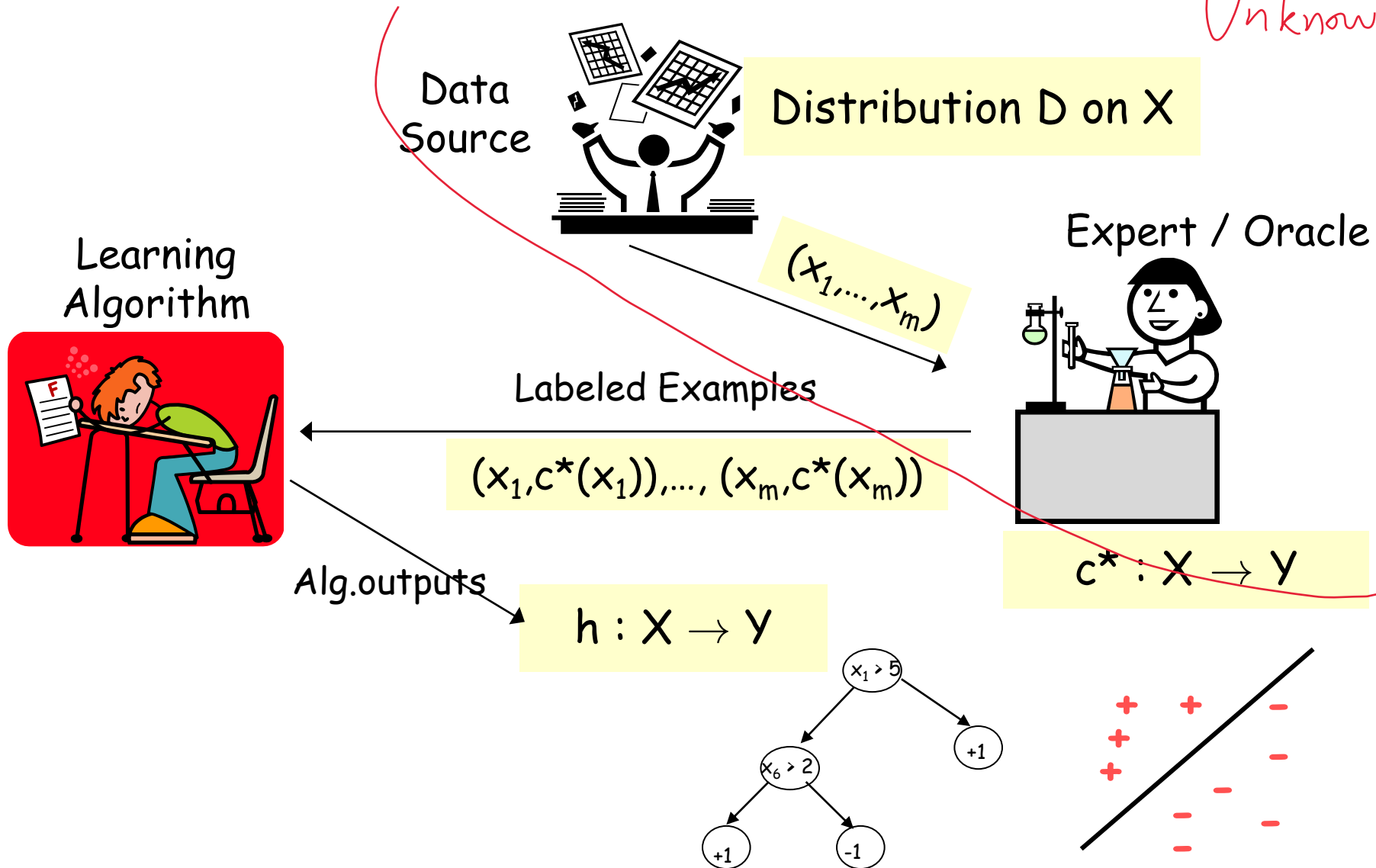# Semi-Supervised Learning

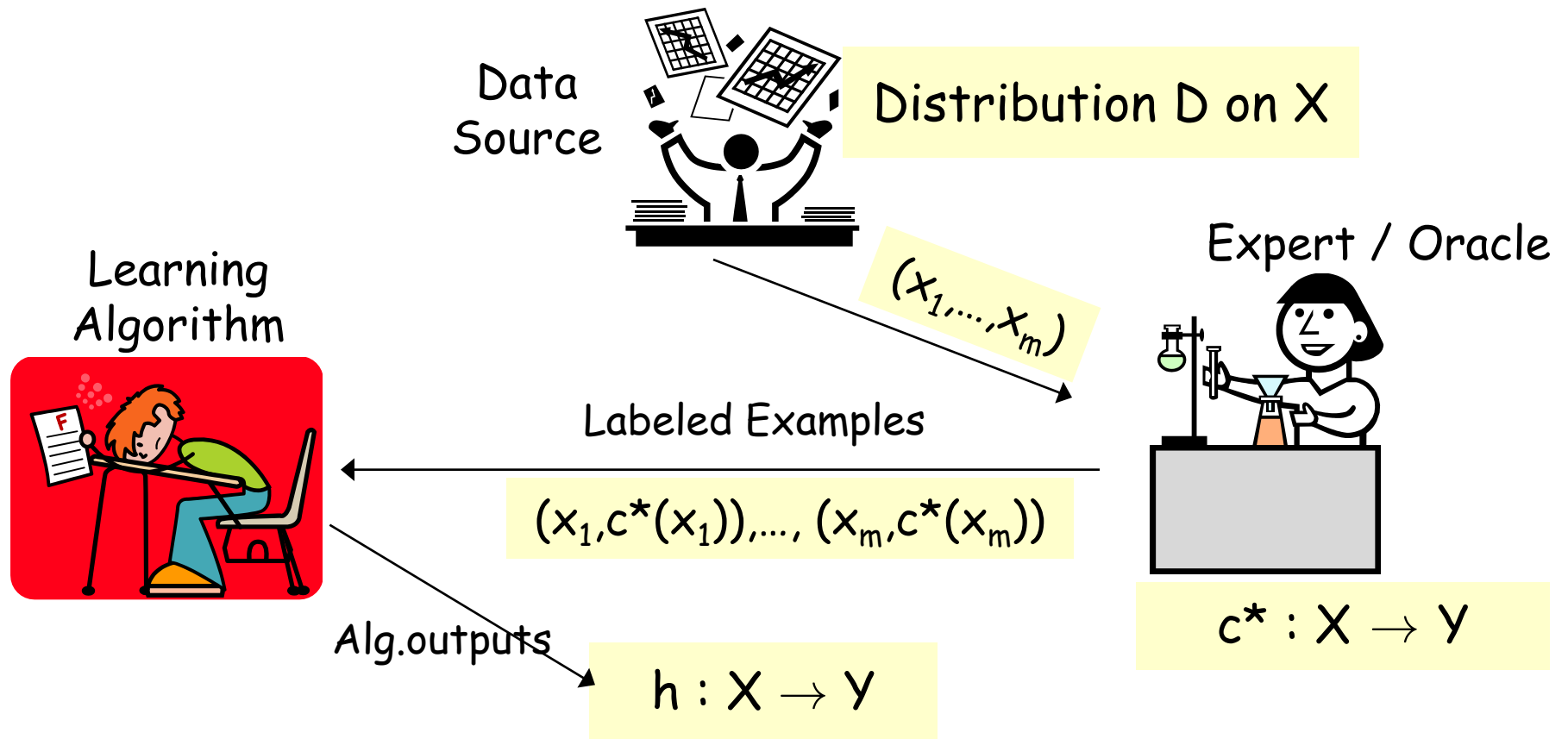## Maria-Florina Balcan
03/30/2015

Readings:

- Semi-Supervised Learning. Encyclopedia of Machine Learning. Jerry Zhu, 2010

- Combining Labeled and Unlabeled Data with Co-Training. Avrim Blum, Tom Mitchell. COLT 1998.

# Fully Supervised Learning

Unknown

Data Source

Distribution D on X

Expert / Oracle

$(x_1,...,x_m)$

Learning Algorithm

Labeled Examples

$(x_1,c^*(x_1)),..., (x_m,c^*(x_m))$

$c^* : X \rightarrow Y$

Alg. outputs

$h : X \rightarrow Y$

$x_1 > 5$

$x_6 > 2$

+1

+1

-1

+  +  -
+      -
+   -   -
     -   -

# Fully Supervised Learning

Data Source

Distribution D on X

$(x_1,...,x_m)$

Expert / Oracle

Learning Algorithm

Labeled Examples

$(x_1,c^*(x_1)),..., (x_m,c^*(x_m))$

$c^* : X \rightarrow Y$

Alg.outputs

$h : X \rightarrow Y$

$S_l=\{(x_1,y_1), ...,(x_{m_l},y_{m_l})\}$

$x_i$ drawn i.i.d from $D$, $y_i = c^*(x_i)$

**Goal**: h has small error over D.

$$err_D(h) = \Pr_{x \sim D}(h(x) \neq c^*(x))$$

# Two Core Aspects of Supervised Learning

Algorithm Design. How to optimize?

Computation

Automatically generate rules that do well on observed data.

- E.g.: Naïve Bayes, logistic regression, SVM, Adaboost, etc.
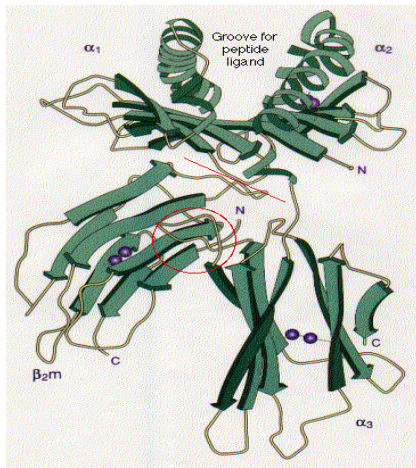
Confidence Bounds, Generalization

(Labeled) Data

Confidence for rule effectiveness on future data.

- VC-dimension, Rademacher complexity, margin based bounds, etc.
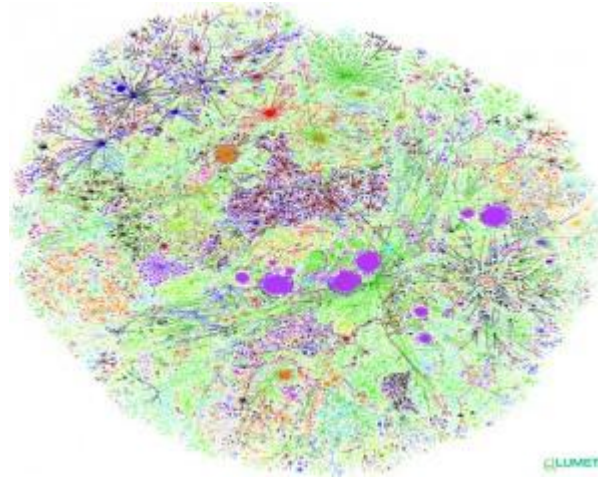
# Classic Paradigm Insufficient Nowadays

Modern applications: **massive amounts** of raw data.

Only *a tiny fraction* can be annotated by human experts.



Protein sequences



Billions of webpages



Images

# Modern ML: New Learning Approaches

Modern applications: massive amounts of raw data.

**Techniques that best utilize data, minimizing need for expert/human intervention.**
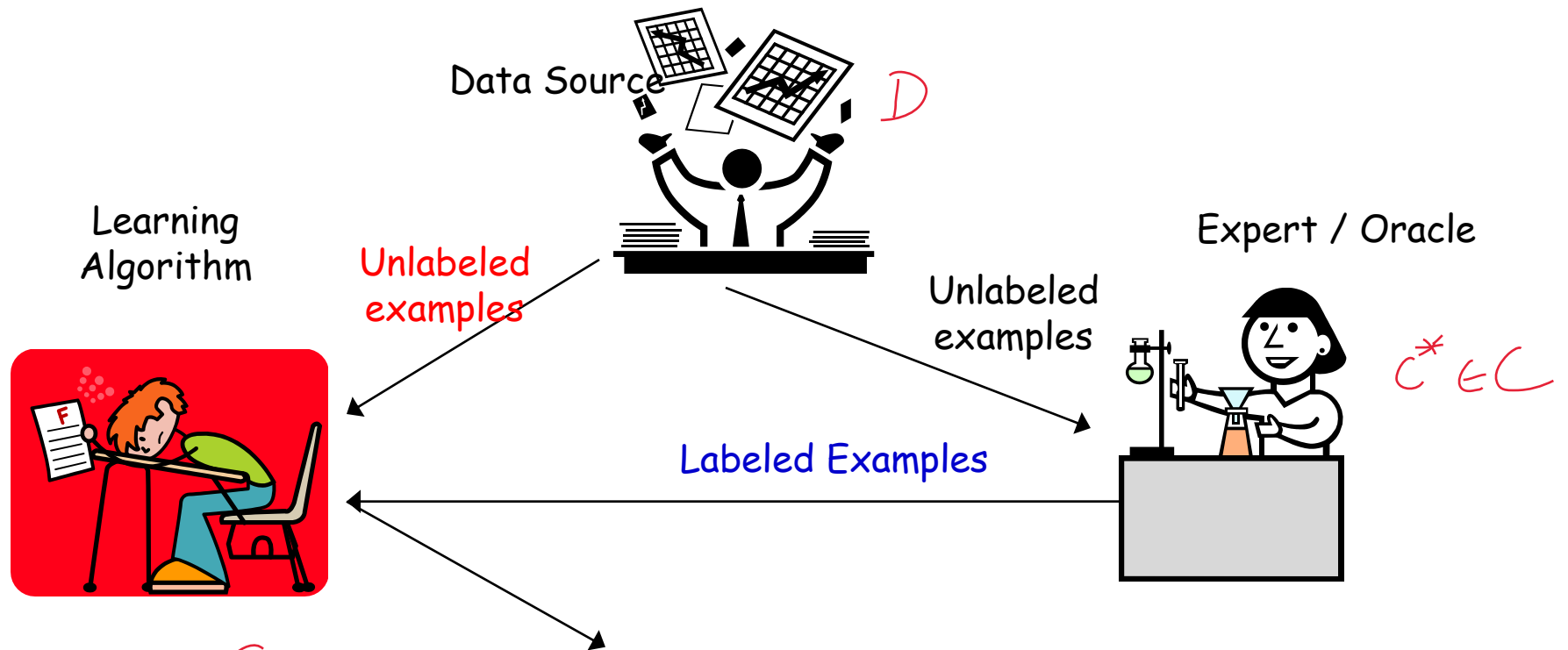
Paradigms where there has been great progress.

- Semi-supervised Learning, (Inter)active Learning.

# Inductive Semi-Supervised Learning

Data Source $D$

Learning Algorithm

Unlabeled examples

Unlabeled examples

Expert / Oracle

$c^* \in C$

Labeled Examples

Algorithm outputs a classifier

$Source$

$S_l = \{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$

$x_i$ drawn i.i.d from $D$, $y_i = c^*(x_i)$

$S_u = \{x_1, \ldots, x_{m_u}\}$ drawn i.i.d from $D$

Target

Transfer Learning

few / one / zero-shot

**Goal**: h has small error over D.

$$err_D(h) = \Pr_{x \sim D}(h(x) \neq c^*(x))$$

Transductive Learning

$$err_{S_u}(h) = \Pr_{x \sim S_u}(h(x) \neq c^*(x))$$

# Semi-supervised Learning

- Major topic of research in ML.

- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:

  - Transductive SVM [Joachims '99]

  - Co-training [Blum & Mitchell '98]

  - Graph-based methods [B&C01], [ZGL03]

Test of time awards at ICML!

Workshops [ICML '03, ICML' 05, …]

Books:
- Semi-Supervised Learning, MIT 2006

  O. Chapelle, B. Scholkopf and A. Zien (eds)

- Introduction to Semi-Supervised Learning, Morgan & Claypool, 2009  Zhu & Goldberg

# Semi-supervised Learning

- Major topic of research in ML.

- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:

  - Transductive SVM [Joachims '99]

  - Co-training [Blum & Mitchell '98]

  - Graph-based methods [B&C01], [ZGL03]

Test of time awards at ICML!

Both wide spread applications and solid foundational understanding!!!

# Semi-supervised Learning

- Major topic of research in ML.

- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:

  - Transductive SVM  [Joachims '99]

  - Co-training  [Blum & Mitchell '98]

  - Graph-based methods  [B&C01], [ZGL03]

Test of time awards at ICML!

Today: discuss these methods.

Very interesting, they all exploit unlabeled data in different, very interesting and creative ways.

**Semi-supervised learning:** no querying. Just have lots of additional unlabeled data.

A bit puzzling; unclear what unlabeled data can do for us…. It is missing the most important info. How can it help us in substantial ways?

Key Insight

Unlabeled data useful if we have beliefs not only about the form of the target, but also about its relationship with the underlying distribution.

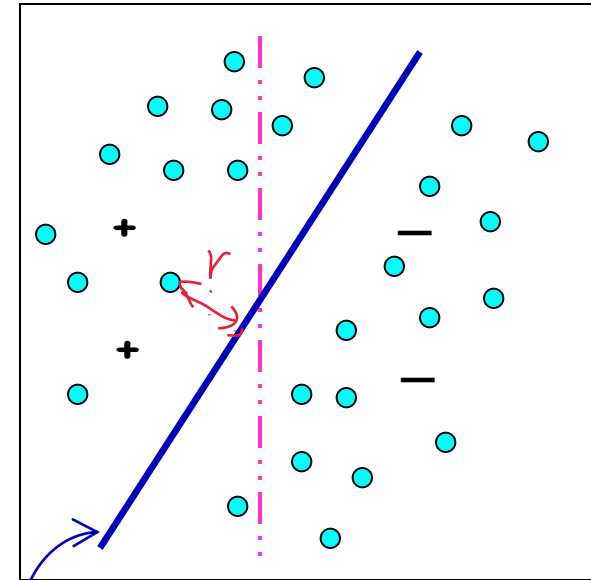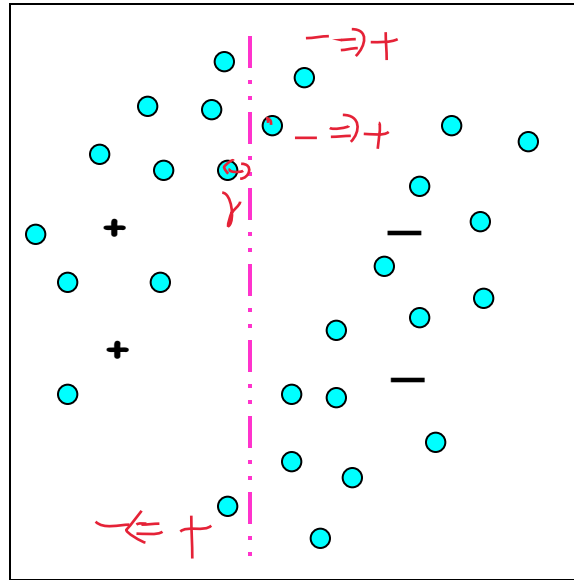# Semi-supervised SVM

[Joachims '99]

# Margins based regularity

Target goes through low density regions (large margin).

- assume we are looking for linear separator
- belief: should exist one with large separation
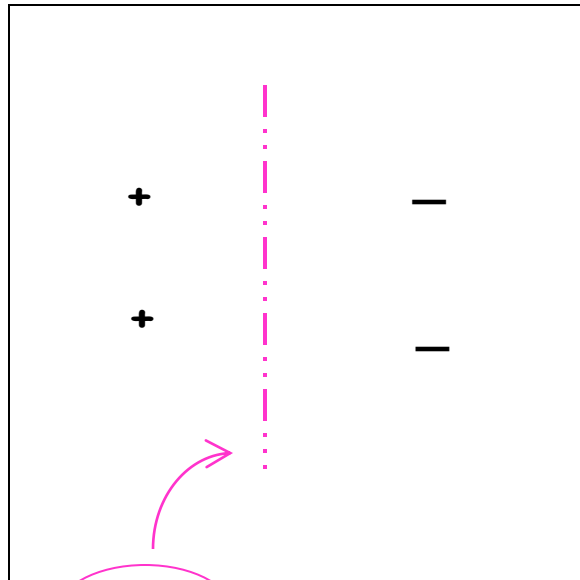


SVM

Labeled data only
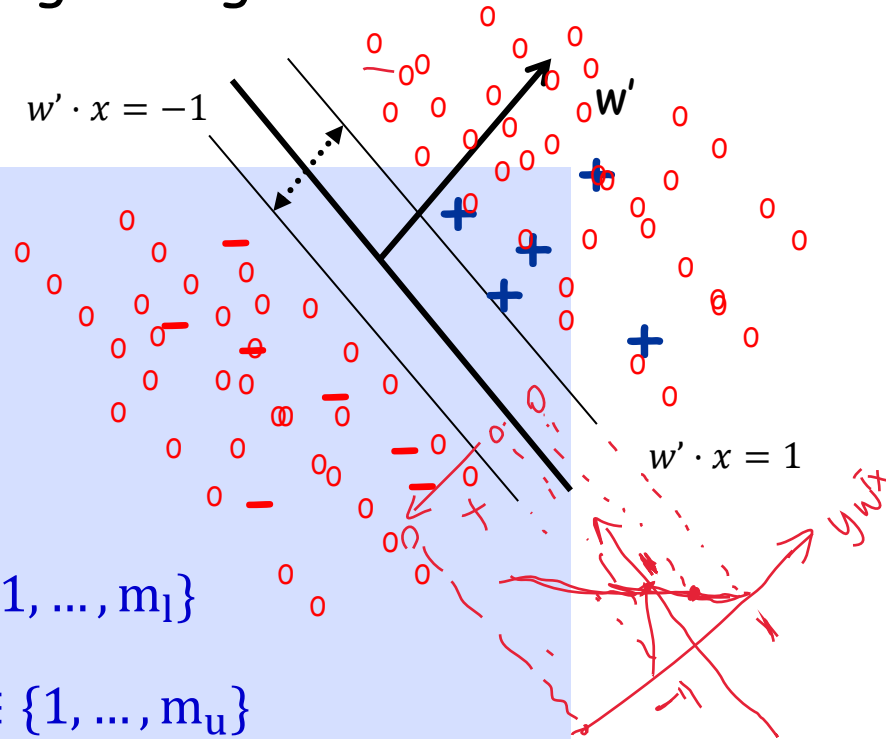
Transductive SVM

# Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]

$w' \cdot x = -1$

$w'$

$w' \cdot x = 1$

**Input**: $S_l = \{(x_1, y_1), \dots, (x_{m_l}, y_{m_l})\}$

$S_u = \{x_1, \dots, x_{m_u}\}$

$\text{argmin}_w \, ||w||^2$ s.t.:

- $y_i \, w \cdot x_i \geq 1$, for all $i \in \{1, \dots, m_l\}$

- $\widehat{y_u} w \cdot x_u \geq 1$, for all $u \in \{1, \dots, m_u\}$

- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \dots, m_u\}$

Find a labeling of the unlabeled sample and $w$ s.t. $w$ separates both labeled and unlabeled data with maximum margin.

# Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data. [Joachims '99]

non-convex

$w' \cdot x = -1$

$C_u \sum\limits_{x_i \in S_u} (1 - |w^T x_i|)_+$

**Input**: $S_l = \{(x_1, y_1), ..., (x_{m_l}, y_{m_l})\}$  $x_i \in S_u$

$S_u = \{x_1, ..., x_{m_u}\}$

$C_u \sum\limits_{x_i \in S_u} (1 - y_i w^T x_i)_+$

$\text{argmin}_w \, ||w||^2 + C \sum_i \xi_i + C_u \sum_u \widehat{\xi_u}$

$y_i \, w \cdot x_i \geq 1 - \xi_i$, for all $i \in \{1, ..., m_l\}$

$\widehat{\xi_m} = (1 - |w^T x|)_+$

$\widehat{y_u} w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, ..., m_u\}$

hat-loss
(sym-hinge)

$\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, ..., m_u\}$

$w' \cdot x = 1$

w'

loss

$y \cdot w^T x$

$\xi_i = (1 - y_i w^T x_i)_+$

Find a labeling of the unlabeled sample and $w$ s.t. $w$ separates both labeled and unlabeled data with maximum margin.

# Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data.

**Input**: $S_l = \{(x_1, y_1), \ldots, (x_{m_l}, y_{m_l})\}$

$S_u = \{x_1, \ldots, x_{m_u}\}$

$$\text{argmin}_w \|w\|^2 + C \sum_i \xi_i + C \sum_u \widehat{\xi_u}$$

$w, \xi, \widehat{y}_u$

- $y_i\, w \cdot x_i \geq 1 - \xi_i$, for all $i \in \{1, \ldots, m_l\}$

- $\widehat{y_u} w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, \ldots, m_u\}$

- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

NP-hard..... Convex only after you guessed the labels... too many possible guesses...

# Transductive Support Vector Machines

Optimize for the separator with large margin wrt labeled and unlabeled data.

Heuristic (Joachims) high level idea:

- First maximize margin over the labeled points

- Use this to give initial labels to unlabeled points based on this separator.

- Try flipping labels of unlabeled points to see if doing so can increase margin

Keep going until no more improvements. Finds a locally-optimal solution.
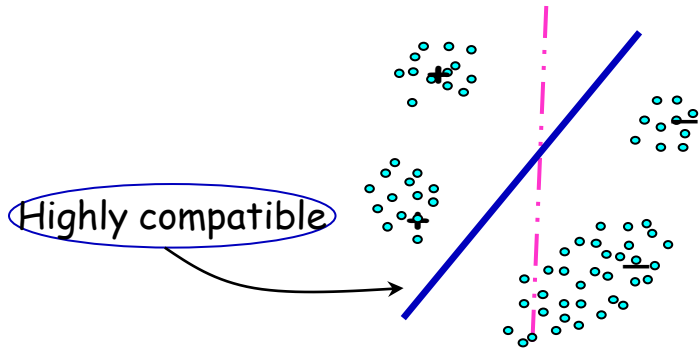
# Experiments [Joachims99]
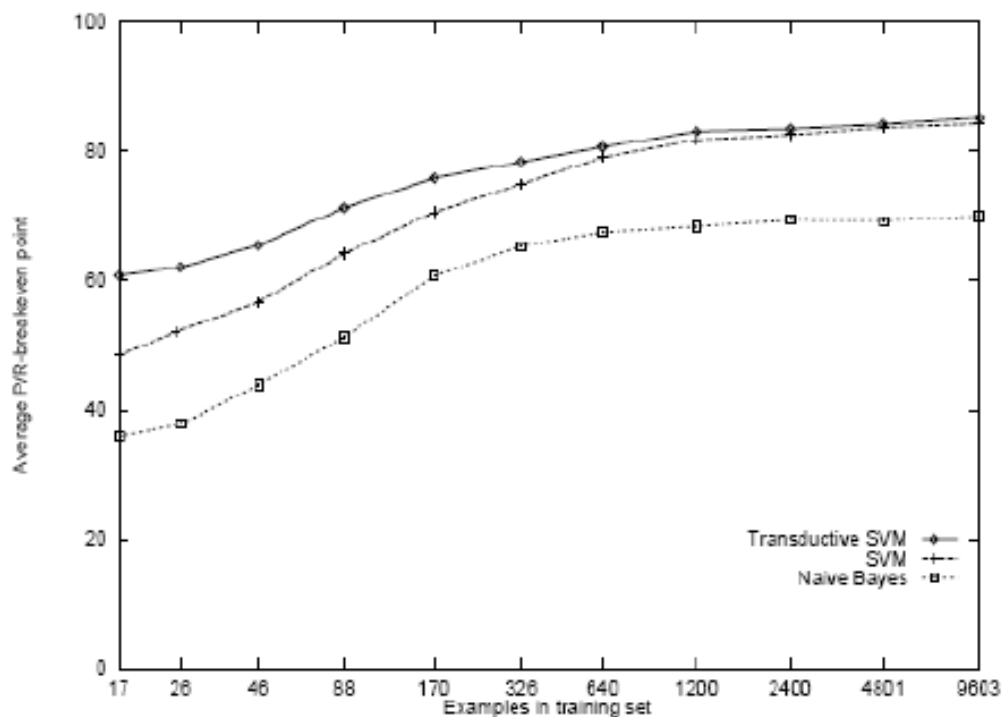


Figure 6: Average P/R-breakeven point on the Reuters dataset for different training set sizes and a test set size of 3,299.

# Transductive Support Vector Machines

## Helpful distribution



Highly compatible

## Non-helpful distributions

### Margin not satisfied
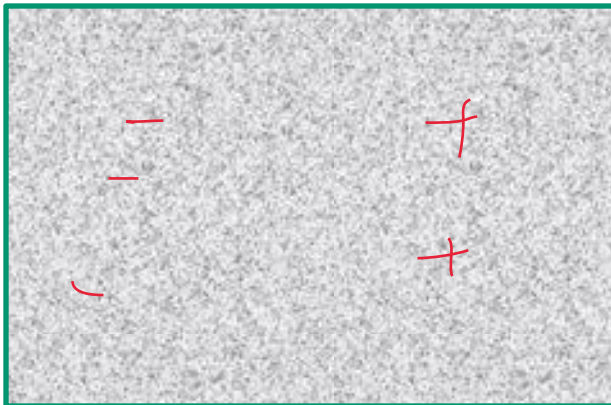


### Margin satisfied

$1/\gamma^2$ clusters, all partitions separable by large margin
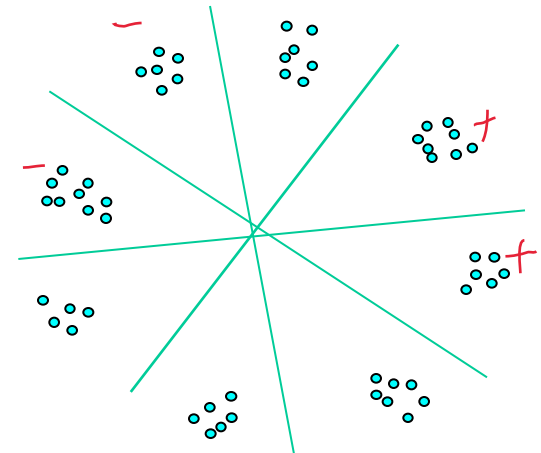
# Co-training

[Blum & Mitchell '98]

Different type of underlying regularity assumption:
    Consistency or Agreement Between Parts
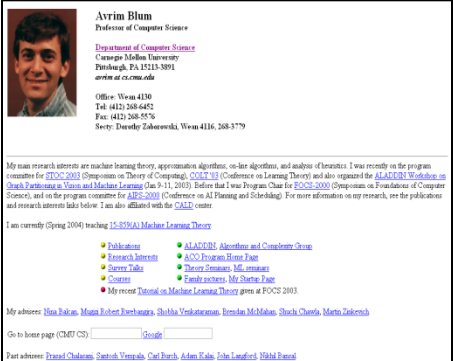
# Co-training: Self-consistency

Agreement between two parts : co-training [Blum-Mitchell98].

- examples contain two sufficient sets of features, $x = \langle x_1, x_2 \rangle$

- belief: the parts are consistent, i.e. $\exists\ c_1, c_2$ s.t. $c_1(x_1)=c_2(x_2)=c^*(x)$
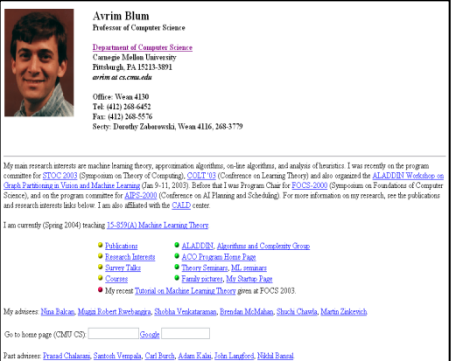
For example, if we want to classify web pages: $x = \langle x_1, x_2 \rangle$

as faculty member homepage or not



x - Link info & Text info

$x_1$- Text info

$x_2$- Link info

# Iterative Co-Training

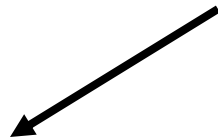**Idea**: Use small labeled sample to learn initial rules.

*link*
- E.g., "my advisor" pointing to a page is a good indicator it is a faculty home page.

*Text*
- E.g., "I am teaching" on a page is a good indicator it is a faculty home page.

**Idea**: Use unlabeled data to  propagate learned information.



my advisor

# Iterative 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.

**Idea**: Use unlabeled data to  propagate learned information.

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

$$C_1(x_1) = C_2(x_2)$$

$$X = (x_1, x_2)$$

$$\langle x_1, x_2 \rangle \quad \langle x_1, x_2 \rangle$$
$$\langle x_1, x_2 \rangle \quad \langle x_1, x_2 \rangle$$
$$\langle x_1, x_2 \rangle \quad \langle x_1, x_2 \rangle$$
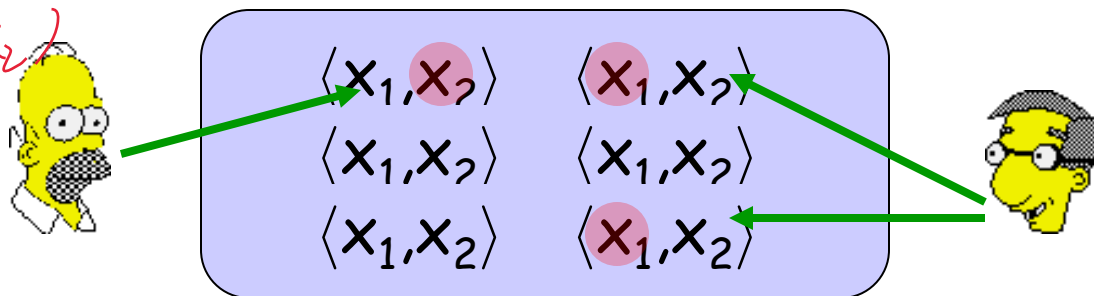
Training 2 classifiers, one on each type of info.  Using each to help train the other.

# Iterative Co-Training

Works by using unlabeled data to propagate learned information.

$$X = (X_1, X_2)$$

$$\left(C_1(X_1) = C_2(X_2)\right)$$

$$= \check{C}(X)$$

$X_1$   $X_2$
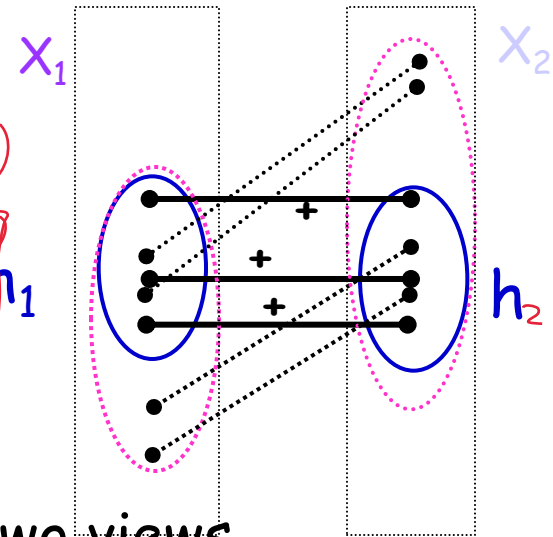
$h_1$   $h_2$

- Have learning algos $A_1$, $A_2$ on each of the two views.
- Use labeled data to learn two initial hyp. $h_1$, $h_2$.

Repeat

- Look through unlabeled data to find examples where one of $h_i$ is confident but other is not.
- Have the confident $h_i$ label it for algorithm $A_{3-i}$.

# Co-Training Algorithm

$h(x) = sign(\bar{w}^T x)$

$|\bar{w}^T x|$

confidence

Wrapper

**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.

labeled: $\ell \lessgtr \overbrace{\boxed{x_1}}^{d_1} \ell \ell' \overbrace{\boxed{x_2}}^{d_2} \ell \lessgtr \boxed{y}$

unlabel: $u \lessgtr \boxed{x_1} u \lessgtr \boxed{x_2}$

# Original Application: Webpage classification

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)
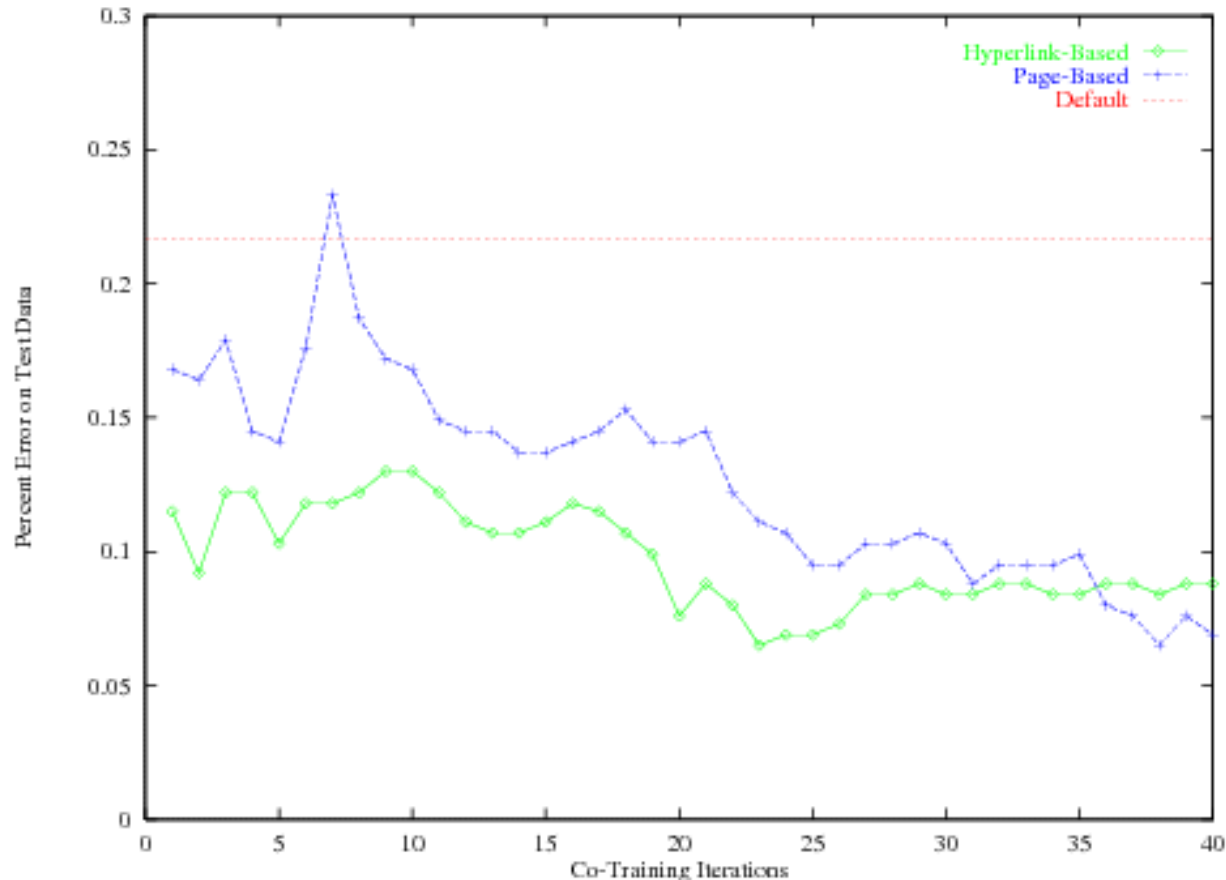
# Iterative Co-Training
## A Simple Example: Learning Intervals



Labeled examples
Unlabeled examples

$c_2$

$c_1$

$+$

$-$   $-$

$h_2^1$

$h_1^1$
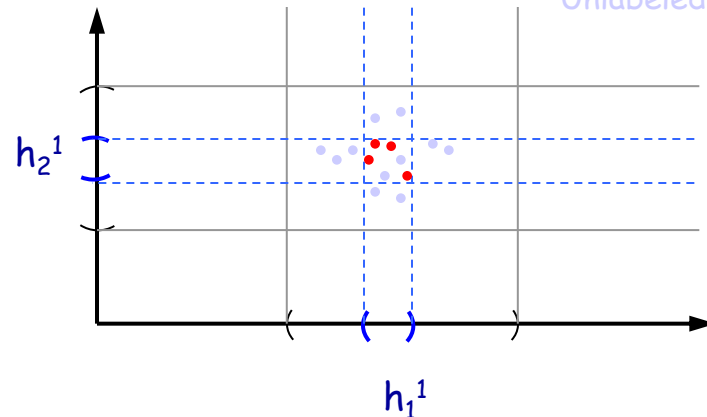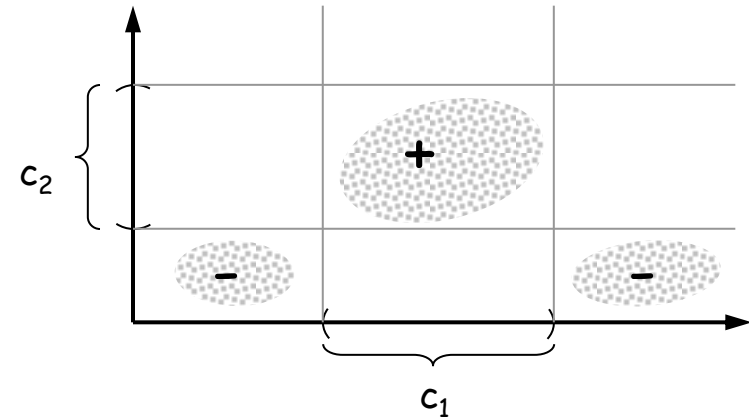
Use labeled data to learn $h_1^1$ and $h_2^1$

Use unlabeled data to bootstrap

$h_2^1$

$h_1^2$

$h_2^2$

$h_1^2$
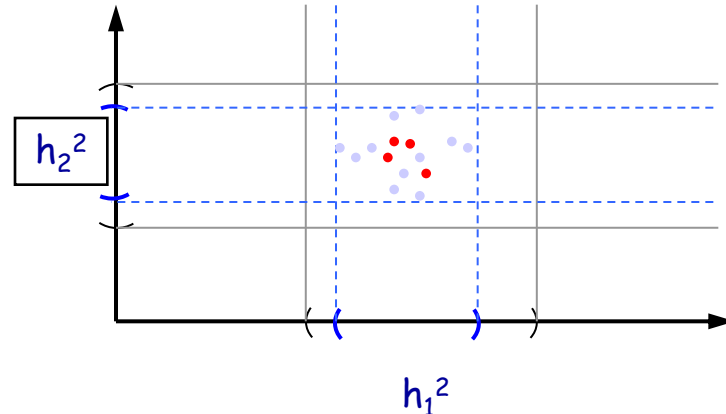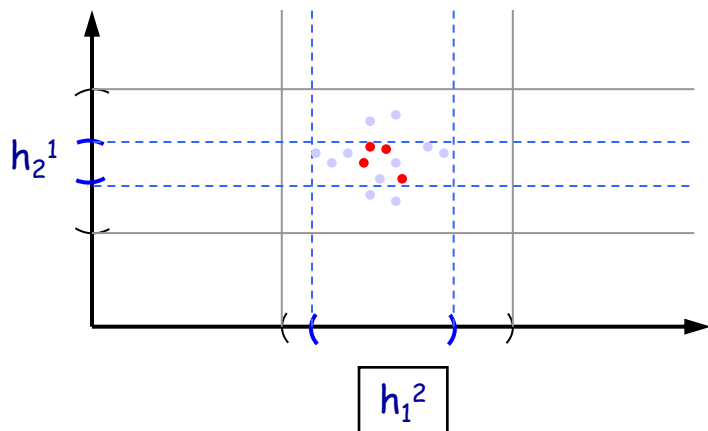
# Expansion, Examples: Learning Intervals

$C_1(x) \neq C_2(x)$
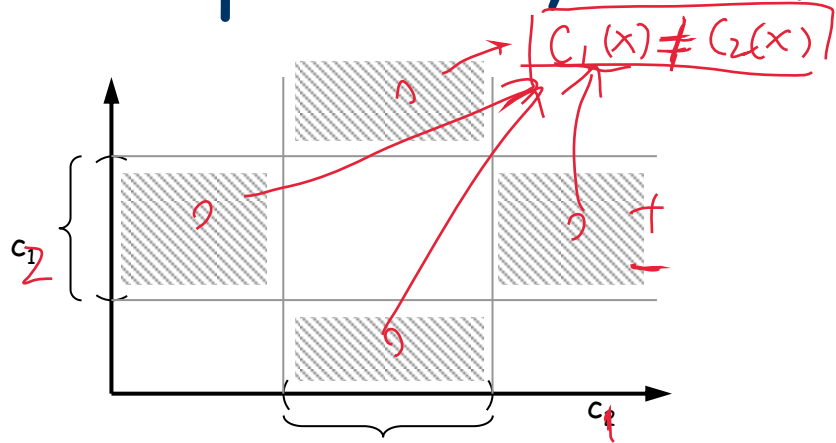
Consistency: zero probability mass in the regions

## Non-expanding (non-helpful) distribution

$D^+$

$S_2$

$S_1$

$c_1$

$c_2$
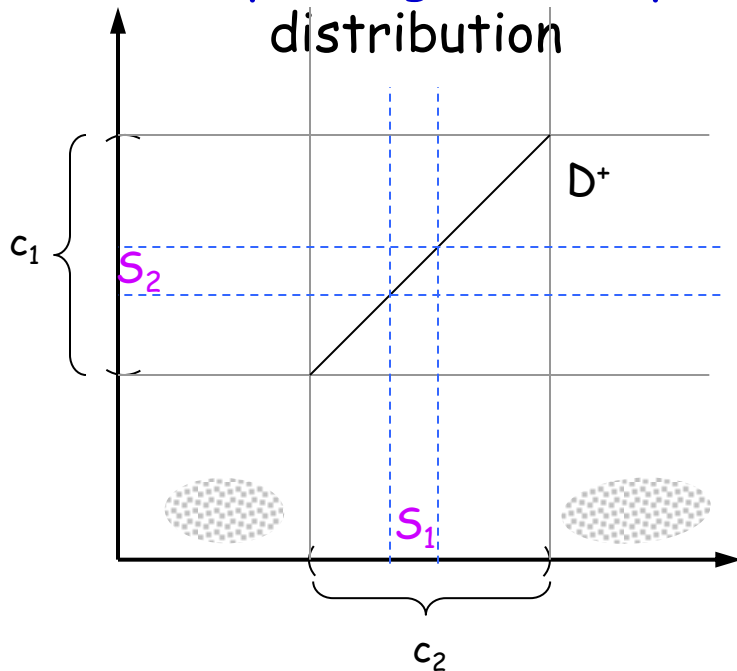
## Expanding distribution

$D^+$

$c_1$

$c_2$
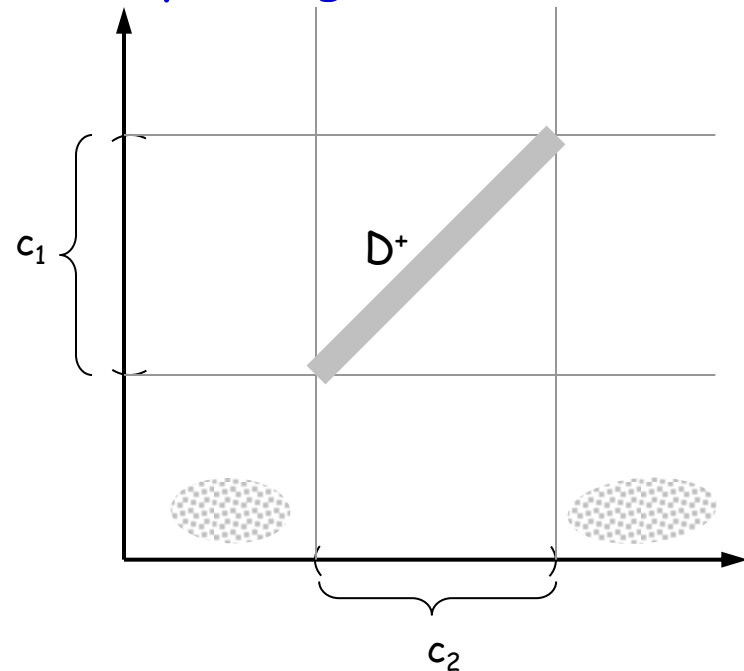
# Co-training [BM'98]

Say that $h_1$ is a weakly-useful predictor if

$$\Pr[h_1(x) = 1 | c_1(x) = 1] > \Pr[h_1(x) = 1 | c_1(x) = 0] + \gamma.$$

Has higher probability of saying positive on a true positive than it does on a true negative, by at least some gap $\gamma$

Say we have enough labeled data to produce such a starting point.

**Theorem:** if $C$ is learnable from random classification noise, we can use a weakly-useful $h_1$ plus unlabeled data to create a strong learner under independence given the label.

# Co-training/Multi-view SSL: Direct Optimization of Agreement

**Input**: $S_l = \{(x_1, y_1), ..., (x_{m_l}, y_{m_l})\}$
$S_u = \{x_1, ..., x_{m_u}\}$

\# views : $\cup$

$$\text{argmin}_{h_1, h_2} \sum_{v=1}^{v} \sum_{i=1}^{m_l} l(h_v(x_i), y_i) + C \sum_{i=1}^{m_u} \text{agreement}(h_1(x_i), h_2(x_i)) \quad + \sum_{v=1}^{v} \Omega(h_v)$$

$v, v'$
$v \neq v'$

Each of them has small labeled error

Regularizer to encourage agreement over unlabeled dat

E.g.,

P. Bartlett, D. Rosenberg, AISTATS 2007;   K. Sridharan, S. Kakade, COLT 2008

# Co-training/Multi-view SSL: Direct Optimization of Agreement

<u>Input</u>: $S_l = \{(x_1, y_1), \dots, (x_{m_l}, y_{m_l})\}$
$S_u = \{x_1, \dots, x_{m_u}\}$

$$\text{argmin}_{h_1, h_2} \sum_{l=1}^{2} \sum_{i=1}^{m_l} l(h_l(x_i), y_i) + C \sum_{i=1}^{m_u} \text{agreement}(h_1(x_i), h_2(x_i))$$

- $l(h(x_i), y_i)$ loss function

  - E.g., square loss $l(h(x_i), y_i) = (y_i - h(x_l))^2$
  - E.g., 0/1 loss $l(h(x_i), y_i) = 1_{y_i \neq h(x_i)}$

E.g.,

P. Bartlett, D. Rosenberg, AISTATS 2007;  K. Sridharan, S. Kakade, COLT 2008

# Many Other Applications

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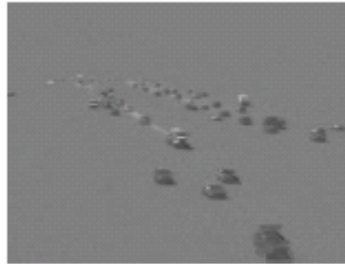


Original images                  Foreground images

Goal: car detection
   #labeled images: 50
   #unlabeled images: 22,000

*Graph*

# Similarity Based Regularity

[Blum&Chwala01], [ZhuGhahramaniLafferty03]

# Graph-based Methods

- Assume we are given a pairwise similarity fnc and that very similar examples probably have the same label.

- If we 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".

E.g., handwritten digits [Zhu07]:

not similar

'indirectly' similar
with stepping stones

# 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.
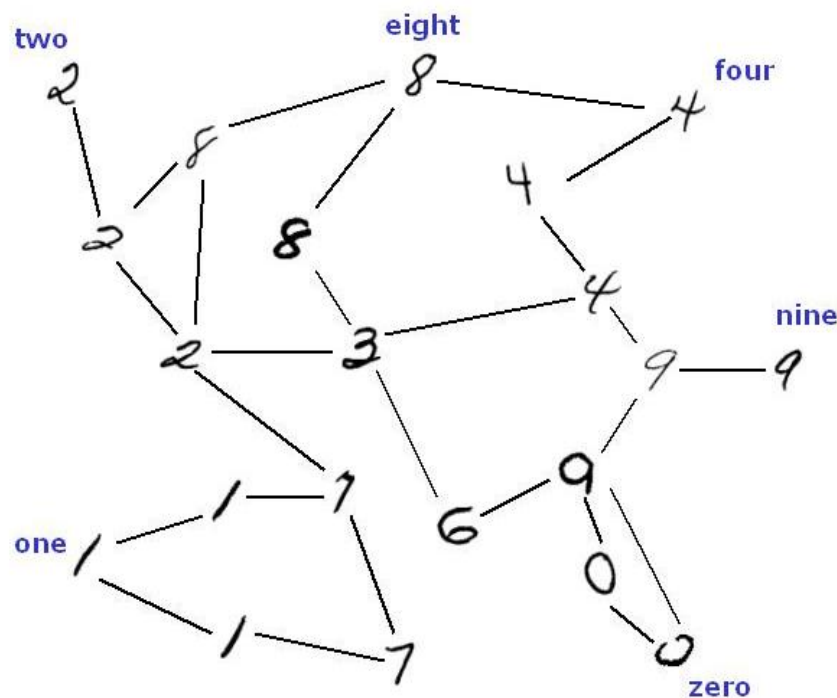
# Graph-based Methods

Often, transductive approach. (Given L + U, output predictions on U). Are alllowed to output any labeling of $L \cup U$.

**Main Idea**:

- Construct graph G with edges between very similar examples.

- Might have also glued together in G examples of different classes.

- Run a <u>graph partitioning</u> algorithm to separate the graph into pieces.
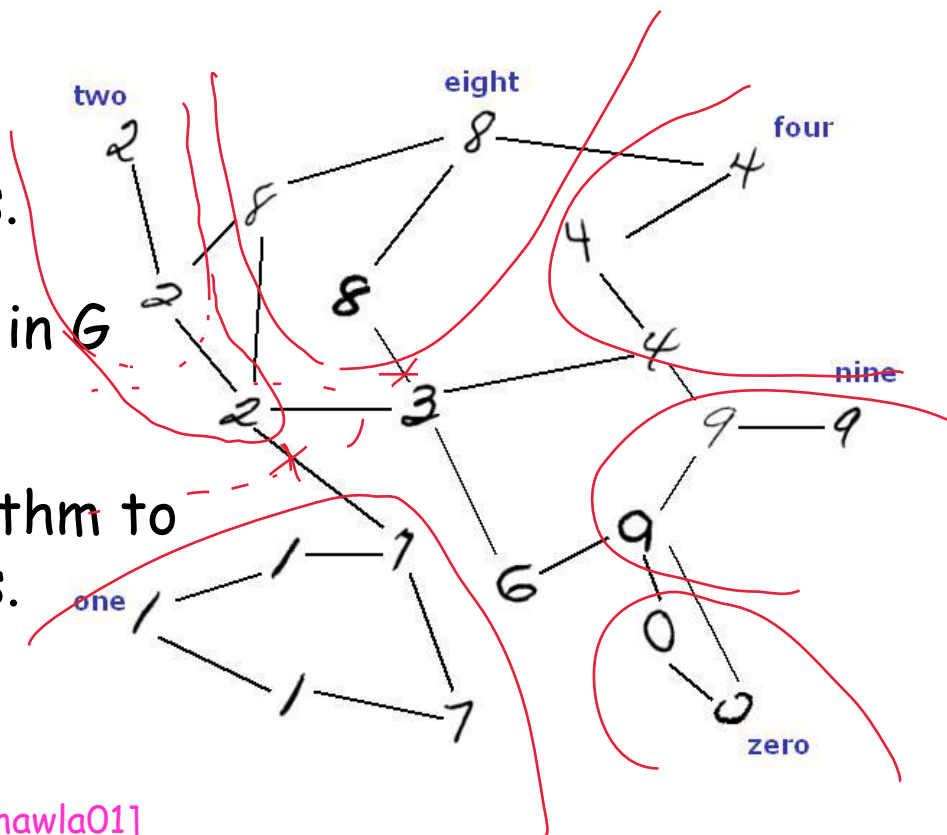
Several methods:
  – Minimum/Multiway cut [Blum&Chawla01]
  – Minimum "soft-cut" [ZhuGhahramaniLafferty'03]
  – Spectral partitioning
  – ...

# How to Create the Graph

- Empirically, the following works well:

  1. Compute distance between $i, j$

  2. For each $i$, connect to its kNN.  k very small but still connects the graph

  3. Optionally put weights on (only) those edges

  $$\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

  4. Tune $\sigma$

$S_l = \{(x_i, y_i)\}_{i=1}^{l}$

$S_u = \{:x_i\}_{i=1}^{n}$

# How to Create the Graph

$G = \langle V, E \rangle$

$V$ (Vertex / Node) :     example $(S_l \cup S_u)$

$E$ (Edge) :     similarity / weight.

Step ①   Adjancey Graph   ← (save local information)     $W \in R^{n \times n}$

    → Sym

    (1)   K−NN     (0−1)

    (2)   $\Sigma$−NN     (0−1)   $\Sigma = dist(X_i, X_j)$   $\forall i,j$

    i-th j-th

Step ②   Graph weighting

    (1) simple

    (2) $e^{\left(-\frac{\|X_i - X_j\|_2^2}{2\sigma^2}\right)}$

$X_i, X_j$

# Minimum "soft cut"

**Objective** Solve for probability vector over labels $f_i$ on each unlabeled point $i$.

(labeled points get coordinate vectors in direction of their known label)

- Minimize $\sum_{e=(i,j)} w_e \|f_i - f_j\|^2$

  where $\|f_i - f_j\|$ is Euclidean distance.

- Can be done efficiently by solving a set of linear equations.



(0100000000) two

(0000000100) eight

(0001000000) four

nine

(0000000010)

(1000000000) one

zero
(0000000001)
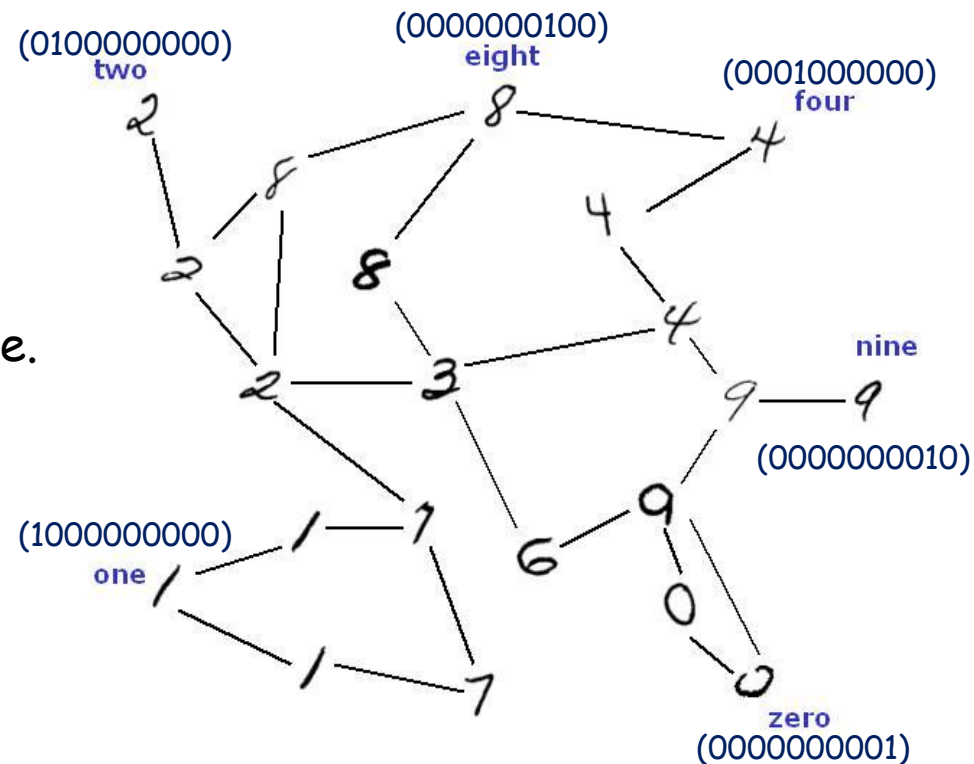
# Minimum "soft cut"

① Initial label assignment

$$f = \{-1, 0, 1\}^n, \quad n = m_t + m_u$$

i-th $\rightarrow$ $f_i = \begin{cases} \pm 1, & x_i \in S_t \\ 0, & x_i \in S_u \end{cases}$

② Prediction.

$$y_i = \text{sign}(f_i).$$

$$\min_{f} \quad \sum_{i,j} W_{ij} \underbrace{(f_i - f_j)^2}_{X}$$

$$\text{s.t.} \quad f_i = y_i, \quad x_i \in S_t$$

$$\forall x_i \in S_u, \begin{cases} f_i \in \{-1, 1\} \longrightarrow \text{hard} \\ f_i \in \mathbb{R} \longrightarrow \text{soft} \end{cases}$$

$$\sum_{i,j} W_{ij} (f_i - f_j)^2$$

$$= \sum_{i,j} W_{ij} (f_i^2 - 2f_i f_j + f_j^2)$$

$$= 2 \sum_{i,j} W_{ij} f_i^2 - 2 \sum_{i,j} W_{i,j} f_i f_j$$

$$= 2 f^T D f - 2 f^T W f$$

$$= 2 f^T (D - W) f$$

$$(D)_{ii} = \sum_{j=1}^{n} W_{i,j}$$

$$L = D - W$$

Laplacian matrix

$\boxed{\text{Inductive}}$

$\text{Local}$

$$\min_{w} \left| \sum_{x_i \in S_e} \left(1 - y_i W^T x_i\right)_+ + \lambda_1 \|W\|_2^2 \right| + \lambda_2 \underbrace{W^T X^T L X w}$$

$X = \left\{ \begin{matrix} l \} \\ \\ n \end{matrix} \right.$  $\boxed{X_l}$  $\boxed{X_u}$

$\text{Global}$

manifold regularization

Convex

$\underline{W^T A w}$

$A = X^T L X$

semi-supervised

SVM

$$\min_{f} \frac{f^T L f}{\underline{\qquad}} \quad \in R^n$$

$$\downarrow$$

$$(Xw)^T L (Xw)$$

$f = Xw$
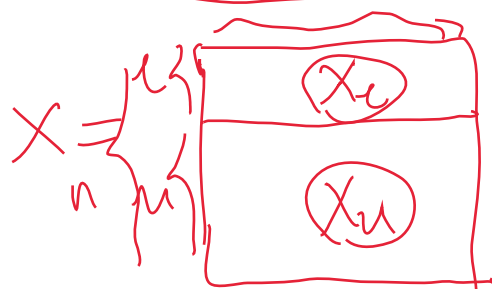
Spectral Clustering

$$\min_{f} f^T L f$$

$$s.t \quad f^T D f = 1$$

# What You Should Know

- Unlabeled data useful if we have beliefs not only about the form of the target, but also about its relationship with the underlying distribution.

- Different types of algorithms (based on different beliefs).

  *Sem-supervised / S3VM*

  - Transductive SVM [Joachims '99]

  - Co-training [Blum & Mitchell '98]

  - Graph-based methods [B&C01], [ZGL03]

# Supplementary Materials

1. Self-Training
2. Generative Models

# Self-Training

Maybe a simple way of using unlabeled data

- Initialize $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and $U = \{\mathbf{x}_j\}_{i=l+1}^{n}$
- Repeat
  1. Train $f$ from $L$ using supervised learning
  2. Apply $f$ to the unlabeled instances in $U$
  3. Remove a subset $S$ from $U$ ; add $\{(\mathbf{x}, f(\mathbf{x})) | \mathbf{x} \in S\}$ to $L$
- Until $U = \phi$

# Self-Training

- A wrapper method
- The choice of learner for $f$ is open
- Good for many real world tasks, e.g., natural language processing
- But mistake in choosing the $f$ can reinforce itself
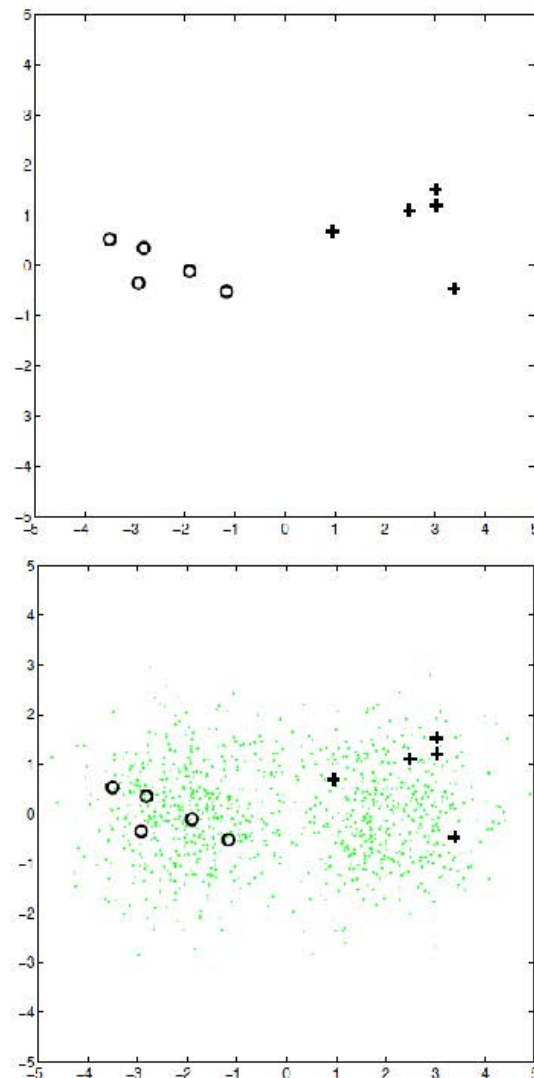
# Generative Model

## Gaussian mixture model (GMM)

- Model parameters:
  $\theta = \{\pi_i, \mu_i, \Sigma_i\}_{i=1}^{K}$, $\pi_i$: class priors, $\mu_i$:Gaussian means, $\Sigma_i$:covariance matrices

- Joint distribution

$$
\begin{aligned}
p(\mathbf{x}, \mathbf{y}|\theta) &= p(\mathbf{y}|\theta)p(\mathbf{x}|\mathbf{y}, \theta) \\
&= \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}; \mu_i, \Sigma_i)
\end{aligned}
$$

- Classification:

$$
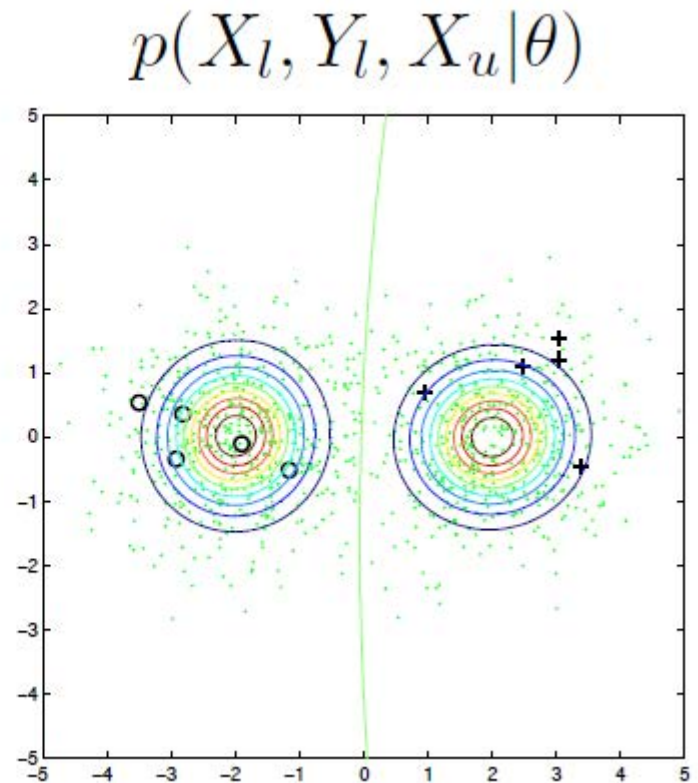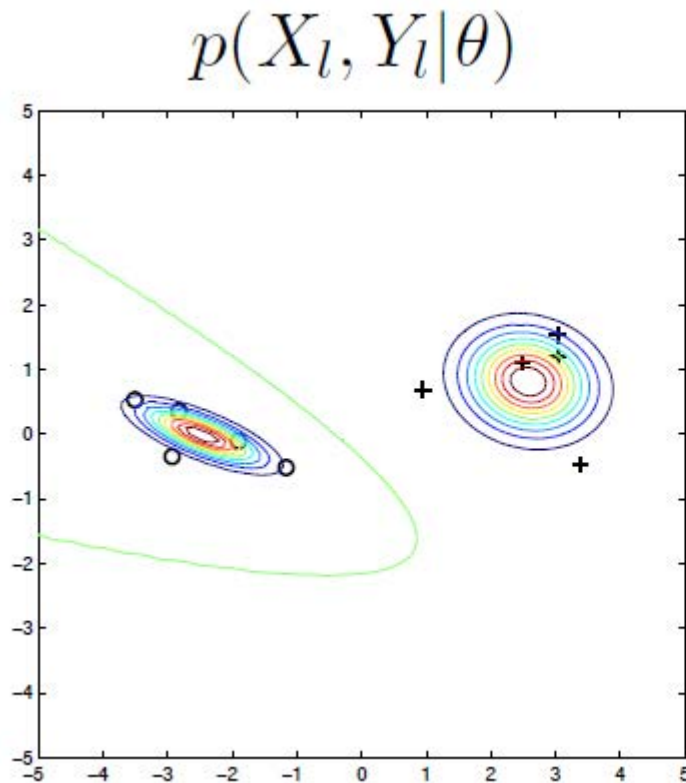p(\mathbf{y}|\mathbf{x}, \theta) = \frac{p(x, y|\theta)}{\sum_{i=1}^{K} p(\mathbf{x}, y_i|\theta)}
$$

# Generative Model

Effect of unlabeled data in GMM

The difference comes from maximizing different quantities



$p(X_l, Y_l|\theta)$

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

# Generative Model

- 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 $\theta$, the maximum a posteriori (MAP) estimate, or be Bayesian
- common mixture models used in semi-supervised learning:
  - ▶ Mixture of Gaussian distributions (GMM) – image classification
  - ▶ Mixture of multinomial distributions (Naïve Bayes) – text categorization
  - ▶ Hidden Markov Models (HMM) – speech recognition
- Learning via the Expectation-Maximization (EM) algorithm

# Generative Model

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 $\theta$ 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

# Generative Model

## The EM algorithm for GMM

1. Start from MLE $\theta = \{w, \mu, \Sigma\}_{1:2}$ on $(X_l, Y_l)$,
   - $w_c$=proportion of class $c$
   - $\mu_c$=sample mean of class $c$
   - $\Sigma_c$=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
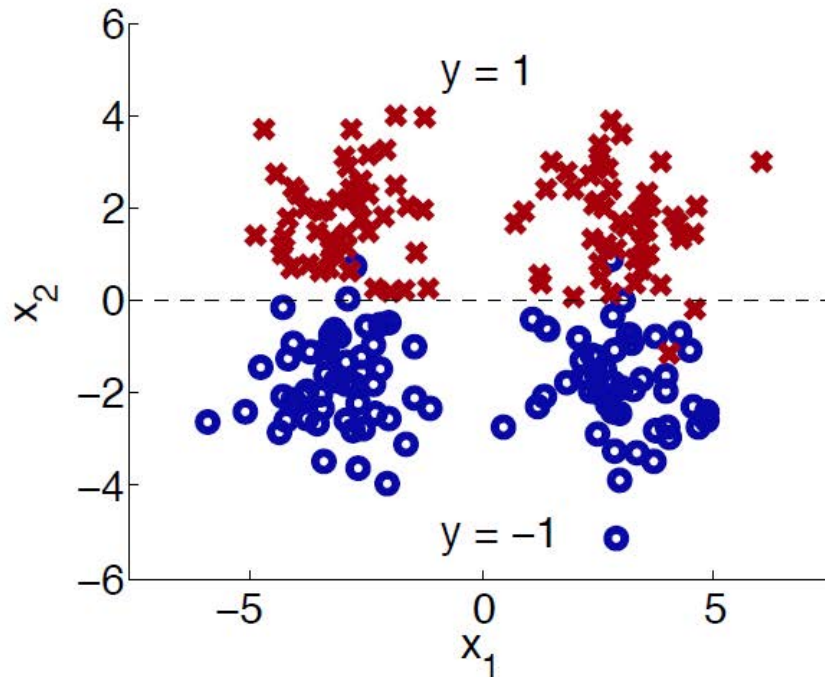   - label $p(y = 2|x, \theta)$-fraction of $x$ with class 2

3. The M-step: update MLE $\theta$ with (now labeled) $X_u$

Can be viewed as a special form of self-training.
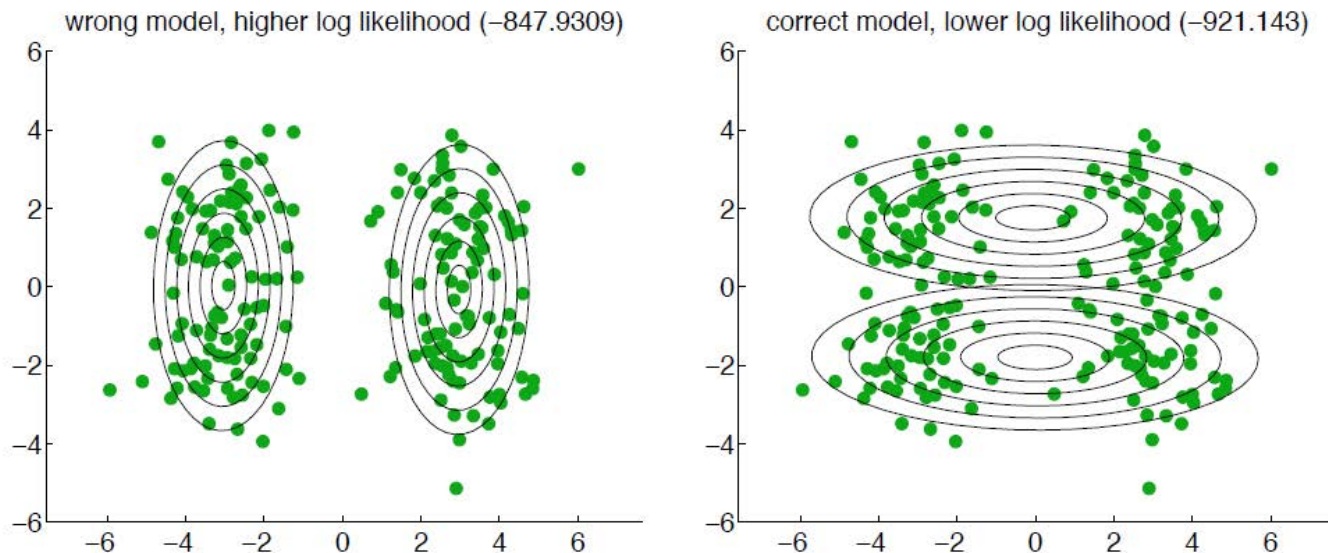
# Generative Model

## The assumption of GMM

- **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:

# Generative Model

## The assumption of GMM



wrong model, higher log likelihood (−847.9309)  |  correct model, lower log likelihood (−921.143)

Heuristics to lessen the danger

- Carefully construct the generative model, e.g., multiple Gaussian distributions per class
- Down-weight the unlabeled data $(\lambda < 1)$

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