for dealing with unbalanced data. Of course, we have a different motivation.

Estimating performance

- We need to estimate the performance in order to select the parameters.
- Since learning from positive and negative examples often arise in retrieval situations, we use F score as the classification performance measure $F = 2pr / (p+r)$ (p: precision, r: recall).
- To get a high F score, both precision and recall have to be high.
- However, without labeled negative examples, we do not know how to estimate the F score.

A performance criterion

- Performance criteria $pr/\Pr[Y=1]$: It can be estimated directly from the validation set as $r^2/\Pr[f(X) = 1]$
  - Recall $r = \Pr[f(X)=1 \mid Y=1]$
  - Precision $p = \Pr[Y=1 \mid f(X)=1]$

To see this

$$\underbrace{\Pr[f(X)=1|Y=1]}_{r} \Pr[Y=1] = \underbrace{\Pr[Y=1|f(X)=1]}_{p} \Pr[f(X)=1]$$

$$\Leftrightarrow \quad \frac{r}{\Pr[f(X)=1]} = \frac{p}{\Pr[Y=1]}$$

//both side times r

- Behavior similar to the F-score (= $2pr / (p+r)$)

A performance criterion (cont …)

- $r^2/\Pr[f(X) = 1]$
- r can be estimated from positive examples in the validation set.
- $\Pr[f(X) = 1]$ can be obtained using the full validation set.
- This criterion actually reflects the theory very well.

Empirical Evaluation

- Two-step strategy: We implemented a benchmark system, called LPU, which is available at http://www.cs.uic.edu/~liub/LPU/LPU-download.html
  - Step 1:
    - Spy
    - 1-DNF
    - Rocchio
    - Naïve Bayesian (NB)
  - Step 2:
    - EM with classifier selection
    - SVM: Run SVM once.
    - SVM-I: Run SVM iteratively and give converged classifier.
    - SVM-IS: Run SVM iteratively with classifier selection
- Biased-SVM (we used SVMlight package)

Table 1: Average F scores on Reuters collection

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Step1	1-DNF	1-DNF		1-DNF		Spy	Spy	Spy	Rocchio	Rocchio	Rocchio		NB	NB	NB	NB	
Step2	EM	SVM	PEBL	SVM-IS	S-EM	SVM	SVM-I	SVM-IS	EM	SVM	SVM-I	Roc-SVM	EM	SVM	SVM-I	SVM-IS	NB
0.1	0.187	0.423	0.001	0.423	0.547	0.329	0.006	0.328	0.644	0.589	0.001	0.589	0.547	0.115	0.006	0.115	0.514
0.2	0.177	0.242	0.071	0.242	0.674	0.507	0.047	0.507	0.631	0.737	0.124	0.737	0.693	0.428	0.077	0.428	0.681
0.3	0.182	0.269	0.250	0.268	0.659	0.733	0.235	0.733	0.623	0.780	0.242	0.780	0.695	0.664	0.235	0.664	0.699
0.4	0.178	0.190	0.582	0.228	0.661	0.782	0.549	0.780	0.617	0.805	0.561	0.784	0.693	0.784	0.557	0.782	0.708
0.5	0.179	0.196	0.742	0.358	0.673	0.807	0.715	0.799	0.614	0.790	0.737	0.799	0.685	0.797	0.721	0.789	0.707
0.6	0.180	0.211	0.810	0.573	0.669	0.833	0.804	0.820	0.597	0.793	0.813	0.811	0.670	0.832	0.808	0.824	0.694
0.7	0.175	0.179	0.824	0.425	0.667	0.843	0.821	0.842	0.585	0.793	0.823	0.834	0.664	0.845	0.822	0.843	0.687
0.8	0.175	0.178	0.868	0.650	0.649	0.861	0.865	0.858	0.575	0.787	0.867	0.864	0.651	0.859	0.865	0.858	0.677
0.9	0.172	0.190	0.860	0.716	0.658	0.859	0.859	0.853	0.580	0.776	0.861	0.861	0.651	0.846	0.858	0.845	0.674

Table 2: Average F scores on 20Newsgroup collection

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Step1	1-DNF	1-DNF		1-DNF		Spy	Spy	Spy	Rocchio	Rocchio	Rocchio		NB	NB	NB	NB	
Step2	EM	SVM	PEBL	SVM-IS	S-EM	SVM	SVM-I	SVM-IS	EM	SVM	SVM-I	Roc-SVM	EM	SVM	SVM-I	SVM-IS	NB
0.1	0.145	0.545	0.039	0.545	0.460	0.097	0.003	0.097	0.557	0.295	0.003	0.295	0.368	0.020	0.003	0.020	0.333
0.2	0.125	0.371	0.074	0.371	0.640	0.408	0.014	0.408	0.670	0.546	0.014	0.546	0.649	0.232	0.013	0.232	0.611
0.3	0.123	0.288	0.201	0.288	0.665	0.625	0.154	0.625	0.673	0.644	0.121	0.644	0.689	0.469	0.120	0.469	0.674
0.4	0.122	0.260	0.342	0.258	0.683	0.684	0.354	0.684	0.671	0.690	0.385	0.682	0.705	0.610	0.354	0.603	0.704
0.5	0.121	0.248	0.563	0.306	0.685	0.715	0.560	0.707	0.663	0.716	0.565	0.708	0.702	0.680	0.554	0.672	0.707
0.6	0.123	0.209	0.646	0.419	0.689	0.758	0.674	0.746	0.663	0.747	0.683	0.738	0.701	0.737	0.670	0.724	0.715
0.7	0.119	0.196	0.715	0.563	0.681	0.774	0.731	0.757	0.660	0.754	0.731	0.746	0.699	0.763	0.728	0.749	0.717
0.8	0.124	0.189	0.689	0.508	0.680	0.789	0.760	0.783	0.654	0.761	0.763	0.766	0.688	0.780	0.758	0.774	0.707
0.9	0.123	0.177	0.716	0.577	0.684	0.807	0.797	0.798	0.654	0.775	0.798	0.790	0.691	0.806	0.797	0.798	0.714

Results of Biased SVM

Table 3: Average F scores on the two collections

	γ	Average F score of Biased-SVM	Previous best F score
Reuters	0.3	0.785	0.78
	0.7	0.856	0.845
20Newsgroup	0.3	0.742	0.689
	0.7	0.805	0.774

Summary

- Gave an overview of the theory on learning with positive and unlabeled examples.
- Described the existing two-step strategy for learning.
- Presented an more principled approach to solve the problem based on a biased SVM formulation.
- Presented a performance measure pr/P(Y=1) that can be estimated from data.
- Experimental results using text classification show the superior classification power of Biased-SVM.
- Some more experimental work are being performed to compare Biased-SVM with weighted logistic regression method [Lee & Liu 2003].

Is there some underlying principle here?

What should be true about the world for unlabeled data to help?

Detour back to standard <u>supervised</u> learning

Then new model

[Joint work with Nina Balcan]

Standard formulation (PAC) for supervised learning

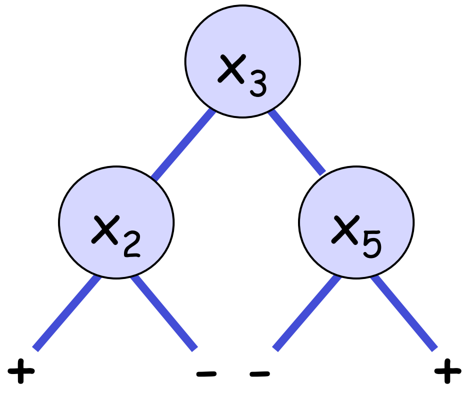
- We are given training set $S = \{(x,f(x))\}$.
  - Assume x's are random sample from underlying distribution D over instance space.
  - Labeled by target function f.
- Alg does optimization over S to produce some hypothesis (prediction rule) h.
- Goal is for h to do well on new examples also from D.

I.e., $\Pr_D[h(x) \neq f(x)] < \varepsilon$.

Standard formulation (PAC) for supervised learning

- Question: why should doing well on S have anything to do with doing well on D?
- Say our algorithm is choosing rules from some class C.
  - E.g., say data is represented by n boolean features, and we are looking for a good decision tree of size O(n).

How big does S have to be to hope performance carries over?



Confidence/sample-complexity

- Consider a rule h with $err(h)>\varepsilon$, that we're worried might fool us.
- Chance that h survives m examples is at most $(1-\varepsilon)^m$.

So, Pr[some rule h with $err(h)>\varepsilon$ is consistent]

$< |C|(1-\varepsilon)^m$.

- This is <0.01 for $m > (1/\varepsilon)[\ln(|C|) + \ln(100)]$

View as just # bits to write h down!

Occam's razor

William of Occam (~1320 AD):

"entities should not be multiplied unnecessarily" (in Latin)

Which we interpret as: "in general, prefer simpler explanations".

Why? Is this a good policy? What if we have different notions of what's simpler?

Occam's razor (contd)

A computer-science-ish way of looking at it:

- Say "simple" = "short description".
- At most 2^s explanations can be < s bits long.
- So, if the number of examples satisfies:

Think of as 10x #bits to write down h.

$$m > (1/\varepsilon)[s \ln(2) + \ln(100)]$$

Then it's unlikely a bad simple (< s bits) explanation will fool you just by chance.

Semi-supervised model

High-level idea:

Intrinsically, using notion of simplicity that is a function of unlabeled data.

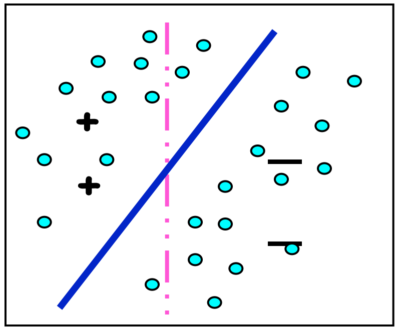
(Formally, of how proposed rule relates to underlying distribution; use unlabeled data to estimate)

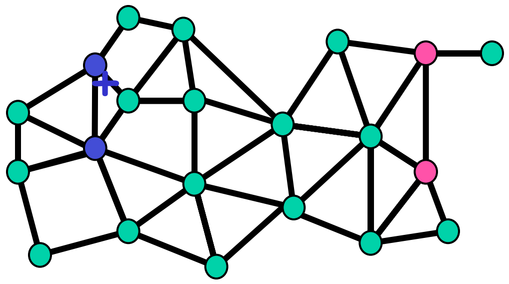
Semi-supervised model

High-level idea:

Intrinsically, using notion of simplicity that is a function of unlabeled data.

E.g., "large margin separator" "small cut" "self-consistent rules"





$h_1(x_1)=h_2(x_2)$

Formally

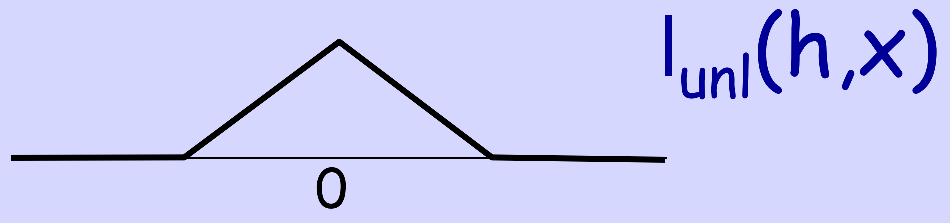
- Convert belief about world into an unlabeled loss function $l_{unl}(h,x)2[0,1]$.
- Defines unlabeled error rate ("incompatibility score")
  - $Err_{unl}(h) = E_{x»D}[l_{unl}(h,x)]$

Co-training: fraction of data pts hx_1,x_2i where $h_1(x_1) \neq h_2(x_2)$

Formally

- Convert belief about world into an unlabeled loss function $l_{unl}(h,x)2[0,1]$.
- Defines unlabeled error rate ("incompatibility score")
  - $Err_{unl}(h) = E_{x»D}[l_{unl}(h,x)]$

S³VM: fraction of data pts x near to separator h.



- Using unlabeled data to estimate this score.

Can use to prove sample bounds

Simple example theorem: (believe target is fully compatible)

Define C(ε) = {h 2 C : err$_{unl}$(h) · ε}.

If we see

$$m_u \geq \frac{1}{\varepsilon}\left[\ln|C| + \ln\frac{2}{\delta}\right]$$

unlabeled examples and

$$m_l \geq \frac{1}{\varepsilon}\left[\ln|C(\varepsilon)| + \ln\frac{2}{\delta}\right]$$

labeled examples, then with probability $\geq 1-\delta$, all $h \in C$ with $\hat{err}(h) = 0$ and $\hat{err}_{unl}(h) = 0$ have $err(h) \leq \varepsilon$.

Bound the # of labeled examples as a measure of the helpfulness of D wrt our incompatibility score.

- a helpful distribution is one in which C(ε) is small

Can use to prove sample bounds

Simple example theorem:

Define C(ε) = {h 2 C : err$_{unl}$(h) · ε}.

If we see

$$m_u \geq \frac{1}{\varepsilon}\left[\ln|C| + \ln\frac{2}{\delta}\right]$$

unlabeled examples and

$$m_l \geq \frac{1}{\varepsilon}\left[\ln|C(\varepsilon)| + \ln\frac{2}{\delta}\right]$$

labeled examples, then with probability $\geq 1-\delta$, all $h \in C$ with $\hat{err}(h) = 0$ and $\hat{err}_{unl}(h) = 0$ have $err(h) \leq \varepsilon$.

Extend to case where target not fully compatible.

Then care about {h2C : err$_{unl}$(h) · ε + err$_{unl}$(f*)}.

When does unlabeled data help?

- Target agrees with beliefs (low unlabeled error rate / incompatibility score).
- Space of rules nearly as compatible as target is "small" (in size or VC-dimension or ε-cover size,…)
- And, have algorithm that can optimize.

Extend to case where target not fully compatible.

Then care about $\{h2C : err_{unl}(h) \cdot \varepsilon + err_{unl}(f^*)\}$.

When does unlabeled data help?

Interesting implication of analysis:

- Best bounds for algorithms that first use unlabeled data to generate set of candidates. (small ε-cover of compatible rules)
- Then use labeled data to select among these.

Unfortunately, often hard to do this algorithmically. Interesting challenge.

(can do for linear separators if have indep given label) learn from single labeled example)

Conclusions

- Semi-supervised learning is an area of increasing importance in Machine Learning.
- Automatic methods of collecting data make it more important than ever to develop methods to make use of unlabeled data.
- Several promising algorithms (only discussed a few). Also new theoretical framework to help guide further development.