Semi-Supervised Learning

©Dr. Min Chi

mchi@ncsu.edu

AI Academy

Supervised Learning

- Document Classification

Sports
Science
News

Supervised Learning (SL)

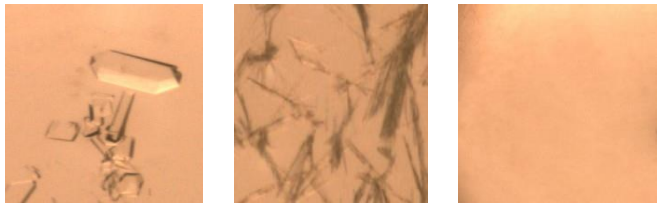
Feature Space $\mathcal{X}$ **Label** Space $\mathcal{Y}$

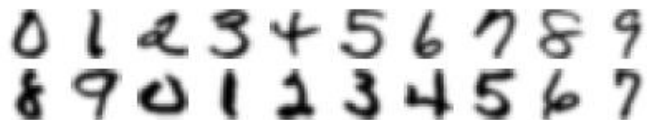
Goal: Construct a **predictor** $f : \mathcal{X} \to \mathcal{Y}$ to minimize

$$R(f) \equiv \mathbb{E}_{XY}\left[\text{loss}(Y, f(X))\right]$$

Training data $\{(X_i, Y_i)\}_{i=1}^{n}$ **Labeled** ⇒ **Learning algorithm** ⇒ Prediction rule $\widehat{f}_n$

Labeled and Unlabeled data





Human expert/
Special equipment/
Experiment

"Crystal" "Needle" "Empty"

"0" "1" "2" …

"Sports"
"News"
"Science"
…

Unlabeled data, X_i

Cheap and abundant !

Labeled data, Y_i

Expensive and scarce !

AI Academy

NC STATE

Free-of-cost labels?

Luis von Ahn: Games with a purpose (ReCaptcha)

Word challenging to OCR
(Optical Character Recognition)

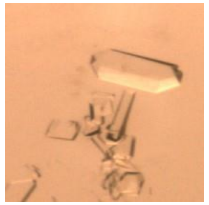
You provide a free label!

Semi-Supervised learning (SSL)

Training data $\{(X_i, Y_i)\}_{i=1}^n$ $\{X_i\}_{i=1}^m$ ⇨ **Learning algorithm** ⇨ Prediction rule $\widehat{f}_{n,m}$

Supervised learning (SL)

Labeled data $\{X_i, Y_i\}_{i=1}^n$



X_i "Crystal" Y_i

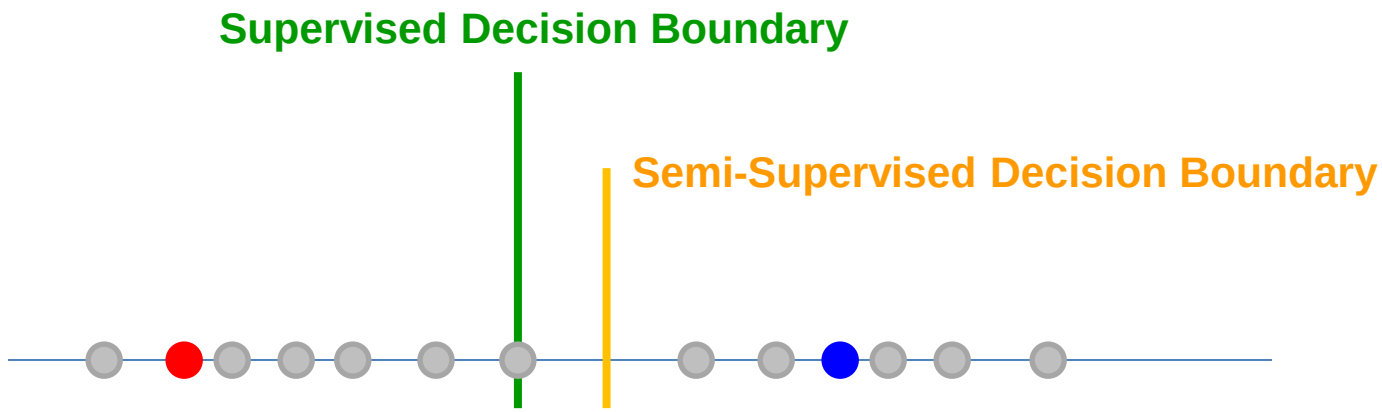
Semi-Supervised learning (SSL)

Labeled data $\{X_i, Y_i\}_{i=1}^n$ **and** Unlabeled data $\{X_i\}_{i=1}^m$ $m \gg n$

Goal: Learn a better prediction rule than based on labeled data alone.

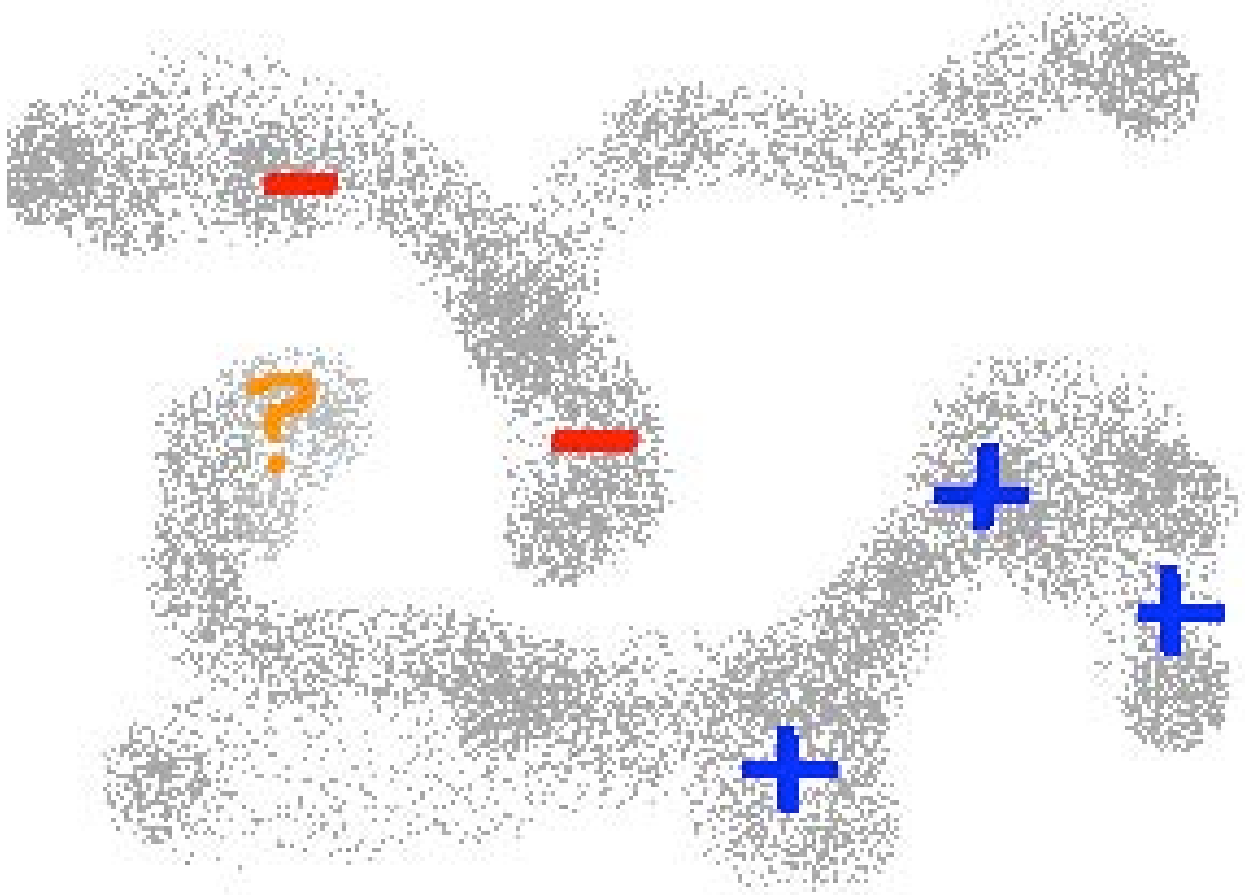
Can unlabeled data help?

- Positive labeled data
- Negative labeled data
- Unlabeled data



Assume each class is a coherent group (e.g. Gaussian)

Then unlabeled data can help identify the boundary more accurately.



When can Unlabeled Data help Supervised Learning

Consider setting:

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

Given:

- iid labeled examples
- iid unlabeled examples

Determine:

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

Notation

- instance $\mathbf{x}$, label y
- learner $f : \mathcal{X} \mapsto \mathcal{Y}$
- labeled data $(X_l, Y_l) = \{(x_{1:l}, y_{1:l})\}$
- unlabeled data $X_u = \{\mathbf{x}_{l+1:l+u}\}$, available during training. Usually $l \ll u$. Let $n = l + u$
- test data $\{(x_{n+1\ldots}, y_{n+1\ldots})\}$, not available during training

Semi-Supervised Learning

- Self-Training
- Re-weight labeled examples
- EM generative models, mixture models
- Label Propagation (Graph-based methods)
- Co-Training
- Semi-supervised SVM
- Many others

Self-training

Our first SSL algorithm:

Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$.

1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat:
3. Train f from L using supervised learning.
4. Apply f to the unlabeled instances in U.
5. Remove a subset S from U; add $\{(\mathbf{x}, f(\mathbf{x}))|\mathbf{x} \in S\}$ to L.

Self-training is a *wrapper* method

- the choice of learner for f in step 3 is left completely open
- good for many real world tasks like natural language processing
- but mistake by f can reinforce itself

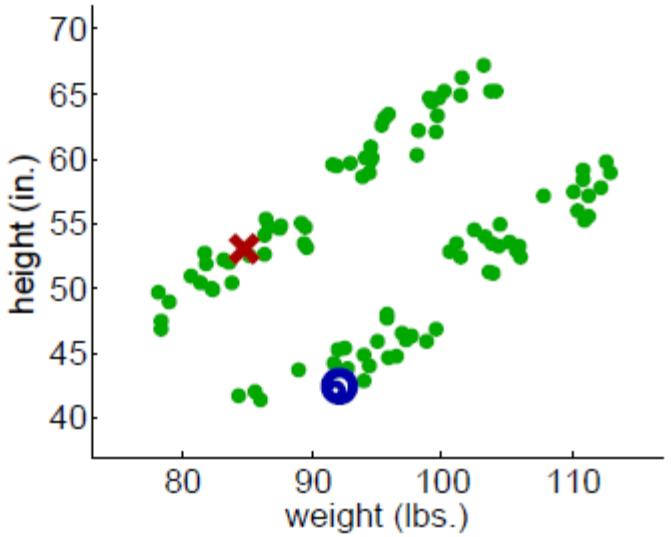
Self-training Example

Propagating 1-NN

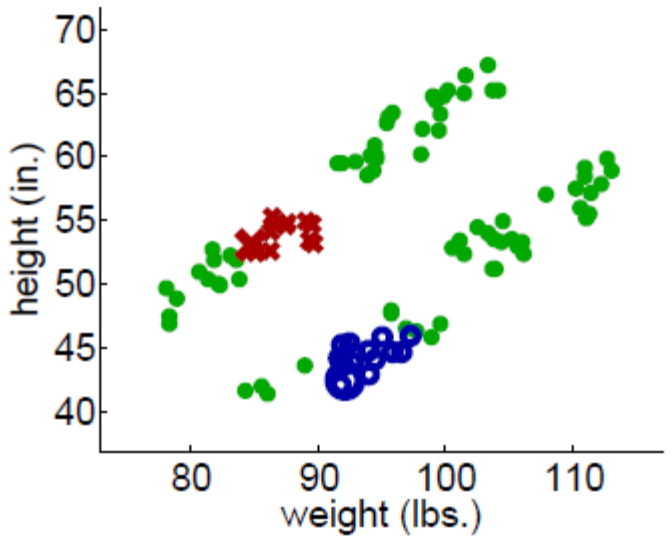
Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$, distance function $d()$.

1. Initially, let $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
2. Repeat until U is empty:
3. Select $\mathbf{x} = \text{argmin}_{\mathbf{x} \in U} \min_{\mathbf{x}' \in L} d(\mathbf{x}, \mathbf{x}')$.
4. Set $f(\mathbf{x})$ to the label of $\mathbf{x}$'s nearest instance in L. Break ties randomly.
5. Remove $\mathbf{x}$ from U; add $(\mathbf{x}, f(\mathbf{x}))$ to L.

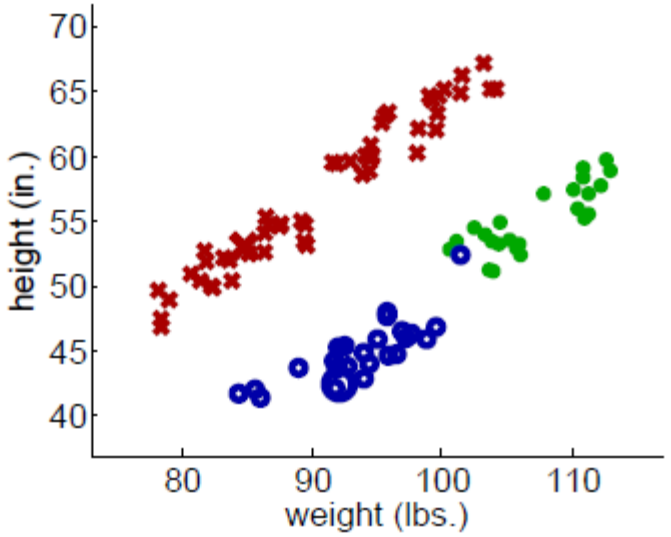
Propagating 1-Nearest-Neighbor: now it works



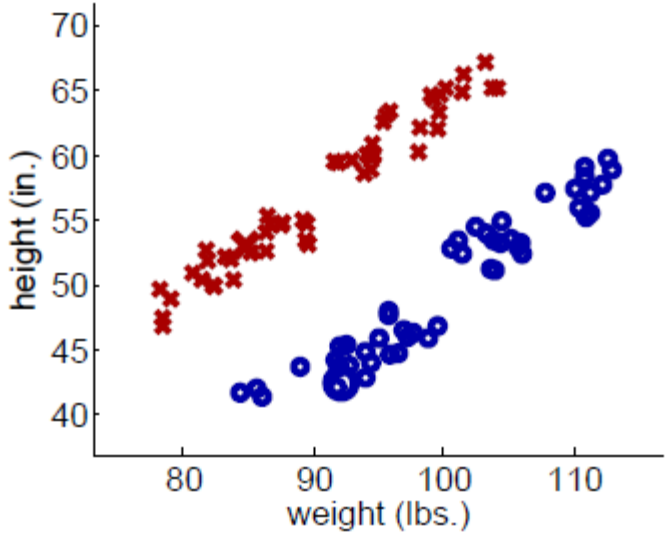
(a) Iteration 1



(b) Iteration 25



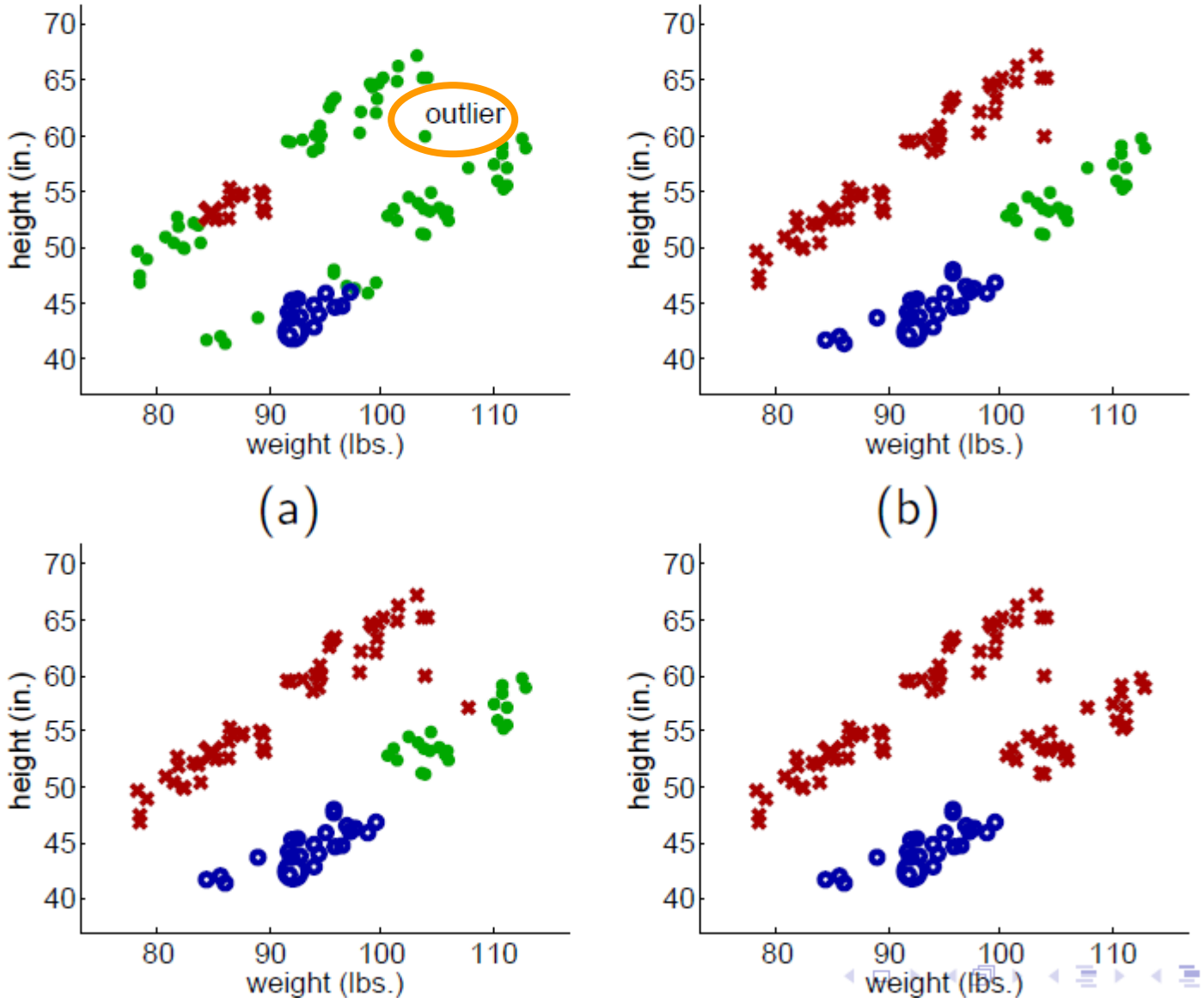
(c) Iteration 74



(d) Final labeling of all instances

Propagating 1-Nearest-Neighbor: now it doesn't

But with a single outlier...



Re-weight labeled examples

Re-weighting labeled examples

- Supervised Learning: *minimize errors over labeled examples*
- Real goal: *minimize error over future examples* drawn from the same underlying distribution
- If we know the underlying distribution, we can weight each training example.
- Unlabeled data allows us to estimate the input data distribution more accurately

Reweight Labeled Examples

From supervised learning:

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

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

Reweight Labeled Examples

From supervised learning:

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

# of times we have x in the labeled set

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

Reweight Labeled Examples

From supervised learning:

$$\hat{f} \leftarrow \underset{h \in H}{\text{argmin}} \frac{1}{L} \sum_{<x,y> \in L} \delta(h(x) \neq y)$$

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

Goal: For whole population.

$$\hat{f} \leftarrow \text{argmin}_{h \in H} \, Pr[h(x) \neq f(x)] \quad x \in p(X)$$

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

From supervised learning:

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

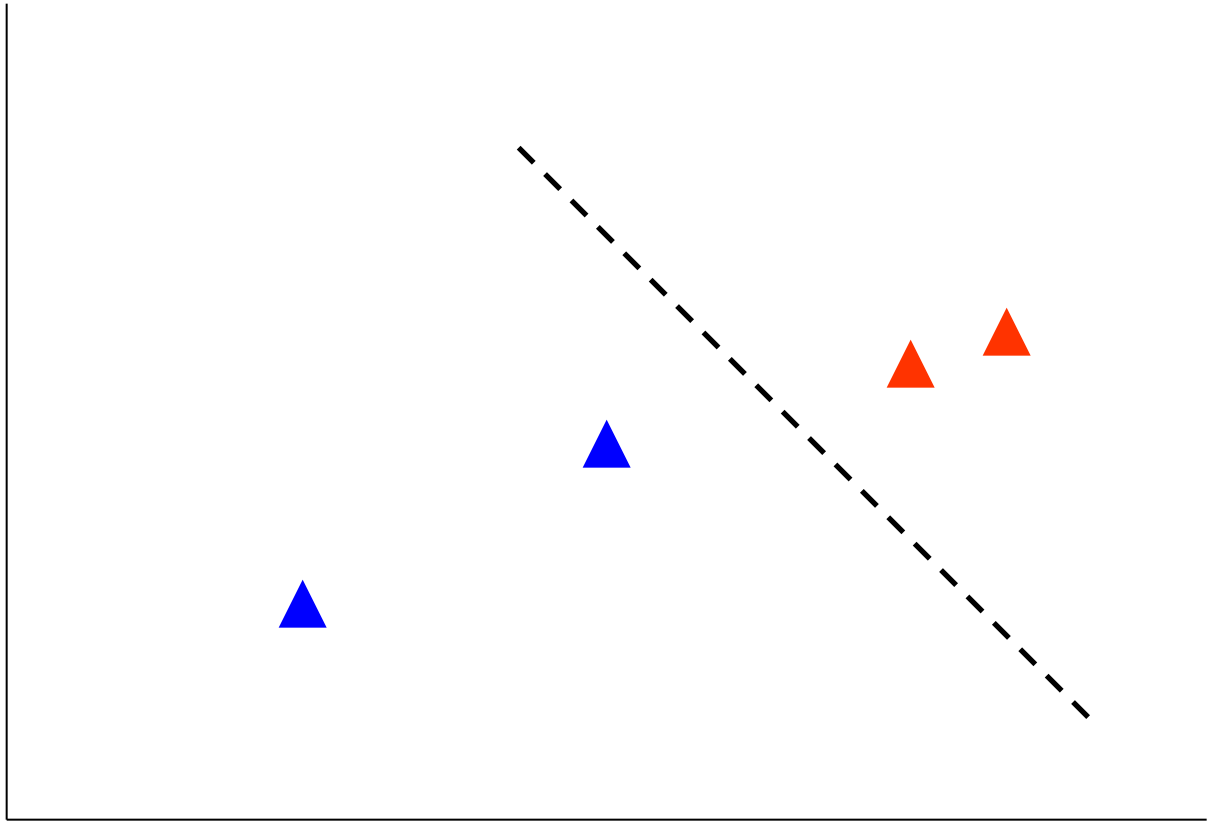
Goal: For whole population.

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

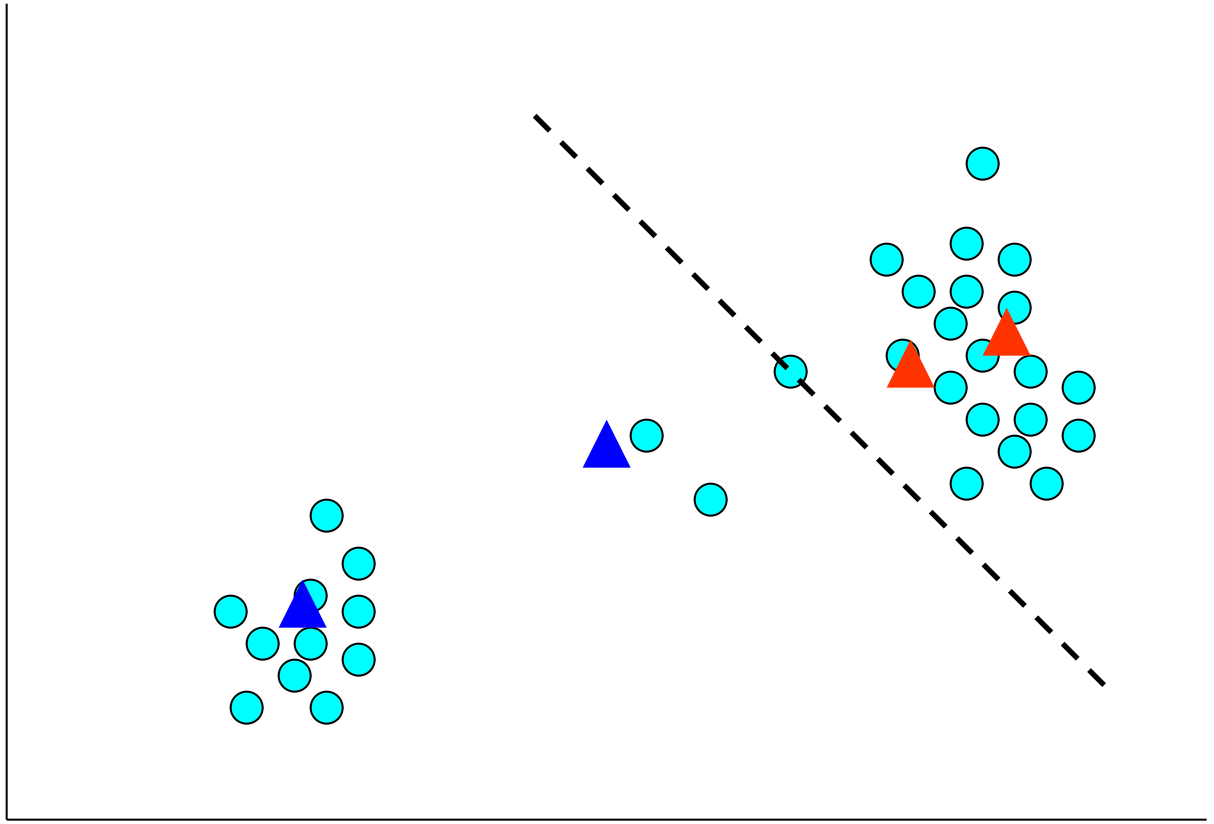
We can use Unlabeled data (U) to improve the approximation.

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

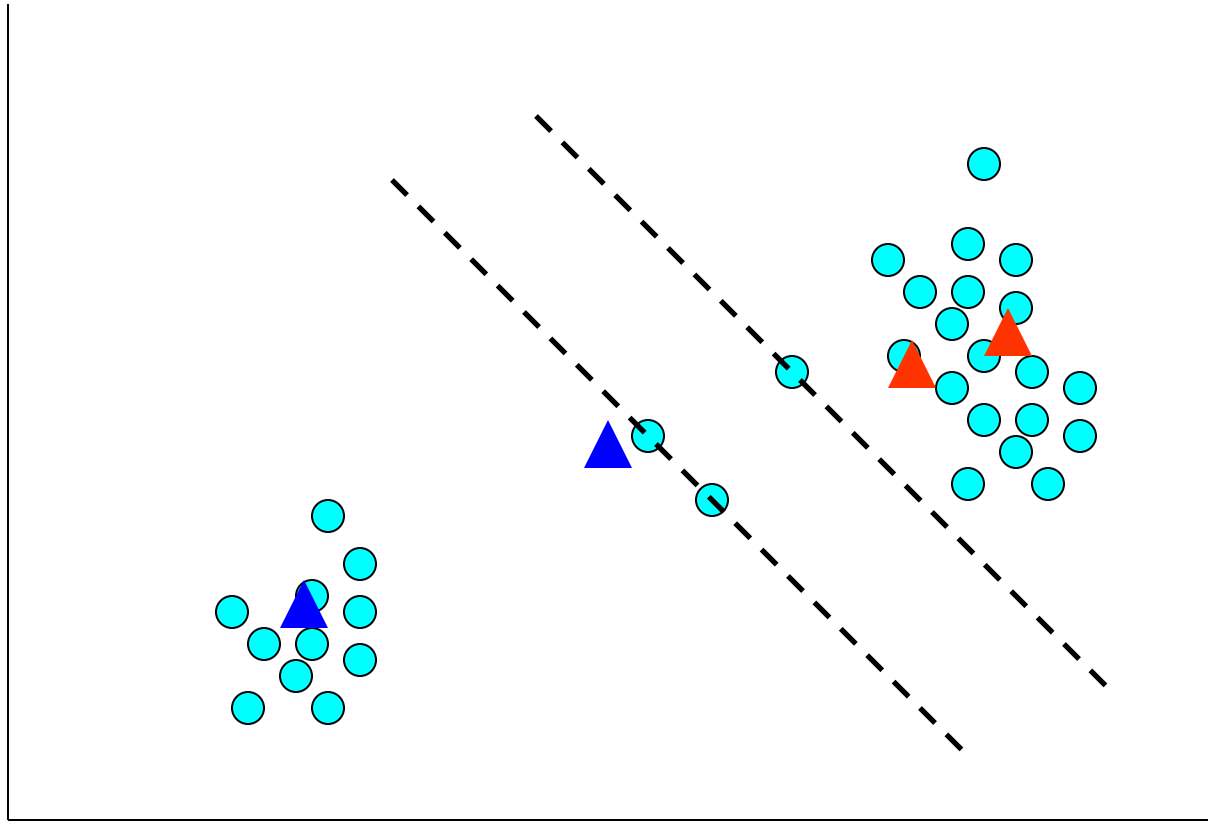
An Example



An Example



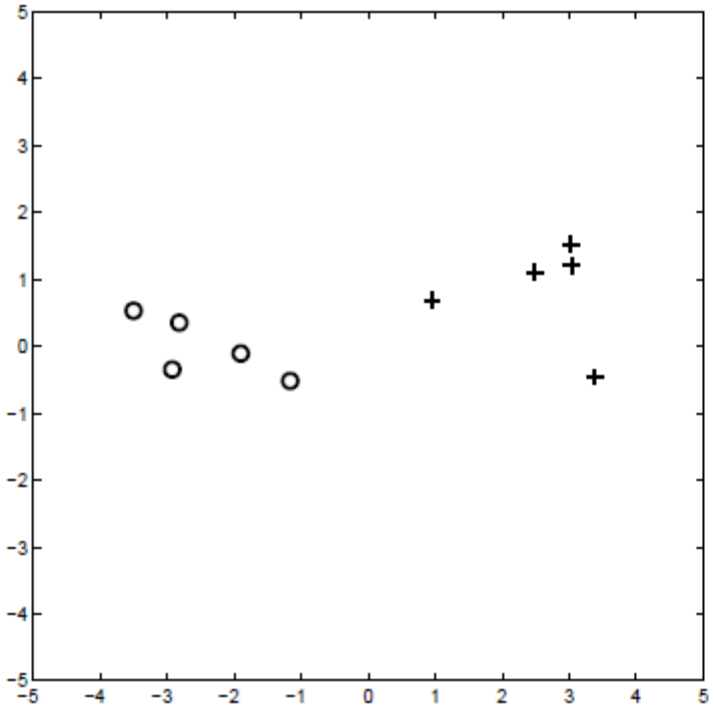
An Example



EM generative models, mixture models

Mixture Models for Labeled Data

Labeled data (X_l, Y_l):



Assuming each class has a Gaussian distribution, what is the decision boundary?

Mixture Models for Labeled Data

Model parameters: $\theta = \{w_1, w_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$
The GMM:

Estimate the parameters from the labeled data

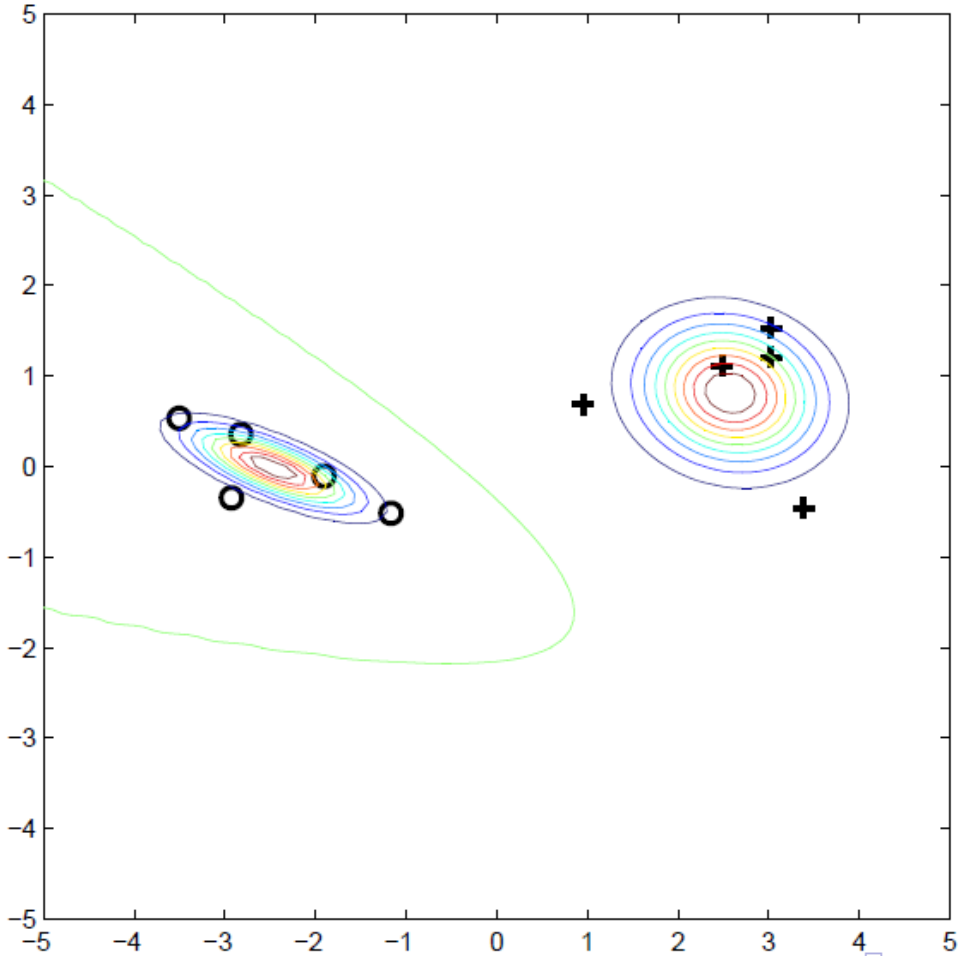
$$\begin{aligned} p(x, y|\theta) &= p(y|\theta)p(x|y, \theta) \\ &= w_y \mathcal{N}(x; \mu_y, \Sigma_y) \end{aligned}$$

Classification: $p(y|x, \theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)} \gtrless 1/2$

Decision for any test point not in the labeled dataset

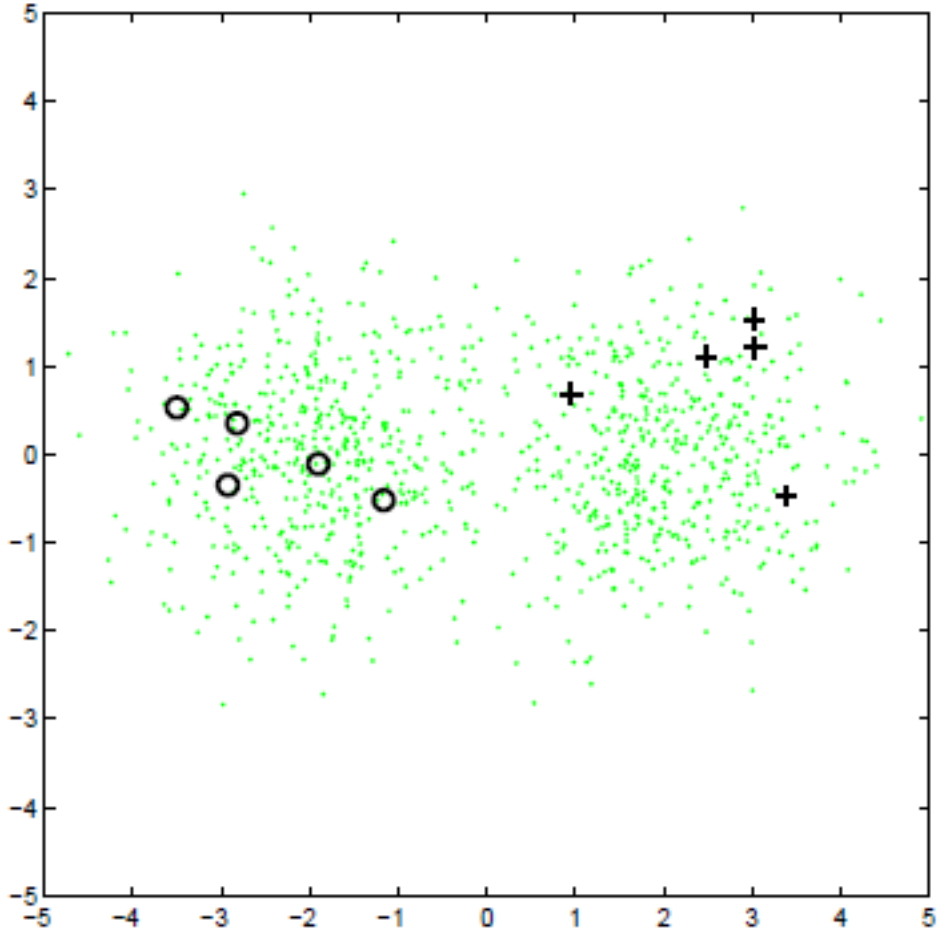
Mixture Models for Labeled Data

The most likely model, and its decision boundary:



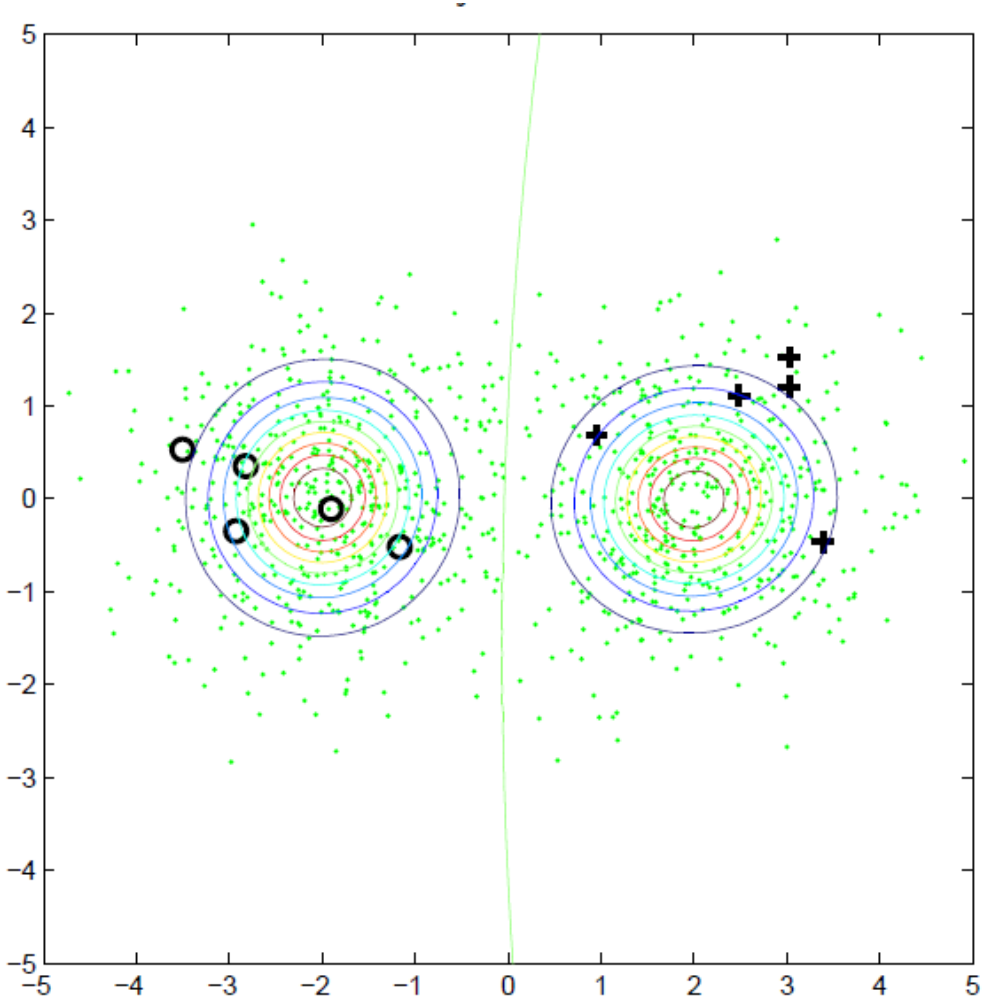
Mixture Models for SSL Data

Adding unlabeled data:



Mixture Models

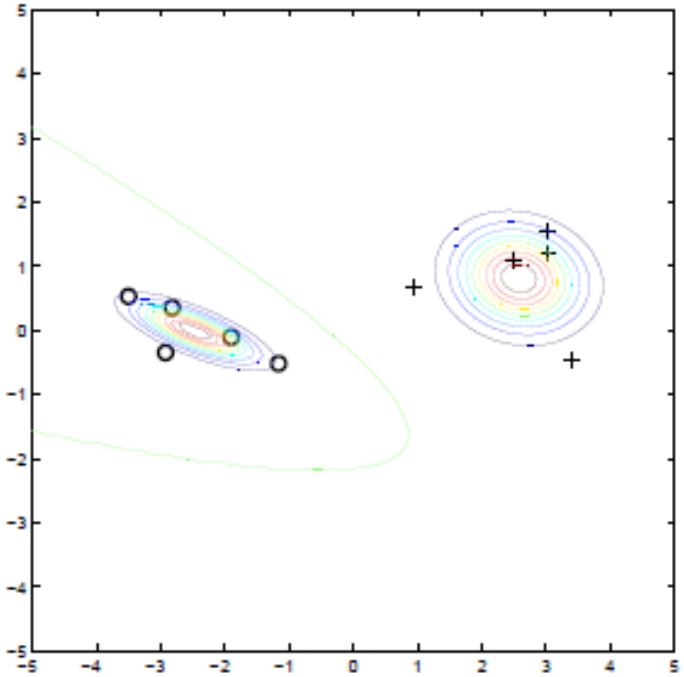
With unlabeled data, the most likely model and its decision boundary:



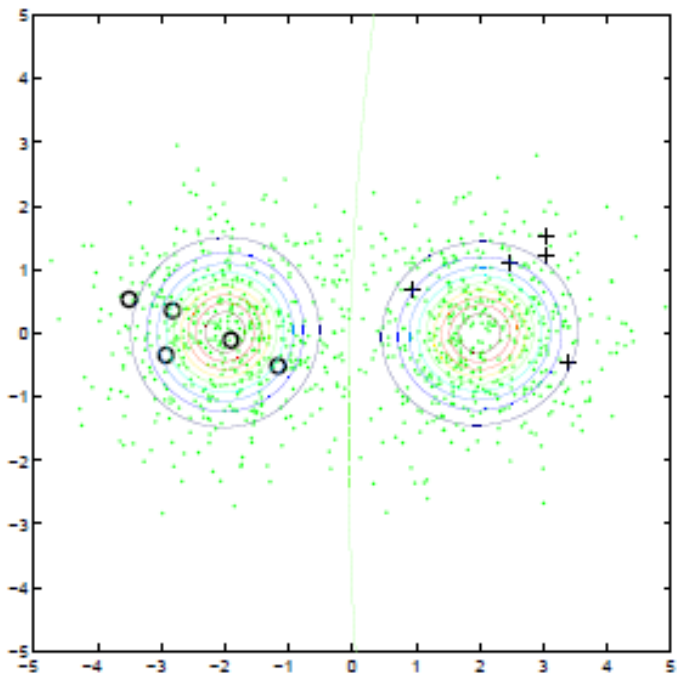
Mixture Models: SL vs SSL

They are different because they maximize different quantities.

$p(X_l, Y_l|\theta)$



$p(X_l, Y_l, X_u|\theta)$



Mixture Models

Assumption

knowledge of the model form $p(X, Y|\theta)$.

- joint and marginal likelihood

$$p(X_l, Y_l, X_u|\theta) = \sum_{Y_u} p(X_l, Y_l, X_u, Y_u|\theta)$$

- find the maximum likelihood estimate (MLE) of θ, the maximum a posteriori (MAP) estimate, or be Bayesian

Gaussian Mixture Models

Binary classification with GMM using MLE.

- with only labeled data
  - $\log p(X_l, Y_l|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i, \theta)$
  - MLE for θ trivial (sample mean and covariance)

- with both labeled and unlabeled data

$$\log p(X_l, Y_l, X_u|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i, \theta) + \sum_{i=l+1}^{l+u} \log \left(\sum_{y=1}^{2} p(y|\theta)p(x_i|y, \theta) \right)$$

  - MLE harder (hidden variables): EM

EM for Gaussian Mixture Models

1. Start from MLE $\theta = \{w, \mu, \Sigma\}_{1:2}$ on (X_l, Y_l),
   - w_c=proportion of class c
   - μ_c=sample mean of class c
   - Σ_c=sample cov of class c

   repeat:

2. The E-step: compute the expected label $p(y|x,\theta) = \frac{p(x,y|\theta)}{\sum_{y'} p(x,y'|\theta)}$ for all $x \in X_u$
   - label $p(y = 1|x, \theta)$-fraction of x with class 1
   - label $p(y = 2|x, \theta)$-fraction of x with class 2

3. The M-step: update MLE θ with (now labeled) X_u

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

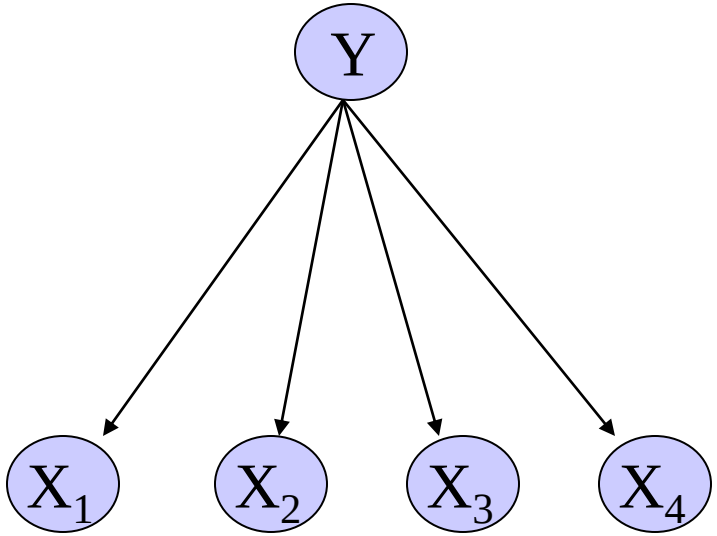
Departments	
departmental	0.01246
colloquia	0.01076
epartment	0.01045
seminars	0.00997
schedules	0.00879
webmaster	0.00879
events	0.00826
facilities	0.00807
eople	0.00772
postgraduate	0.00764

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

Generative Bayes model

Learn P(Y|X)



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

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 the labels assigned to U (**soft membership**)

- Under certain conditions, guaranteed to converge to (local) maximum likelihood h

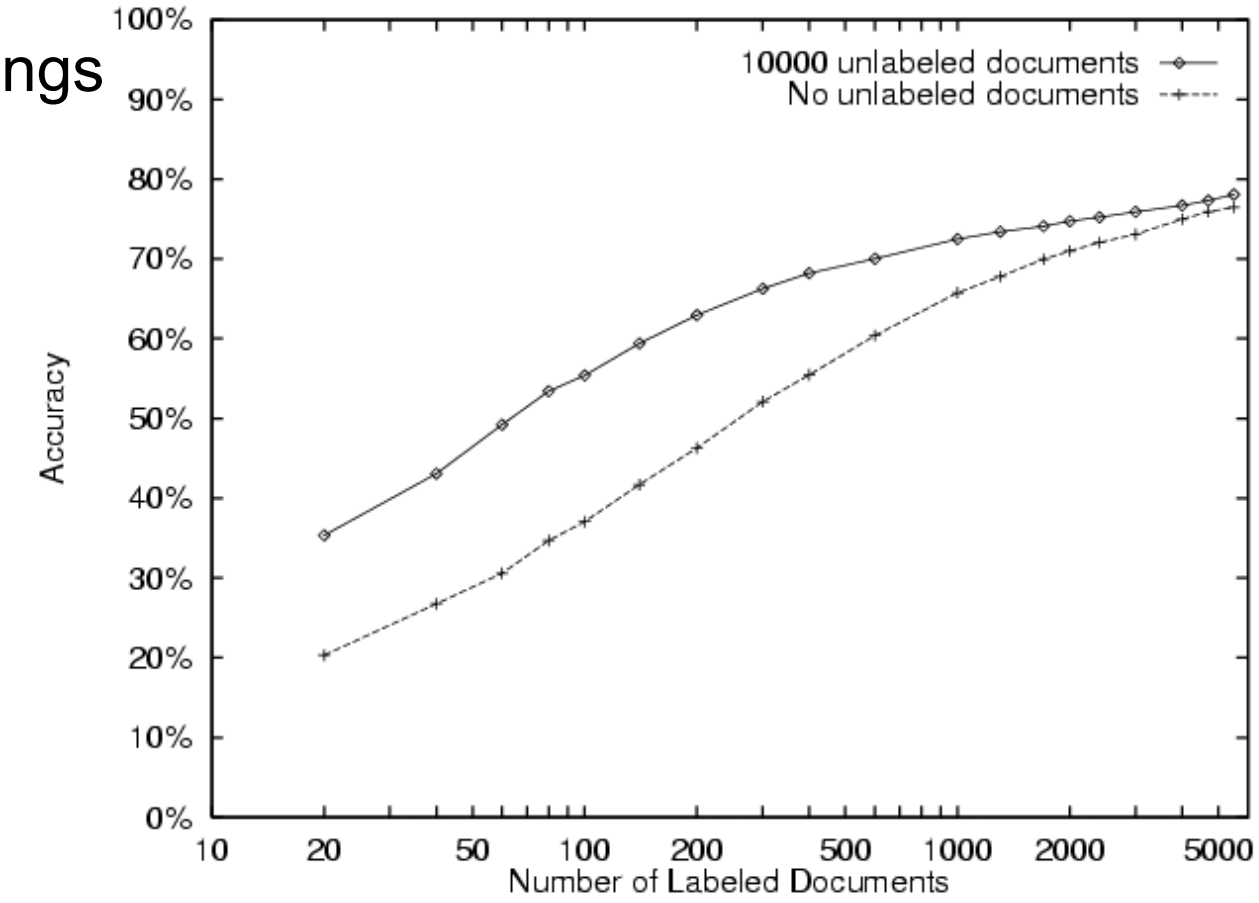
Table 3. Lists of the words most predictive of the course class in the WebKB data set, as they change over iterations of EM for a specific trial. By the second iteration of EM, many common course-related words appear. The symbol *D* indicates an arbitrary digit.

Iteration 0	Iteration 1	Iteration 2
intelligence	*DD*	*D*
DD	*D*	*DD*
artificial	lecture	lecture
understanding	cc	cc
*DD*w	*D**	*DD:DD*
dist	*DD:DD*	due
identical	handout	*D**
rus	due	homework
arrange	problem	assignment
games	set	handout
dartmouth	tay	set
natural	*DD*am	hw
cognitive	yurttas	exam
logic	homework	problem
proving	kfoury	*DD*am
prolog	sec	postscript
knowledge	postscript	solution
human	exam	quiz
representation	solution	chapter
field	assaf	ascii

Experimental Evaluation

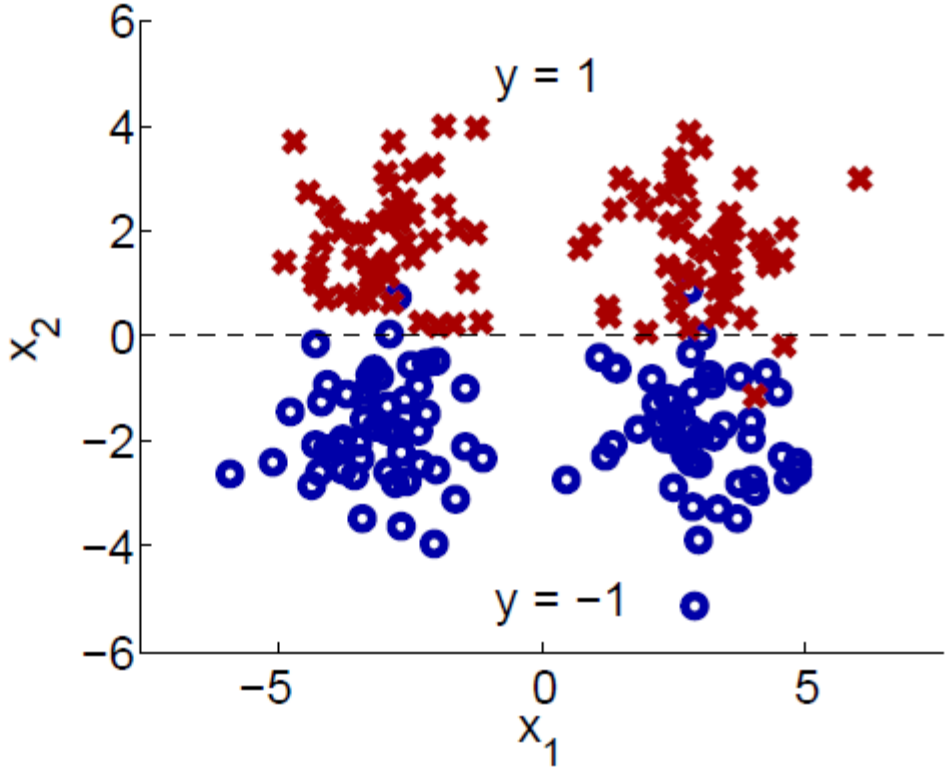
Newsgroup postings

– 20 newsgroups, 1000/group

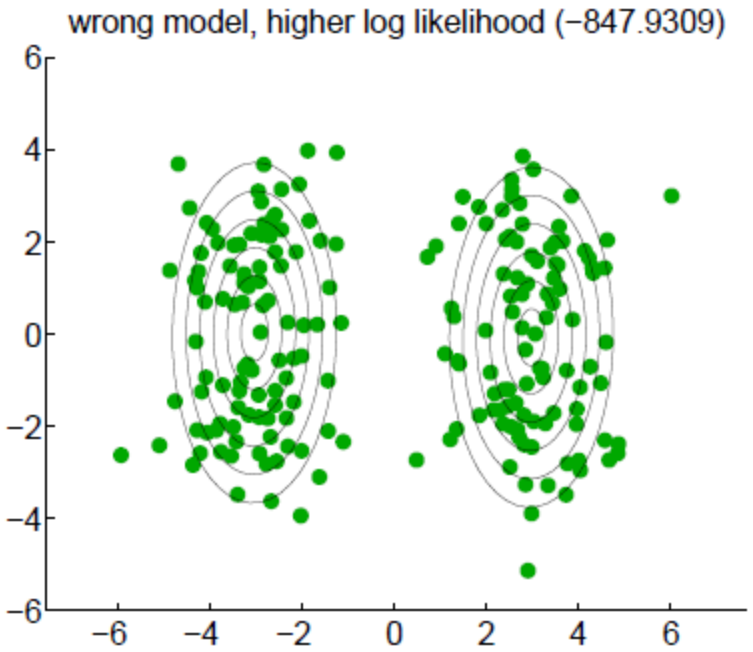


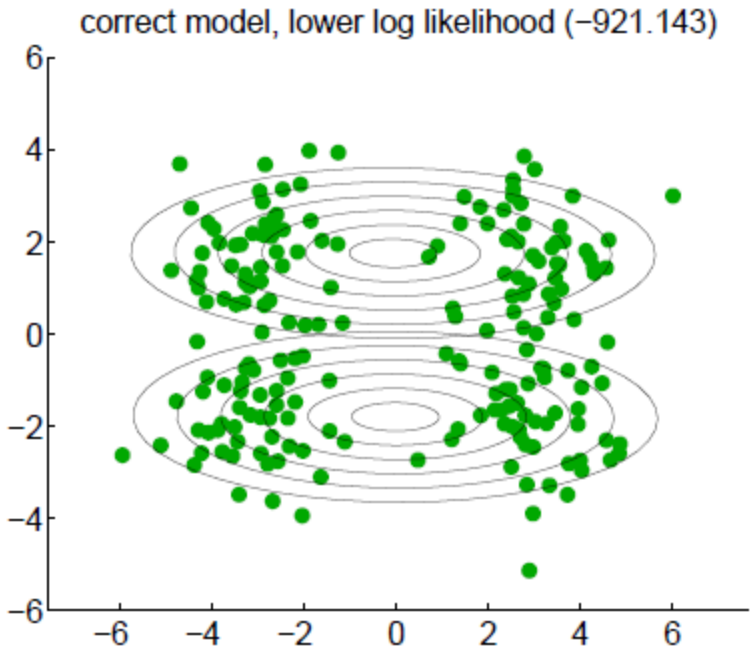
Assumption for GMMs

- **Assumption**: the data actually comes from the mixture model, where the number of components, prior $p(y)$, and conditional $p(\mathbf{x}|y)$ are all correct.
  - When the assumption is wrong:



Assumption for GMMs





Assumption for GMMs

Heuristics to lessen the danger

- Carefully construct the generative model, e.g., multiple Gaussian distributions per class

- Down-weight the unlabeled data ($\lambda < 1$)

$$\log p(X_l, Y_l, X_u|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(x_i|y_i,\theta) + \lambda \sum_{i=l+1}^{l+u} \log \left(\sum_{y=1}^{2} p(y|\theta)p(x_i|y,\theta) \right)$$

Related: Cluster and Label

Input: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)$, $\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}$,
a clustering algorithm $\mathcal{A}$, a supervised learning algorithm $\mathcal{L}$

1. Cluster $\mathbf{x}_1, \ldots, \mathbf{x}_{l+u}$ using $\mathcal{A}$.
2. For each cluster, let S be the labeled instances in it:
3. Learn a supervised predictor from S: $f_S = \mathcal{L}(S)$.
4. Apply f_S to all unlabeled instances in this cluster.

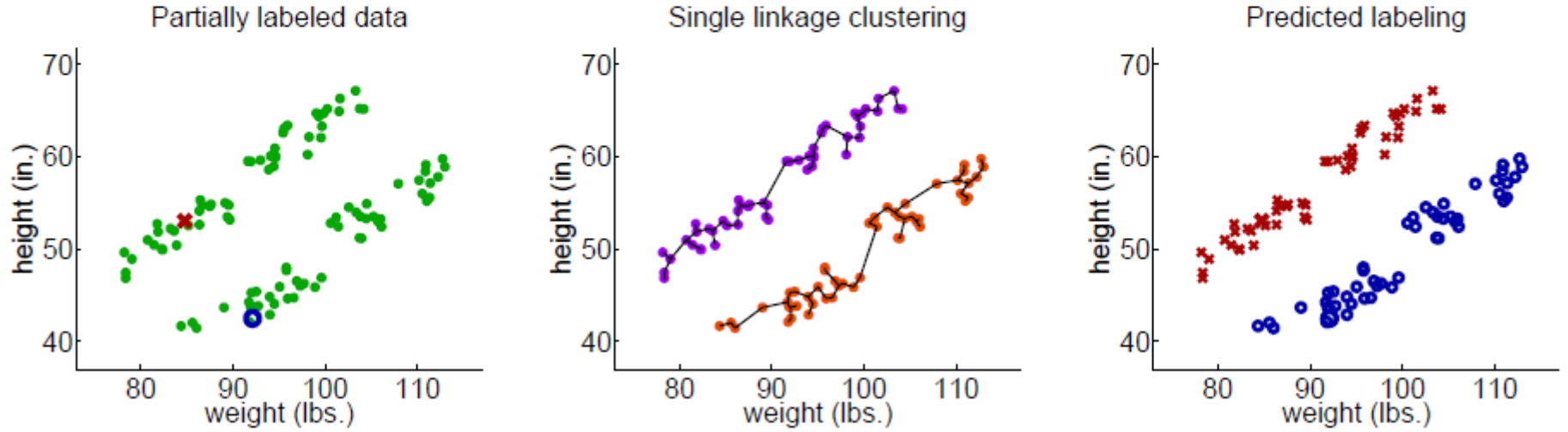
Output: labels on unlabeled data $y_{l+1}, \ldots, y_{l+u}$.

But again: **SSL sensitive to assumptions**—in this case, that the clusters coincide with decision boundaries. If this assumption is incorrect, the results can be poor.

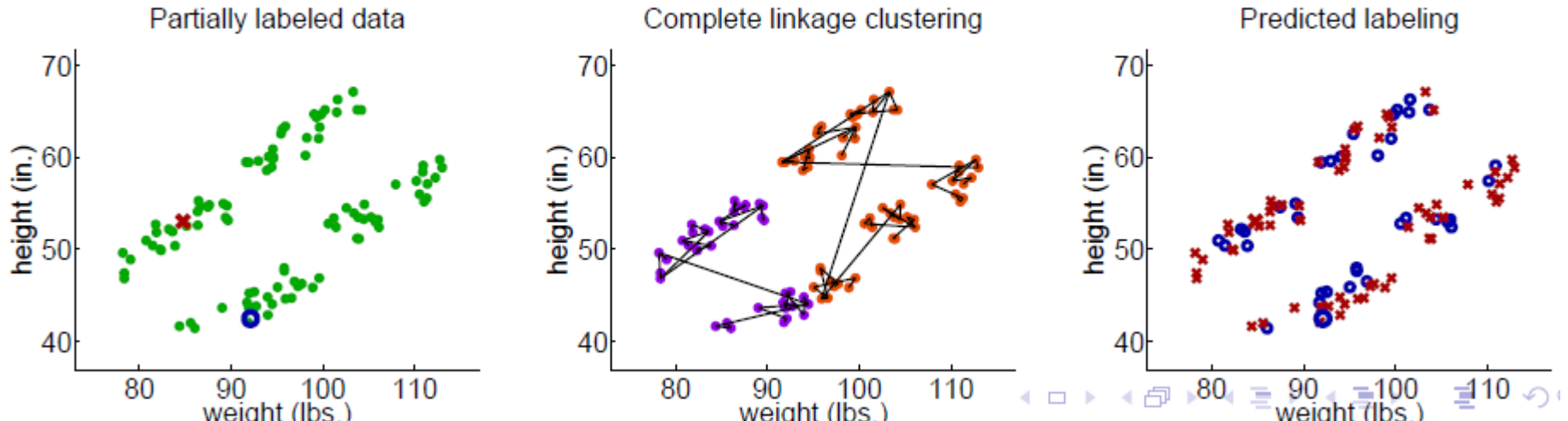
Cluster-and-label: now it works, now it doesn't

Example: $\mathcal{A}$=Hierarchical Clustering, $\mathcal{L}$=majority vote.

single linkage



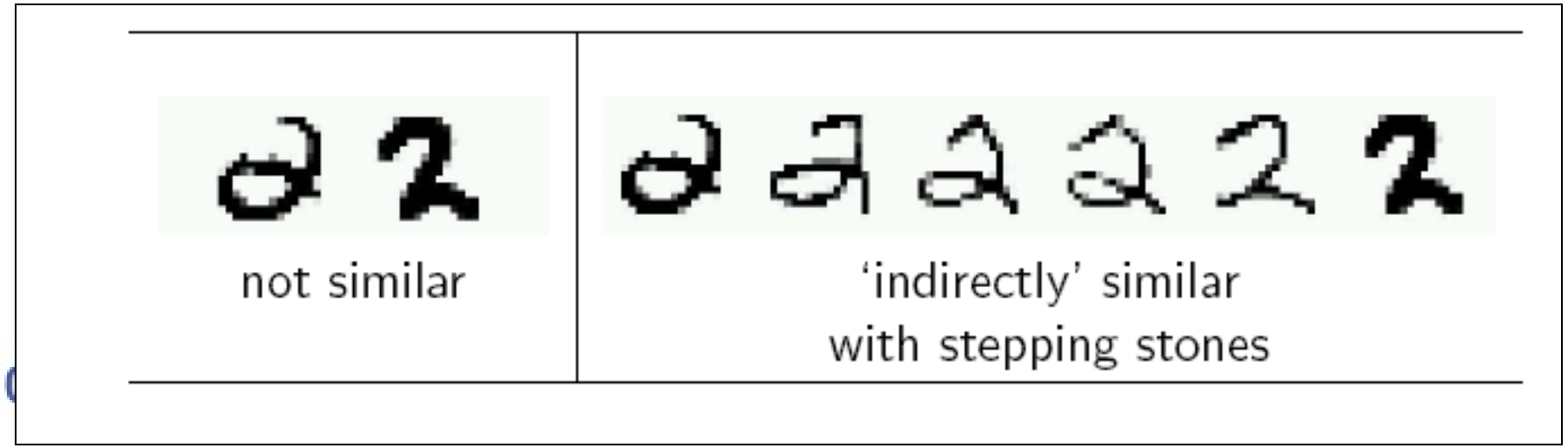
complete linkage



Label Propagation (Graph-Based Methods)

Label Propagation(Graph-based methods)

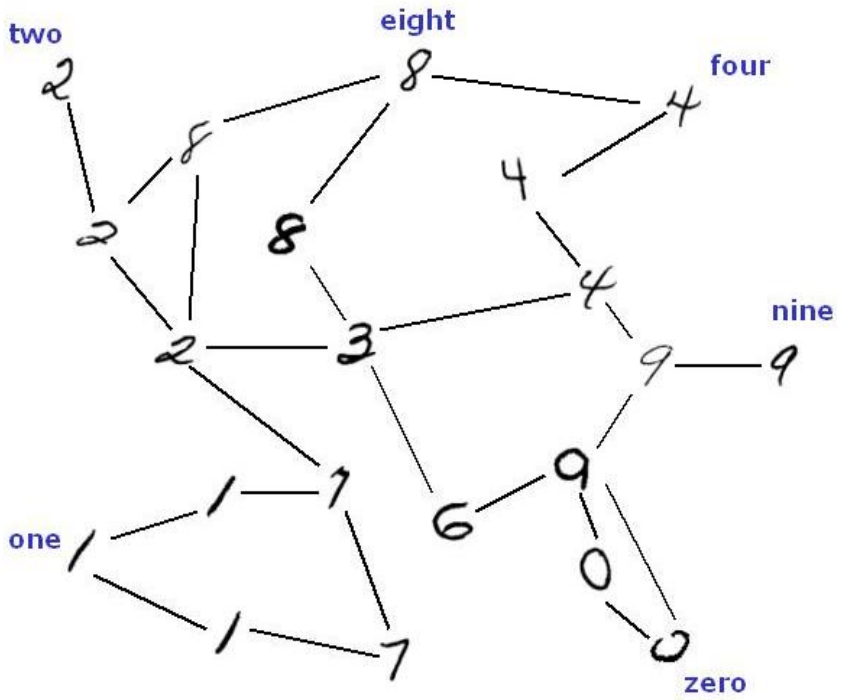
- Suppose that very similar examples probably have the same label
- If you have a lot of labeled data, this suggests a Nearest-Neighbor type of algorithm
- If you have a lot of unlabeled data, perhaps can use them as "stepping stones"



Label Propagation(Graph-based methods)

- Idea: construct a graph with edges between very similar examples.

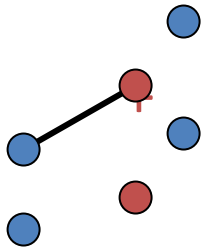
- Unlabeled data can help “glue” the objects of the same class together.

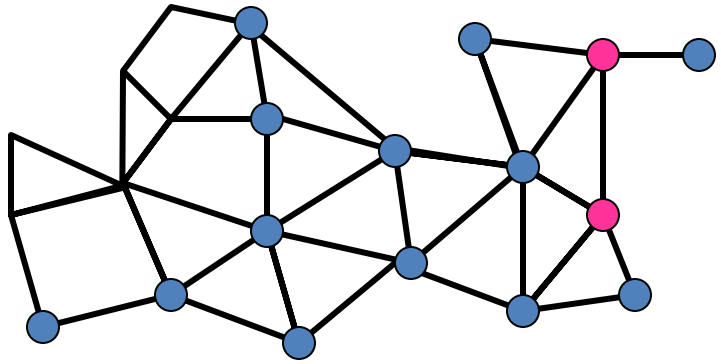


Label Propagation(Graph-based methods)

Suppose just two labels: 0 & 1. Solve for labels f(x) for unlabeled examples x to minimize:

- Label propagation: average of neighbor labels
- Minimum cut $\sum_{e=(u,v)} |f(u)-f(v)|$
- Minimum "soft-cut" $\sum_{e=(u,v)} (f(u)-f(v))^2$



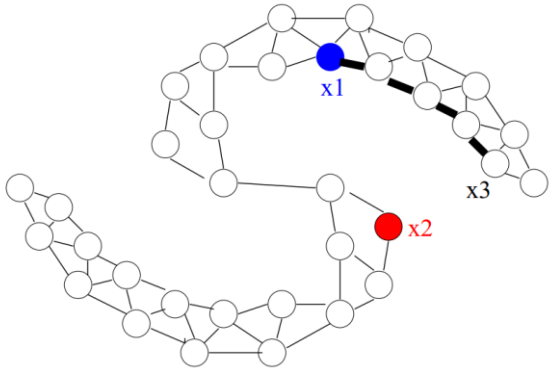


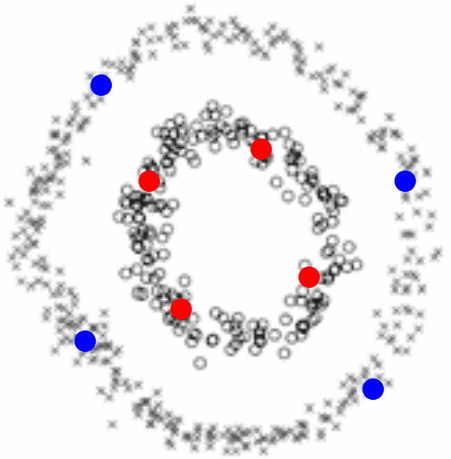
Graph Regularization

Similarity Graphs: Model local neighborhood relations between data points

- Nodes: $X_l \cup X_u$
- Edges: similarity weights computed from features, e.g.,
  - k-nearest-neighbor graph, unweighted (0, 1 weights)
  - fully connected graph, weight decays with distance
    $w_{ij} = \exp\left(-\|x_i - x_j\|^2/\sigma^2\right)$
  - ϵ-radius graph

Assumption:
Nodes connected by heavy edges tend to have similar label



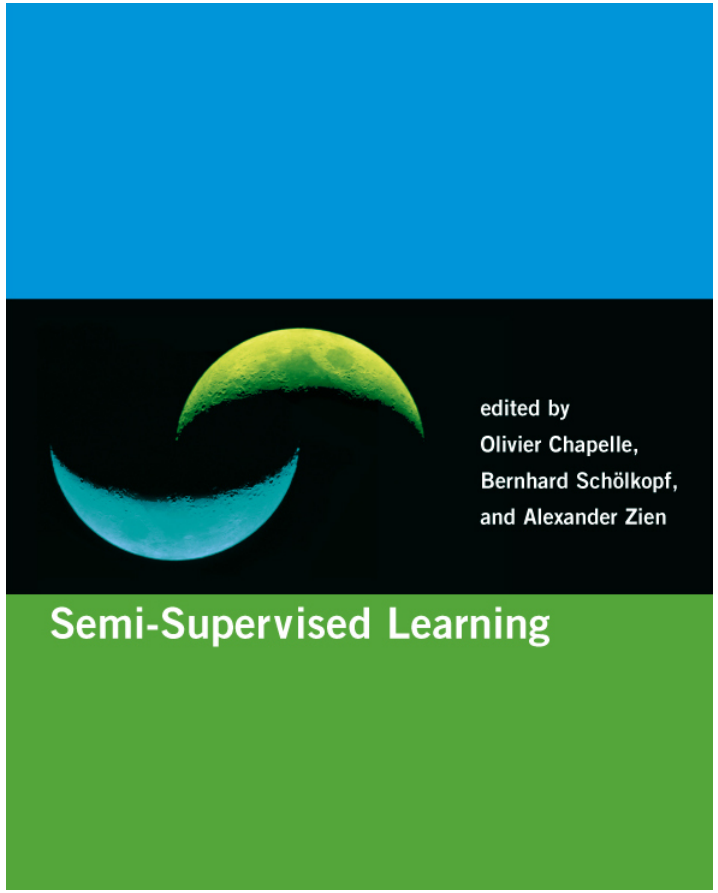


Graph Regularization

If data points i and j are similar (i.e. weight w_{ij} is large), then their labels are similar $f_i = f_j$

$$\min_f \underbrace{\sum_{i \in l} (y_i - f_i)^2}_{\text{Loss on labeled data (mean square,0-1)}} + \lambda \underbrace{\sum_{i,j \in l,u} w_{ij}(f_i - f_j)^2}_{\text{Graph based smoothness prior on labeled and unlabeled data}}$$

SSL Book. http://www.kyb.tuebingen.mpg.de/ssl-book/



- MIT Press, Sept. 2006
- edited by B. Sch¨olkopf, O. Chapelle, A. Zien
- contains many state-of-art algorithms by top researchers
- extensive SSL benchmark
- online material:
  - sample chapters
  - benchmark data
  - more information

Xiaojin Zhu. Semi-Supervised Learning Literature Survey. TR 1530, U. Wisconsin.

Co-Training

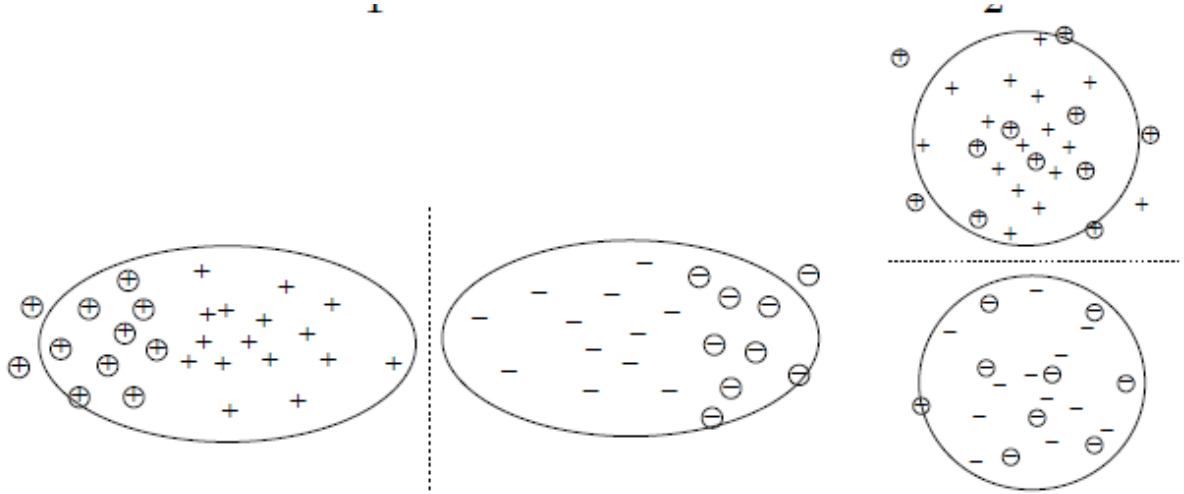
Co-Training using Redundant Features

- 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

Co-training

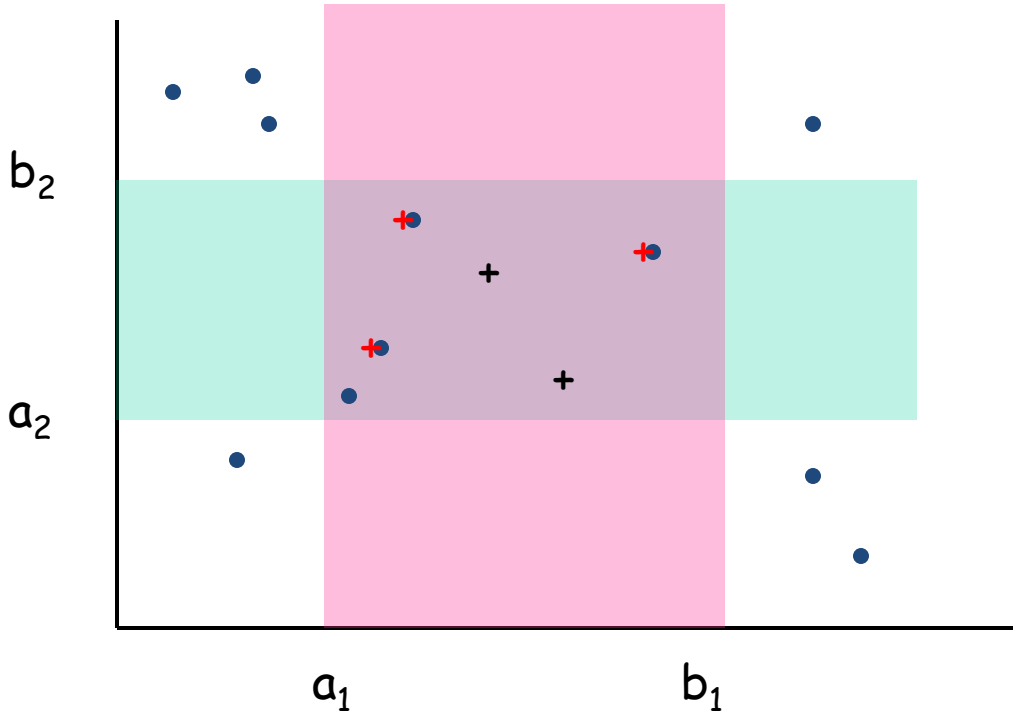
Assumptions

- feature split $x = [x^{(1)}; x^{(2)}]$ exists
- $x^{(1)}$ or $x^{(2)}$ alone is sufficient to train a good classifier
- $x^{(1)}$ and $x^{(2)}$ are conditionally independent given the class



Toy example: intervals

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



Classifying webpages: Using text and links [Blum & Mitchell]

Professor Tom Mitchell

Tom Mitchell

E. Fredkin University Professor
Machine Learning Department
School of Computer Science
Carnegie Mellon University

Resume

Tom.Mitchell@cmu.edu, 412 268 2611, GHC 8203
Assistant: Mary Stech, 412 268-6869

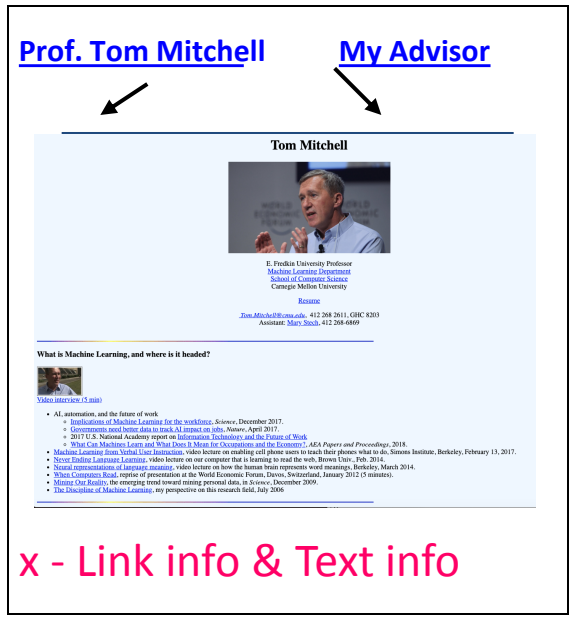
What is Machine Learning, and where is it headed?

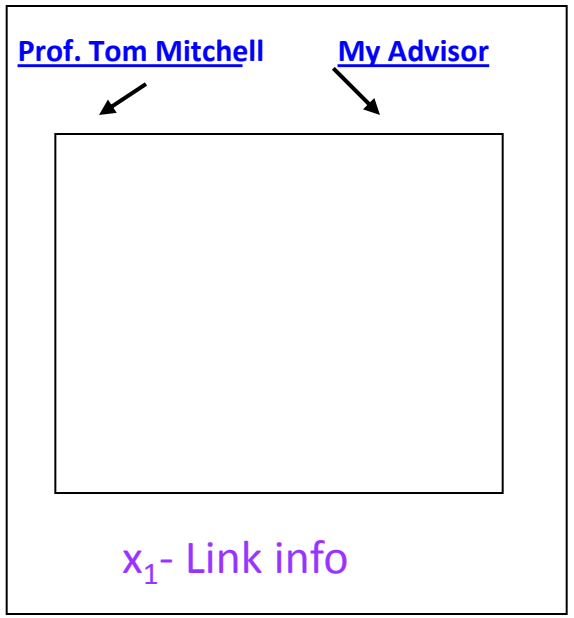
Video interview (5 min)

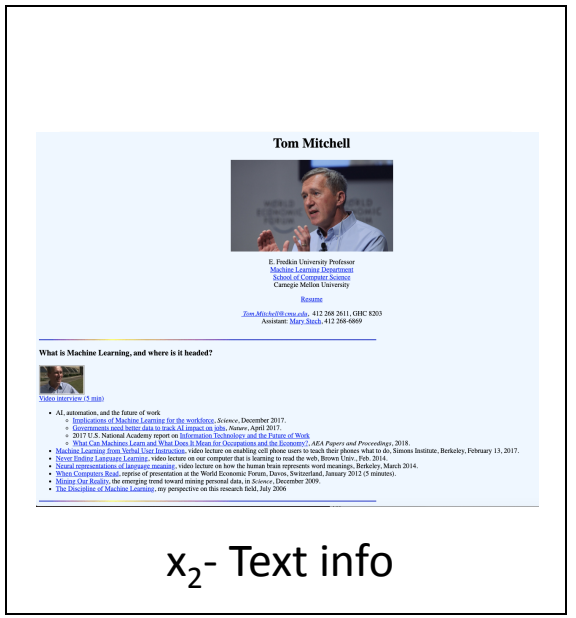
- AI, automation, and the future of work
  - Implications of Machine Learning for the workforce, *Science*, December 2017.
  - Governments need better data to track AI impact on jobs, *Nature*, April 2017.
  - 2017 U.S. National Academy report on Information Technology and the Future of Work
  - What Can Machines Learn and What Does It Mean for Occupations and the Economy?, *AEA Papers and Proceedings*, 2018.
- Machine Learning from Verbal User Instruction, video lecture on enabling cell phone users to teach their phones what to do, Simons Institute, Berkeley, February 13, 2017.
- Never Ending Language Learning, video lecture on our computer that is learning to read the web, Brown Univ., Feb. 2014.
- Neural representations of language meaning, video lecture on how the human brain represents word meanings, Berkeley, March 2014.
- When Computers Read, reprise of presentation at the World Economic Forum, Davos, Switzerland, January 2012 (5 minutes).
- Mining Our Reality, the emerging trend toward mining personal data, in *Science*, December 2009.
- The Discipline of Machine Learning, my perspective on this research field, July 2006

Co-training

- 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.

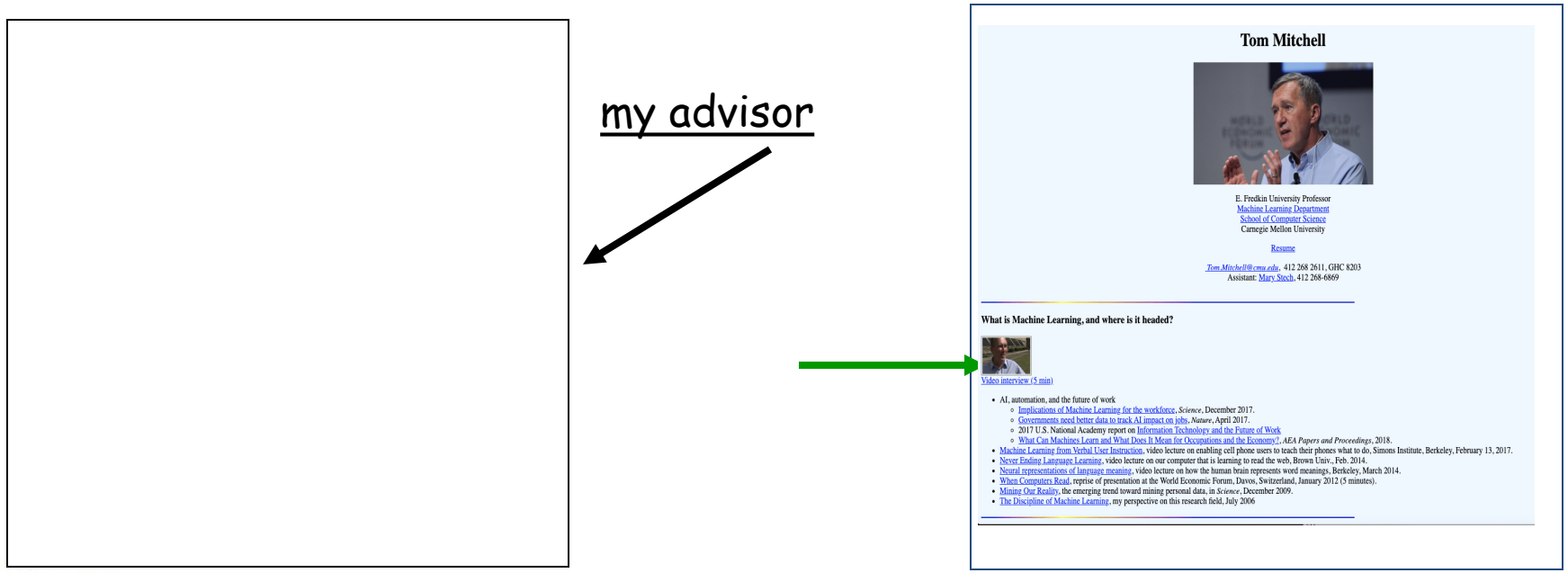






Co-training

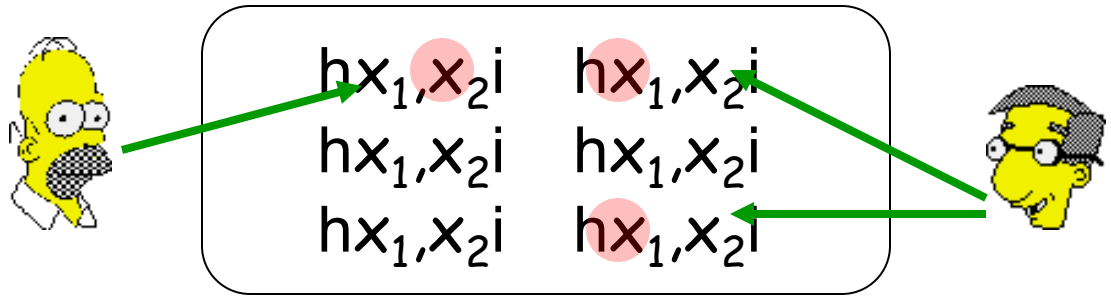
- Idea: Use small labeled sample to learn initial rules.
  - E.g., “my advisor” pointing to a page is a good indicator it is a faculty home page.
  - E.g., “I am teaching” on a page is a good indicator it is a faculty home page.



Co-training

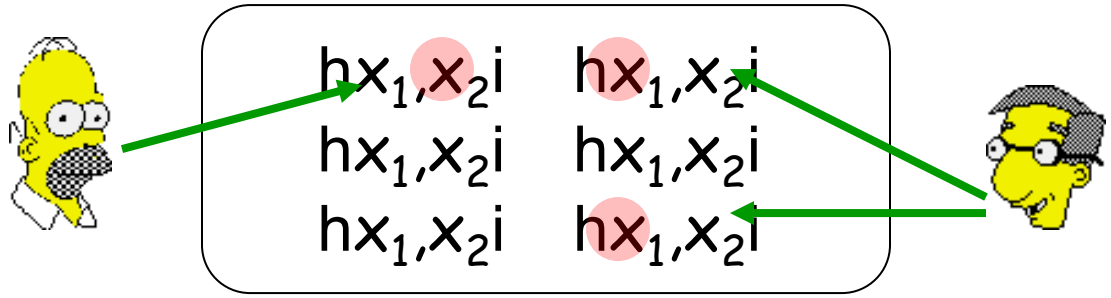
- Then look for unlabeled examples where one rule is confident and the other is not. Have it label the example for the other.

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



Co-training

- Setting is each example $x = (x_1, x_2)$, 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.



Co-training Algorithm

Co-training (Blum & Mitchell, 1998) (Mitchell, 1999) assumes that

(i) features can be split into two sets;

(ii) each sub-feature set is sufficient to train a good classifier.

- Initially two separate classifiers are trained with the labeled data, on the two sub-feature sets respectively.
- Each classifier then classifies the unlabeled data, and 'teaches' the other classifier with the few unlabeled examples (and the predicted labels) they feel most confident.
- Each classifier is retrained with the additional training examples given by the other classifier, and the process repeats.

Co-training Algorithm

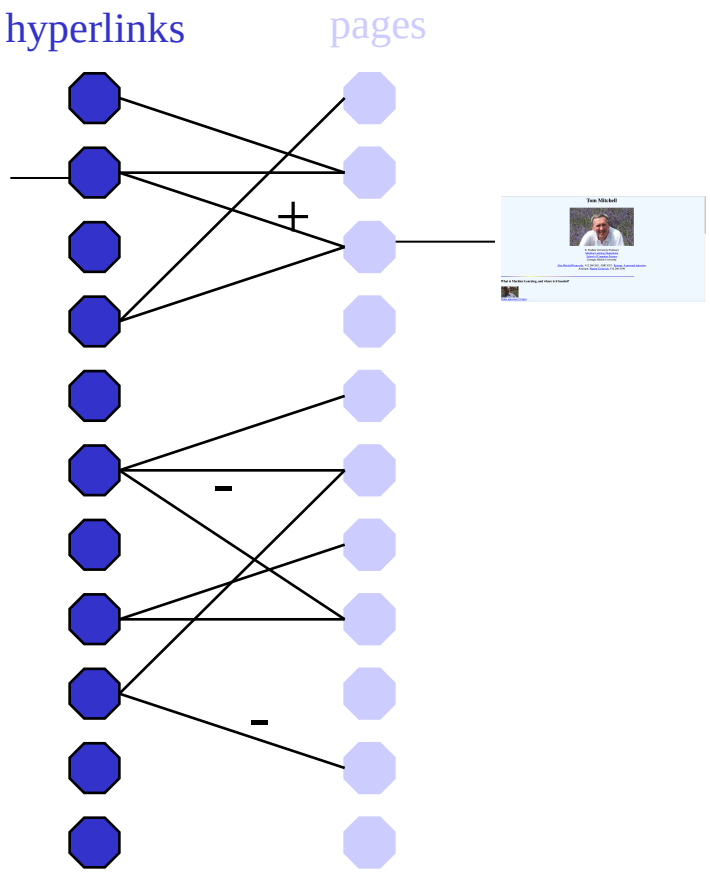
Blum & Mitchell'98

Input: labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$
each instance has two views $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}]$,
and a learning speed k.

1. let $L_1 = L_2 = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$.
2. Repeat until unlabeled data is used up:
3. Train view-1 $f^{(1)}$ from L_1, view-2 $f^{(2)}$ from L_2.
4. Classify unlabeled data with $f^{(1)}$ and $f^{(2)}$ separately.
5. Add $f^{(1)}$'s top k most-confident predictions $(\mathbf{x}, f^{(1)}(\mathbf{x}))$ to L_2.
   Add $f^{(2)}$'s top k most-confident predictions $(\mathbf{x}, f^{(2)}(\mathbf{x}))$ to L_1.
   Remove these from the unlabeled data.

Co-Training Rote Learner

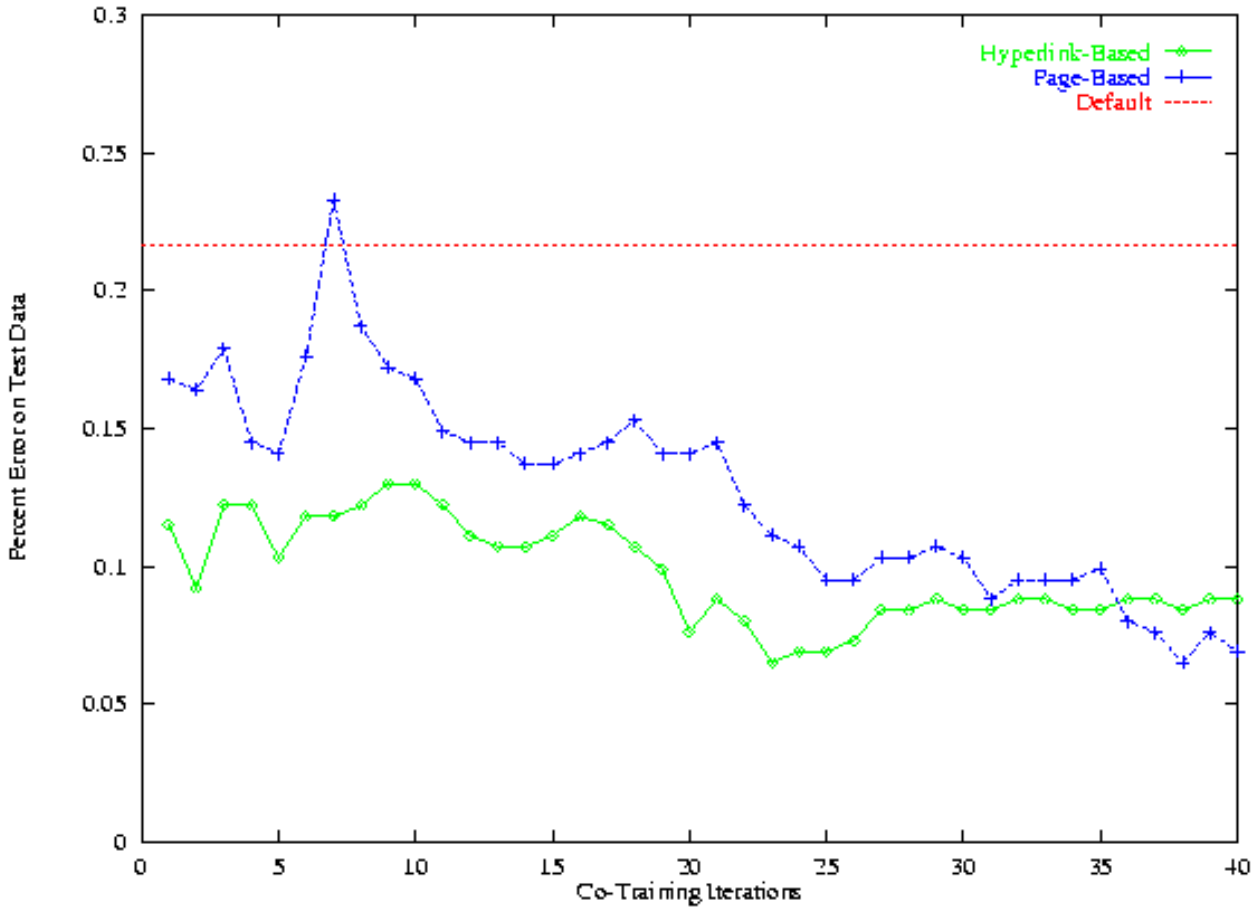
- For links: Use text of page / link pointing to the page of interest
- For pages: Use actual text of the page



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:



The Semi-Supervised SVM (S3VM)

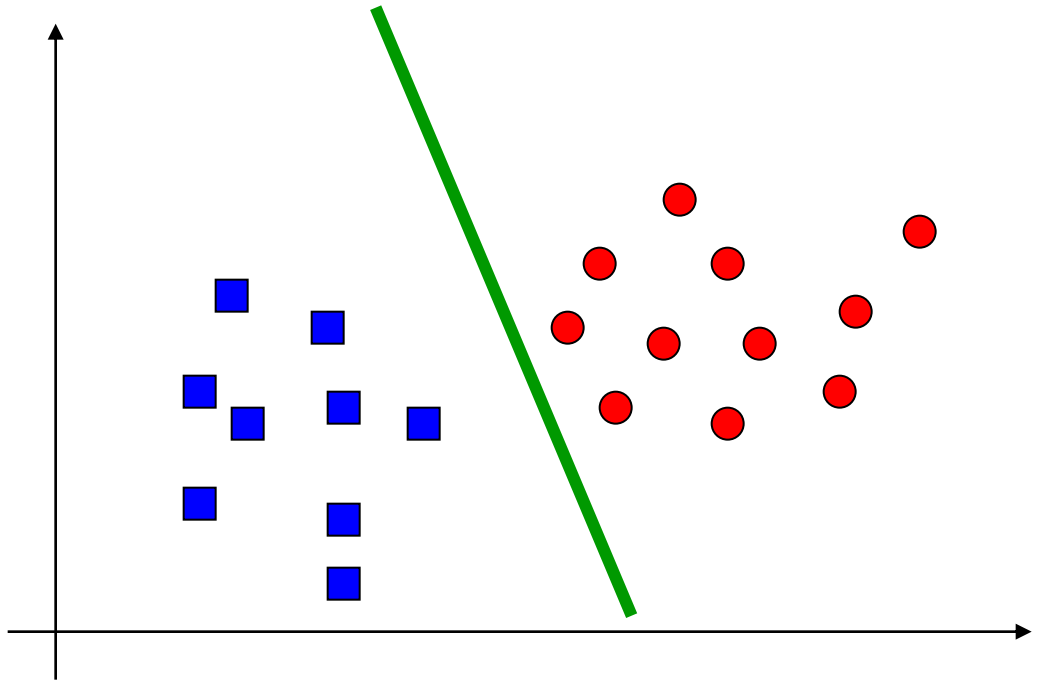
Linearly separable classes

There is a **hyperplane** that separates training instances with no error

Hyperplane:

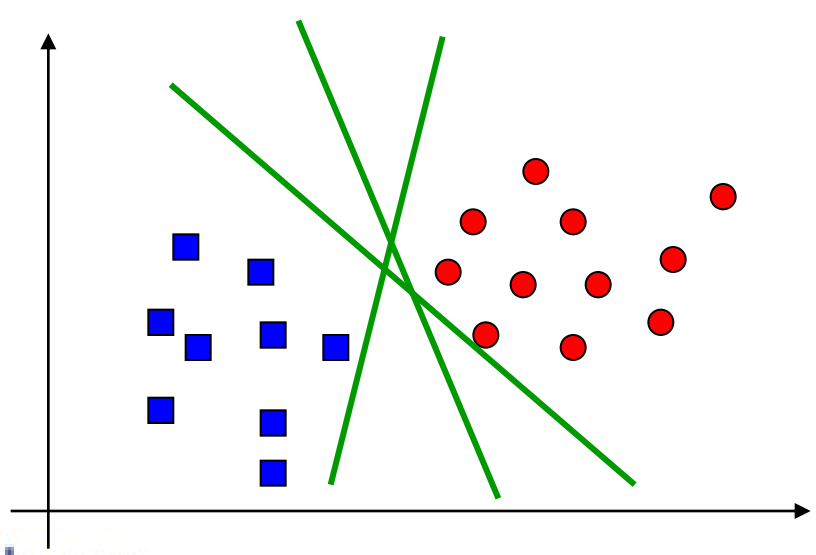
$$\mathbf{w}^T\mathbf{x} + w_0 = 0$$

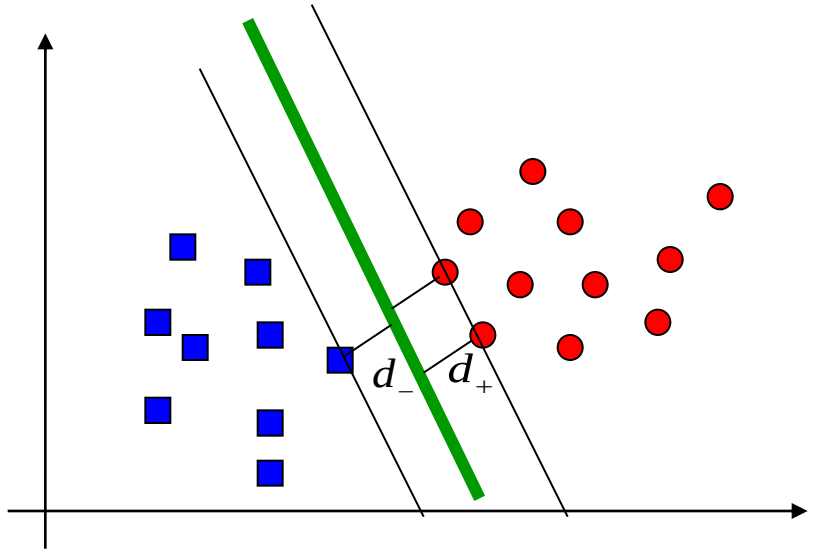
Class (+1) $\mathbf{w}^T\mathbf{x} + w_0 > 0$
Class (-1) $\mathbf{w}^T\mathbf{x} + w_0 < 0$



Optimal separating hyperplane

- There are multiple hyperplanes that separate the data points
  - Which one to choose?
- **Maximum margin** choice: maximum distance of $d_+ + d_-$
  - where d_+ s the shortest distance of a positive example from the hyperplane (similarly d_- for negative examples)

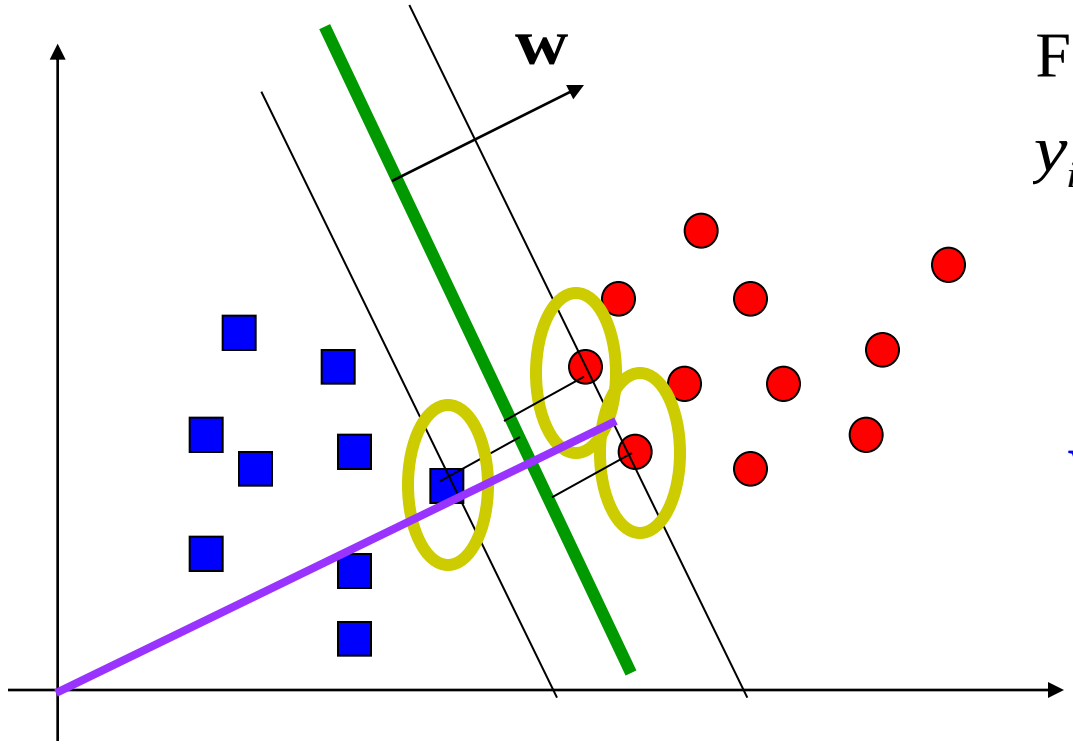




Finding the maximum margin hyperplane

- **Geometrical margin:**
  - measures the distance of a point **x** from the hyperplane

    w - normal to the hyperplane $\|..\|_{L2}$ - Euclidean norm



For points satisfying:

$$y_i(\mathbf{w}^T\mathbf{x}_i + w_0) - 1 = 0$$

Width of the margin:

$$d_+ + d_- = \frac{2}{\|\mathbf{w}\|_{L2}}$$

Maximum margin hyperplane

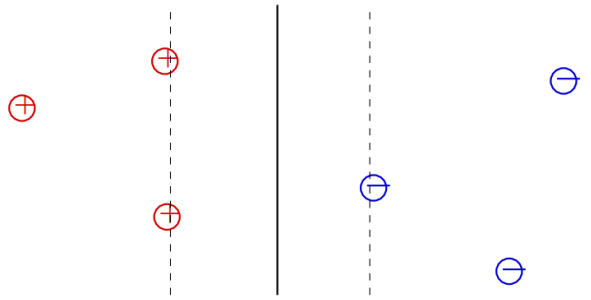
- **We want to maximize** $d_+ + d_- = \frac{2}{\|\mathbf{w}\|_{L2}}$

  – But we also need to enforce the constraints on points:

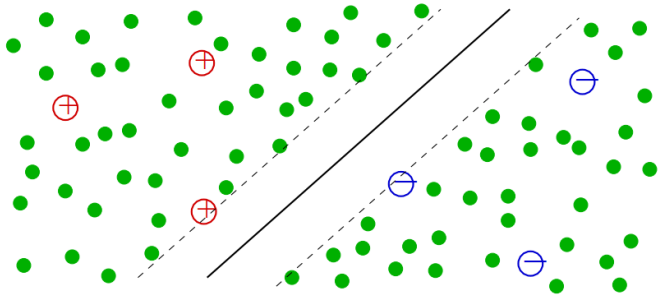
$$\left[y_i(\mathbf{w}^T\mathbf{x} + w_0) - 1\right] \geq 0$$

Semi-Supervised SVMs

SVMs



Semi-supervised SVMs (S3VMs) = Transductive SVMs (TSVMs)

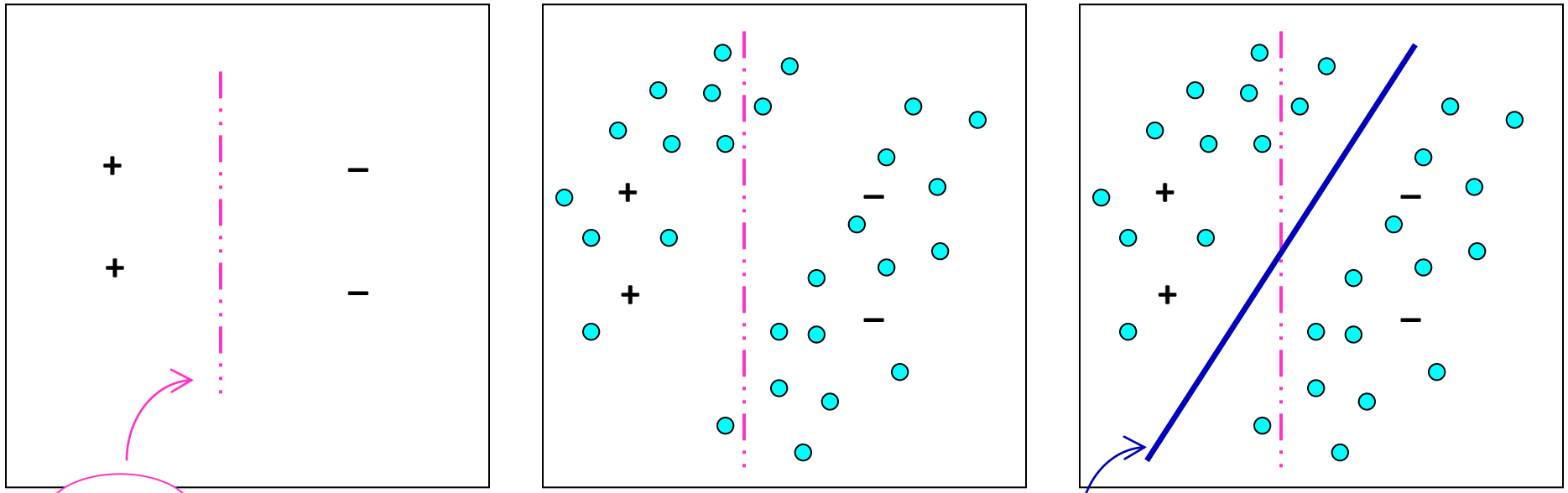


Assumption: Unlabeled data from different classes are separated with large margin.

Semi-Supervised SVM (S^{3}VM)

- Suppose we believe decision boundary goes through low density regions of the space/large margin.

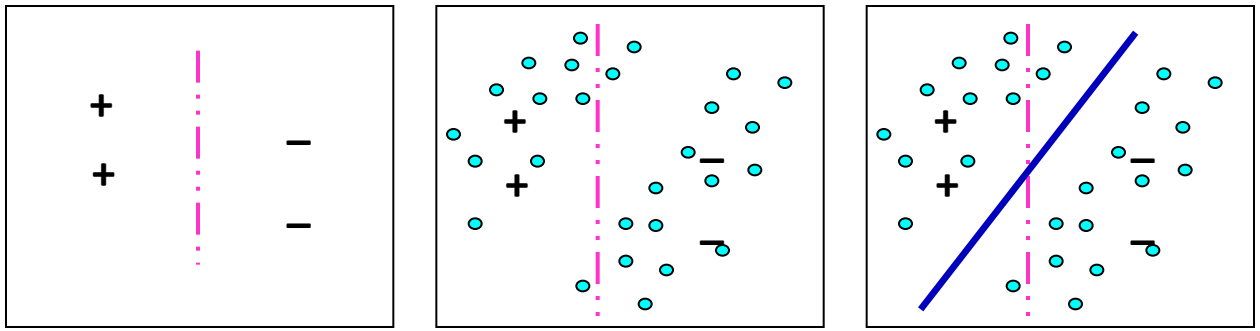
- Aim for classfiers with large margin wrt labeled and unlabeled data. (L+U)



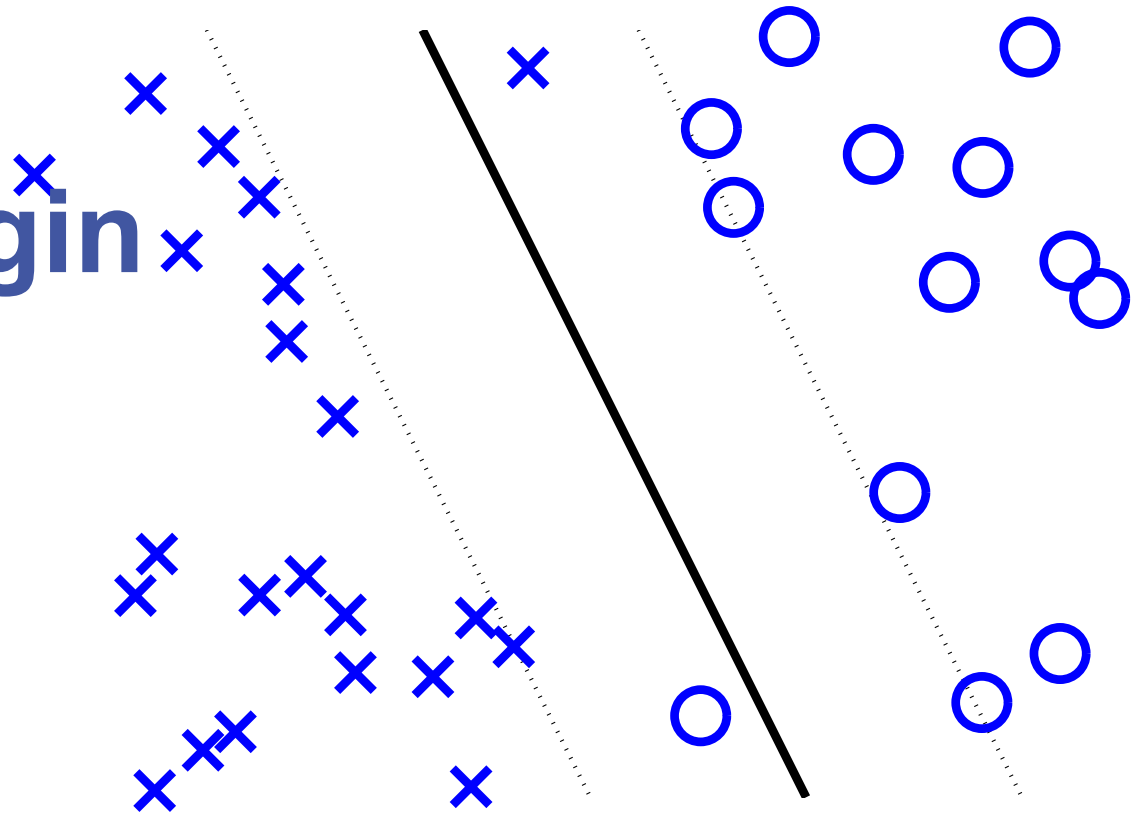
Semi-Supervised SVM (S^3VM)

- 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.
  - Or, branch-and-bound, other methods (Chapelle etal06)

- Quite successful on text data.

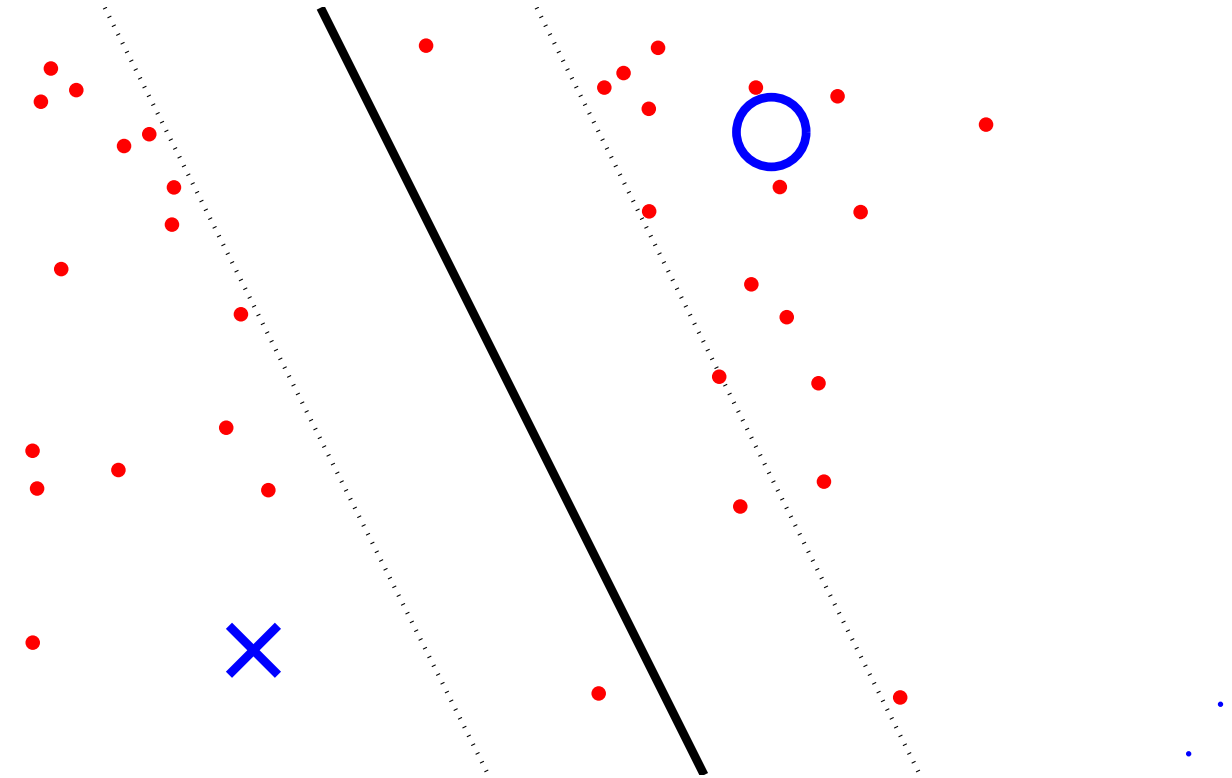


soft margin SVM



$$\min_{\mathbf{w},b,(\xi_k)} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_i \xi_i \qquad s.t. \qquad \xi_i \geq 0 \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$$

soft margin
S^{3}VM



$$\min_{\mathbf{w},b,(y_j),(\xi_k)} \quad \begin{array}{l} \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle \\ +C\sum_i \xi_i \\ +C^*\sum_j \xi_j \end{array} \quad s.t. \quad \begin{array}{c} \xi_i \geq 0 \quad \xi_j \geq 0 \\ y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\ y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b) \geq 1 - \xi_j \end{array}$$

Supervised Support Vector Machine (SVM)

$$\min_{\mathbf{w},b,(\xi_k)} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_i \xi_i \quad s.t. \quad \xi_i \geq 0, \; y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$$

maximize margin on (labeled) points

convex optimization problem (QP, quadratic programming)

Semi-Supervised Support Vector Machine (S3VM)

$$\min_{\mathbf{w},b,(y_j),(\xi_k)} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_i \xi_i + C^* \sum_j \xi_j \quad s.t. \quad \xi_i \geq 0, \; \xi_j \geq 0, \; y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \; y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b) \geq 1 - \xi_j$$

maximize margin on labeled and unlabeled points

also QP?

$$\min_{\mathbf{w},b,(y_j),(\xi_k)} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C\sum_i \xi_i + C^*\sum_j \xi_j$$

$$s.t. \quad \begin{array}{ll} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i & \xi_i \geq 0 \\ y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b) \geq 1 - \xi_j & \xi_j \geq 0 \end{array}$$

Problem!

- y_j **are discrete!**
- **Combinatorial** task.
- **NP-hard!**

Optimization methods used for S^3VM training

exact:

- Mixed Integer Programming [Bennett, Demiriz; NIPS 1998]
- Branch & Bound [Chapelle, Sindhwani, Keerthi; NIPS 2006]

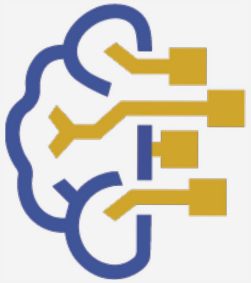
approximative:

- self-labeling heuristic S^3VM^{light} [T. Joachims; ICML 1999]
- gradient descent [Chapelle, Zien; AISTATS 2005]
- CCCP-S^3VM [R. Collobert et al.; ICML 2006]
- contS^3VM [Chapelle et al.; ICML 2006]

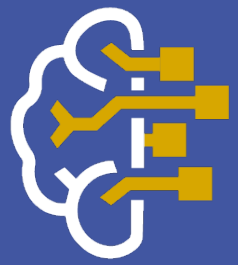
Stay Connected

Dr. Min Chi

Department of Computer Science
mchi@ncsu.edu (919) 515-7825



AI Academy



AI Academy

go.ncsu.edu/aiacademy

NC STATE UNIVERSITY