Semi-Supervised Learning

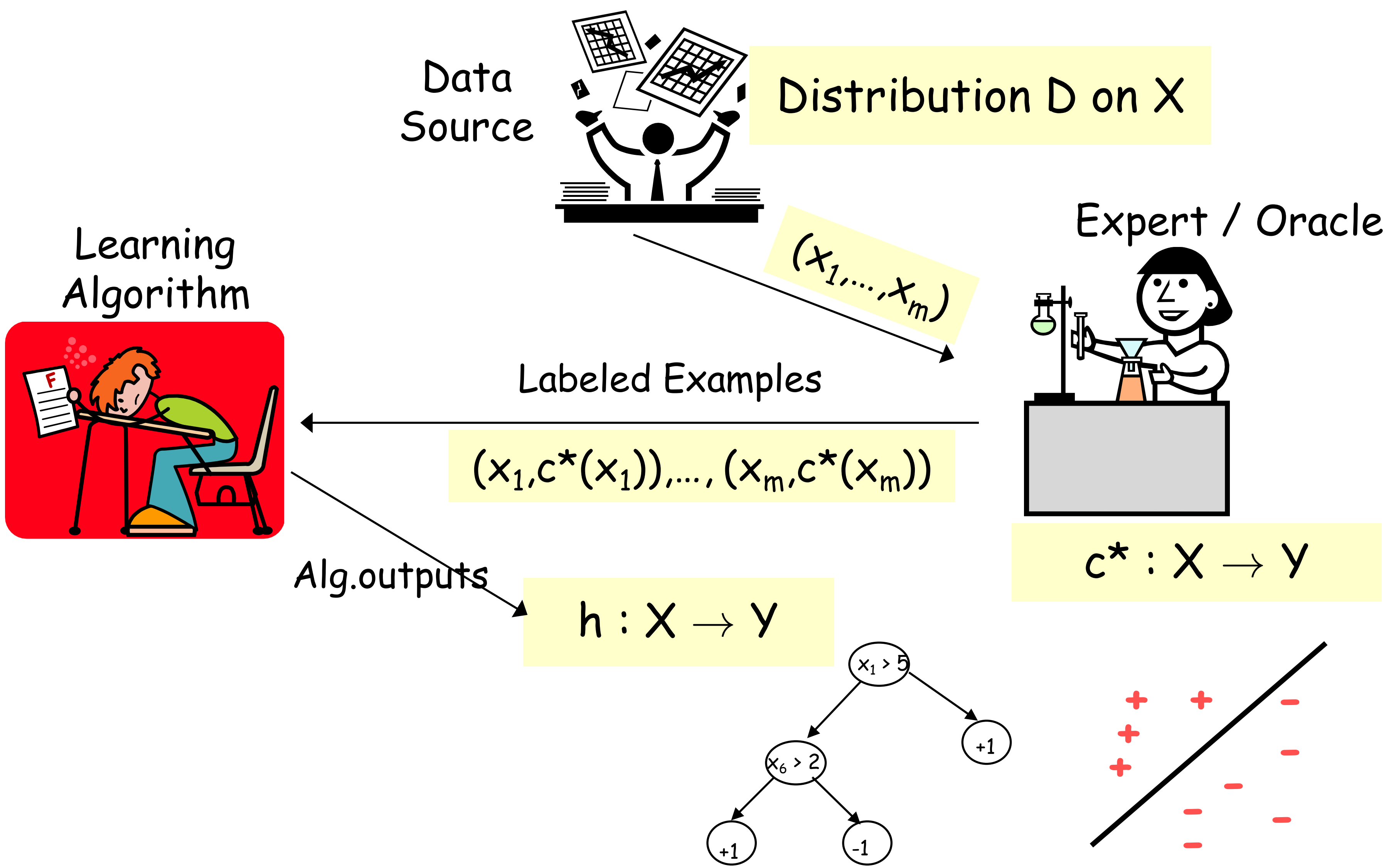
Maria-Florina Balcan

04/22/2019

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

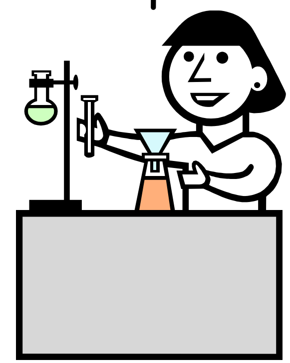


Fully Supervised Learning

Data Source

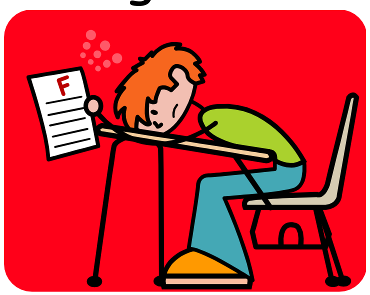
Distribution D on X

Expert / Oracle



$(x_1, \ldots, x_m)$

Learning Algorithm



Labeled Examples

$(x_1, c^*(x_1)), \ldots, (x_m, c^*(x_m))$

$c^* : X \rightarrow Y$

Alg.outputs

$h : X \rightarrow Y$

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

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

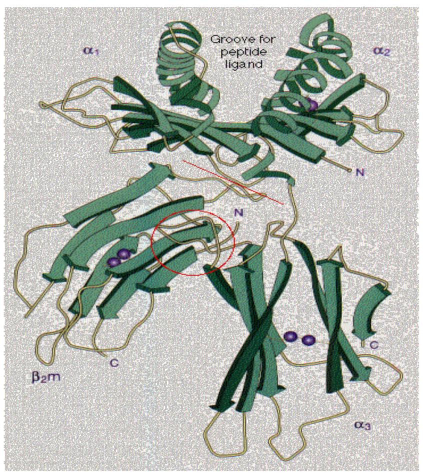
Goal: h has small error over D.

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

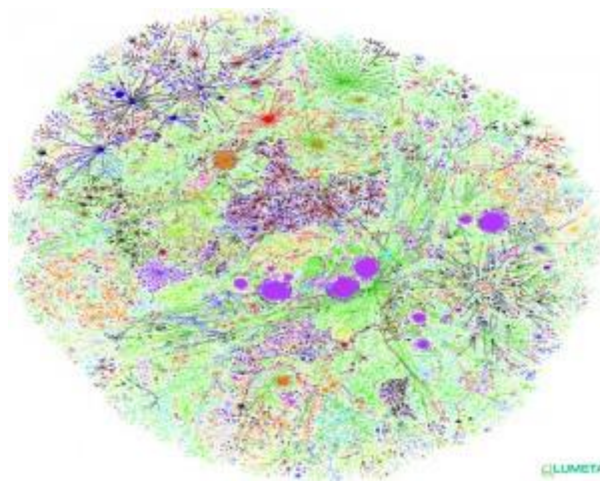
Classic Paradigm Insufficient Nowadays

Modern applications: massive amounts of raw data.

Only a tiny fraction can be annotated by human experts.



Protein sequences



Billions of webpages

Images

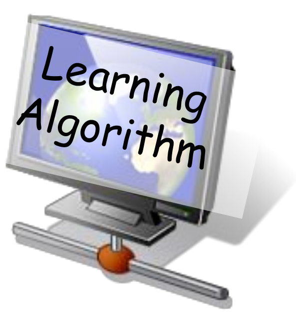
Modern ML: New Learning Approaches

Modern applications: massive amounts of raw data.

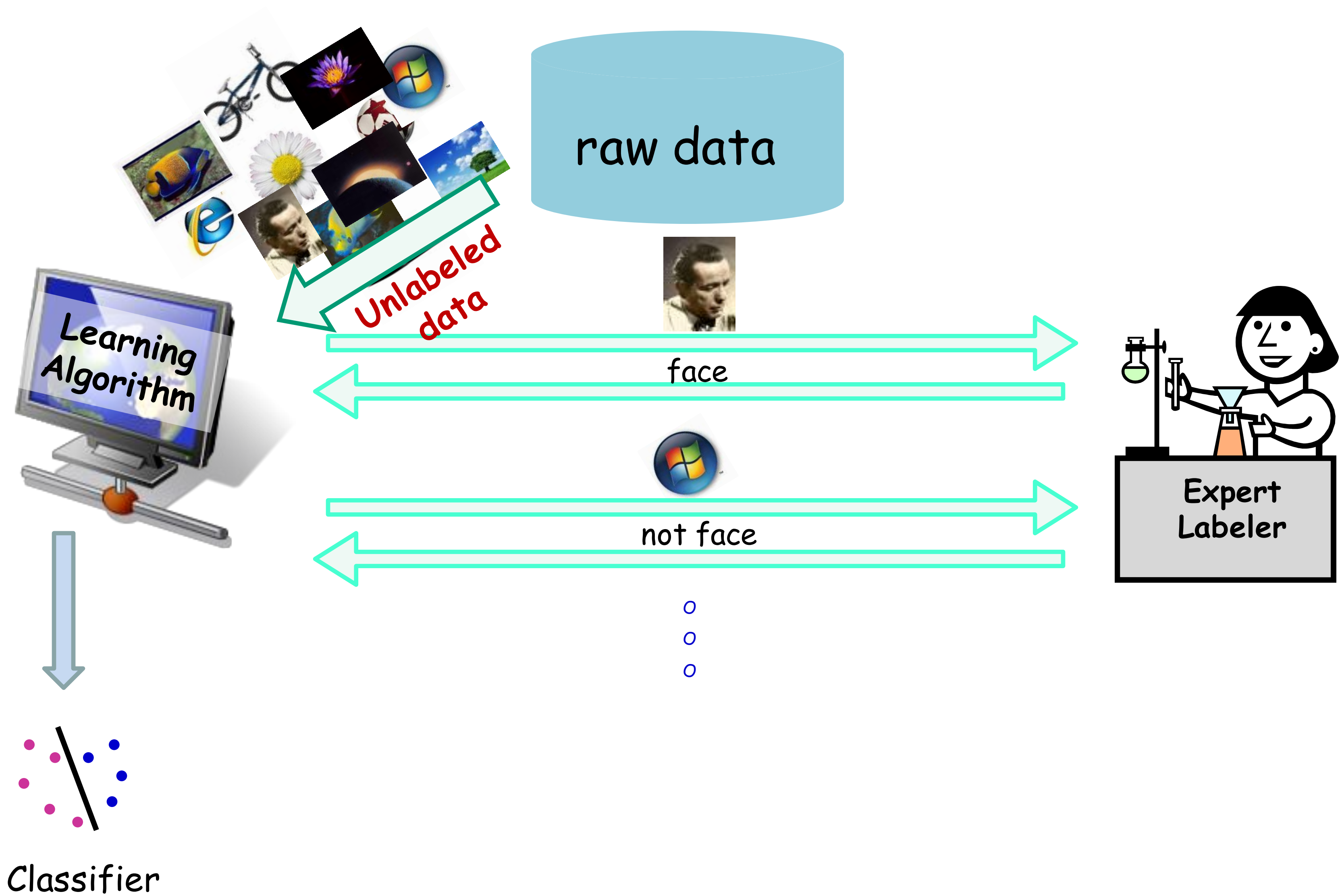
Techniques that best utilize data, minimizing need for expert/human intervention.

Paradigms where there has been great progress.

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

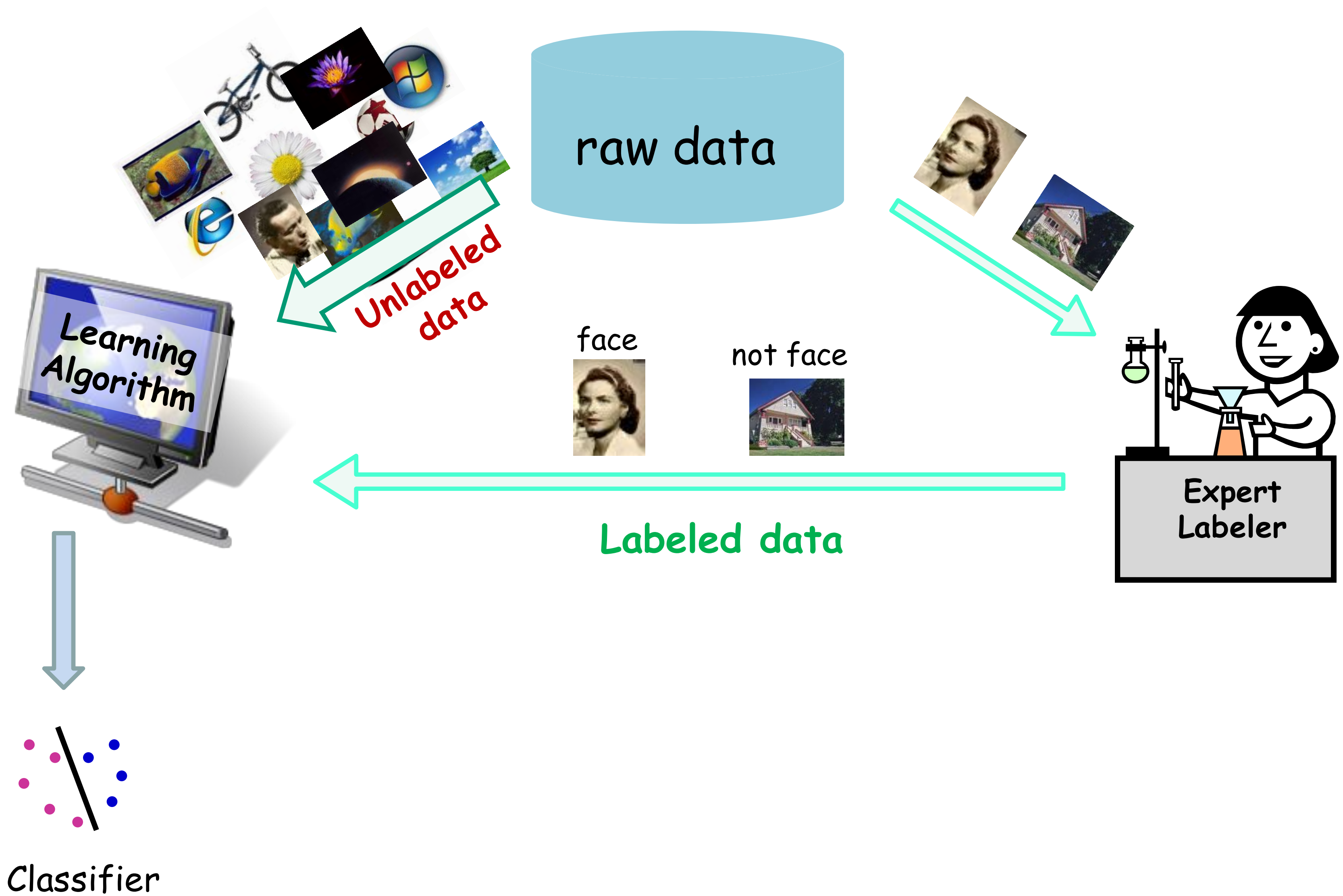


Active Learning

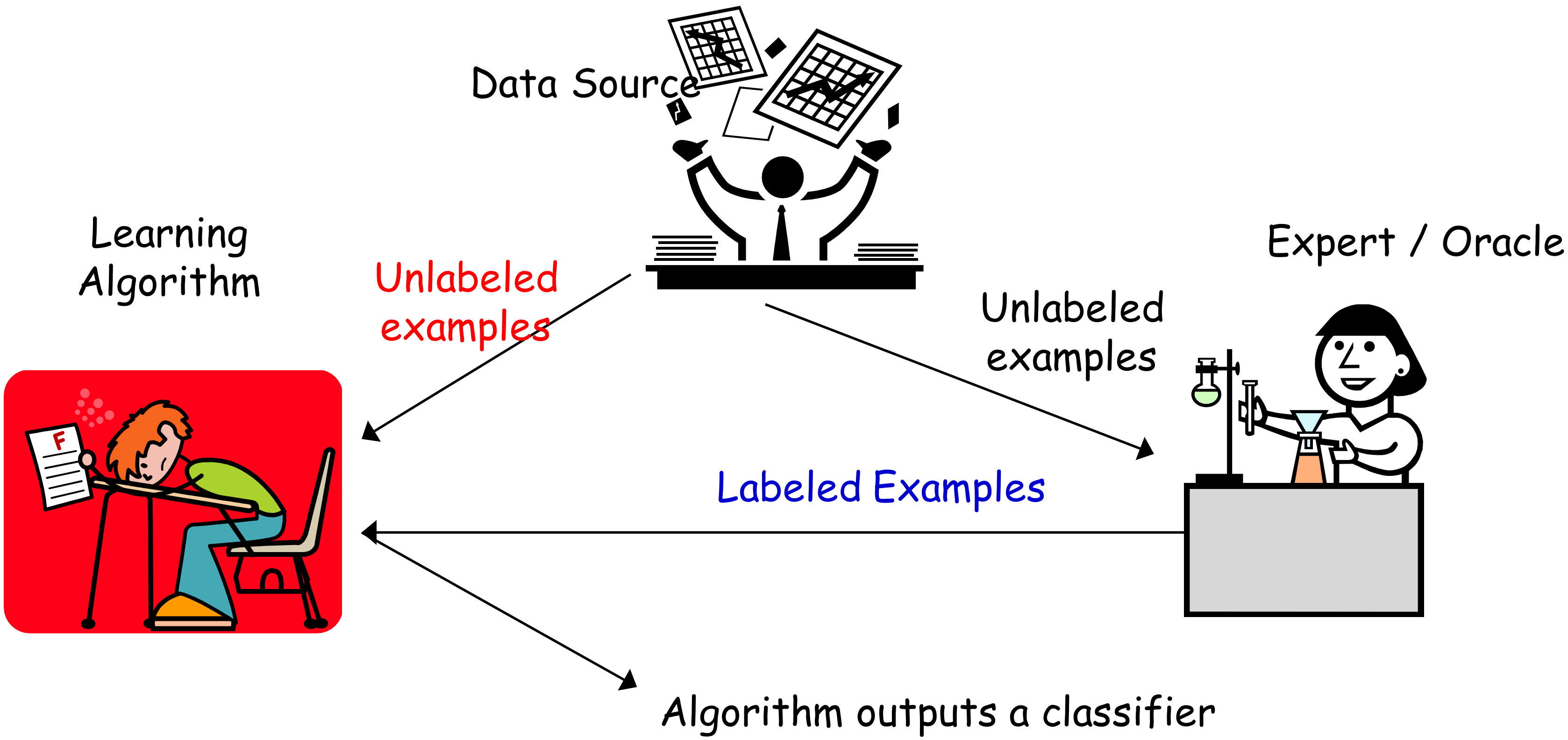


POLL

Semi-Supervised Learning



Semi-Supervised Learning



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

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

Goal: h has small error over D.

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

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]

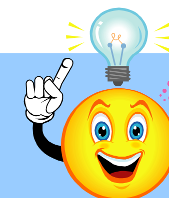
Test of time awards at ICML!

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.

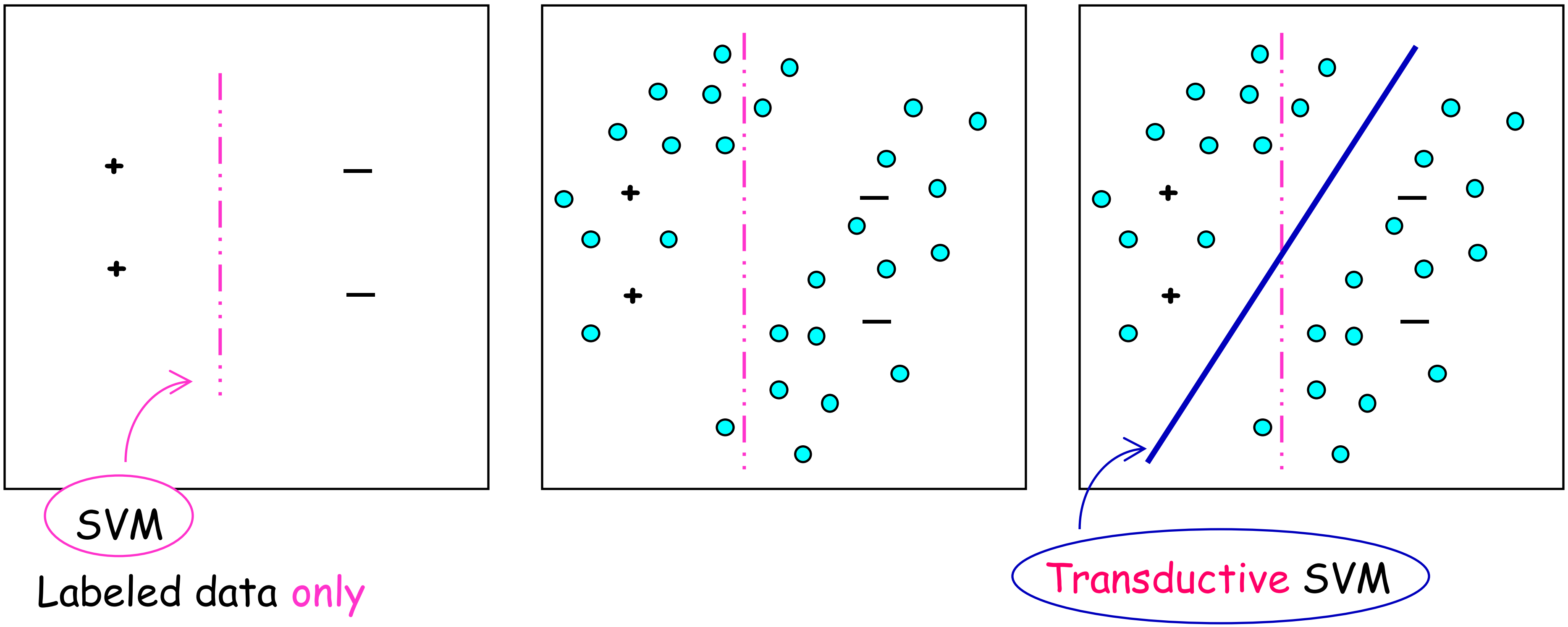
Semi-supervised SVM

[Joachims '99]

Margins based regularity

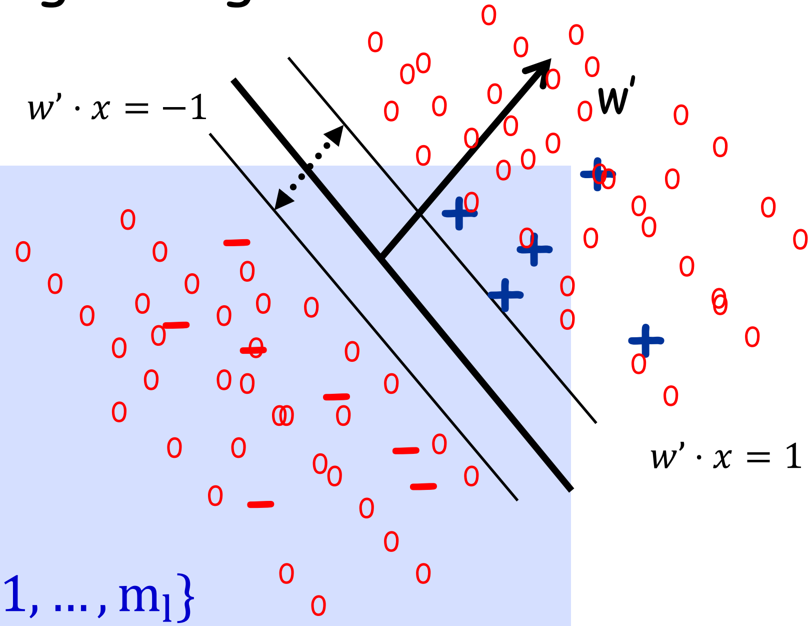
Target goes through low density regions (large margin).

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



Transductive Support Vector Machines

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



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

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

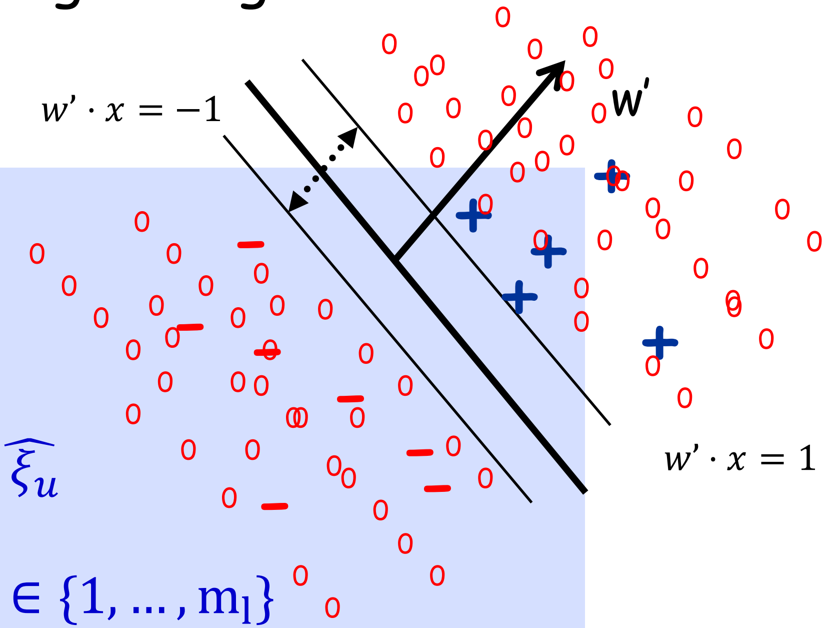
$\text{argmin}_w \, ||w||^2$ s.t.:

- $y_i \, w \cdot x_i \geq 1$, for all $i \in \{1, \ldots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1$, for all $u \in \{1, \ldots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \ldots, m_u\}$

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]



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

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

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

- $y_i \, w \cdot x_i \geq 1 - \xi_i$, 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 $u \in \{1, \ldots, m_u\}$

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

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

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

- $y_i \, w \cdot x_i \geq 1 - \xi_i$, for all $i \in \{1, \dots, m_l\}$
- $\widehat{y_u} w \cdot x_u \geq 1 - \widehat{\xi_u}$, for all $u \in \{1, \dots, m_u\}$
- $\widehat{y_u} \in \{-1, 1\}$ for all $u \in \{1, \dots, m_u\}$

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]

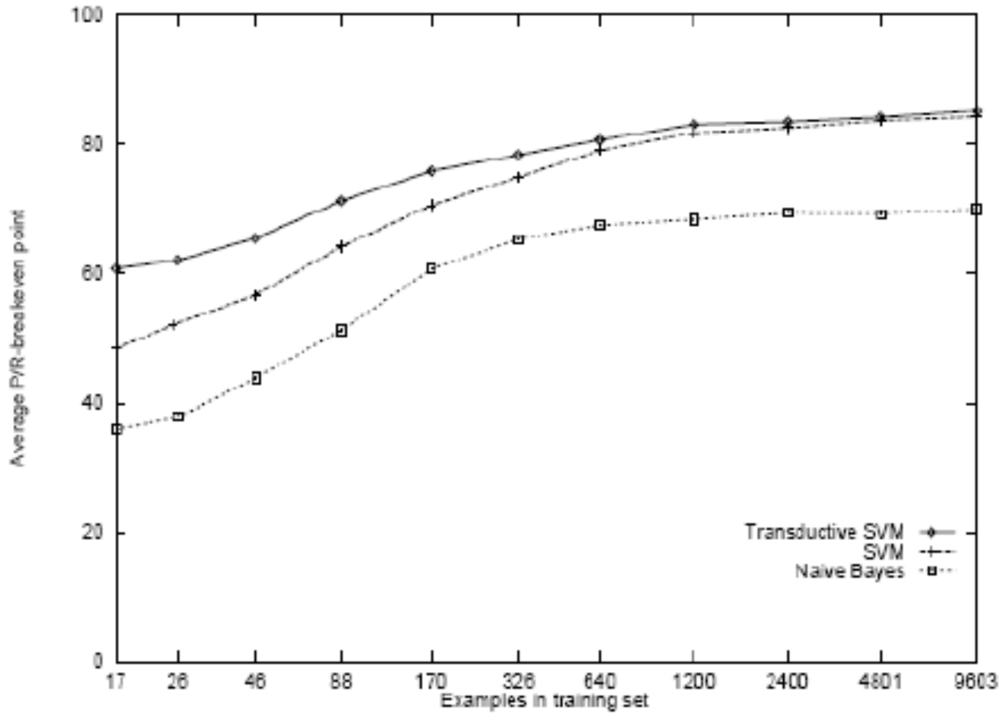
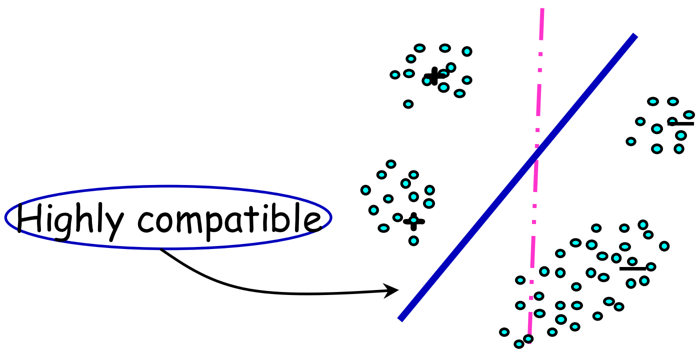


Figure 6: Average P/R-breakeven point on the Reuters dataset for different training set sizes and a test set size of 3,299.

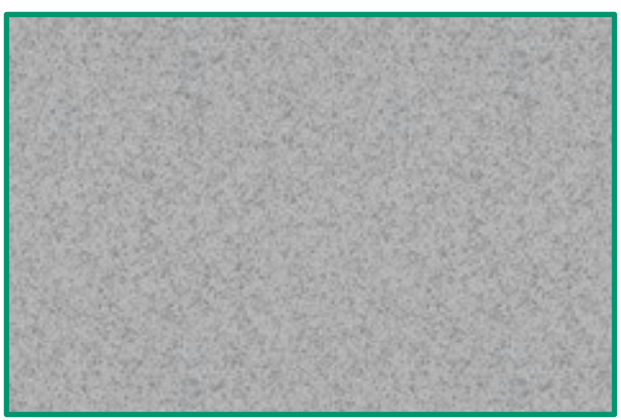
Transductive Support Vector Machines

Helpful distribution



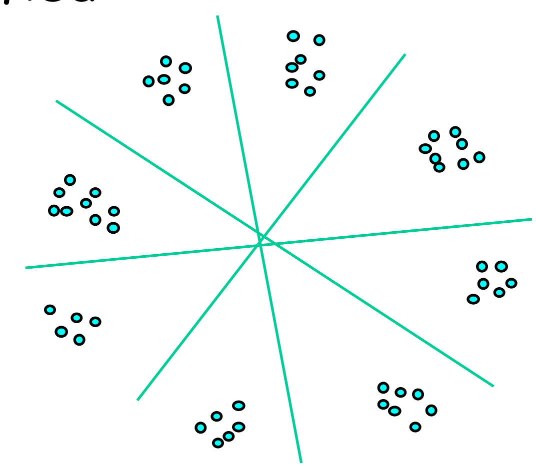
Non-helpful distributions

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

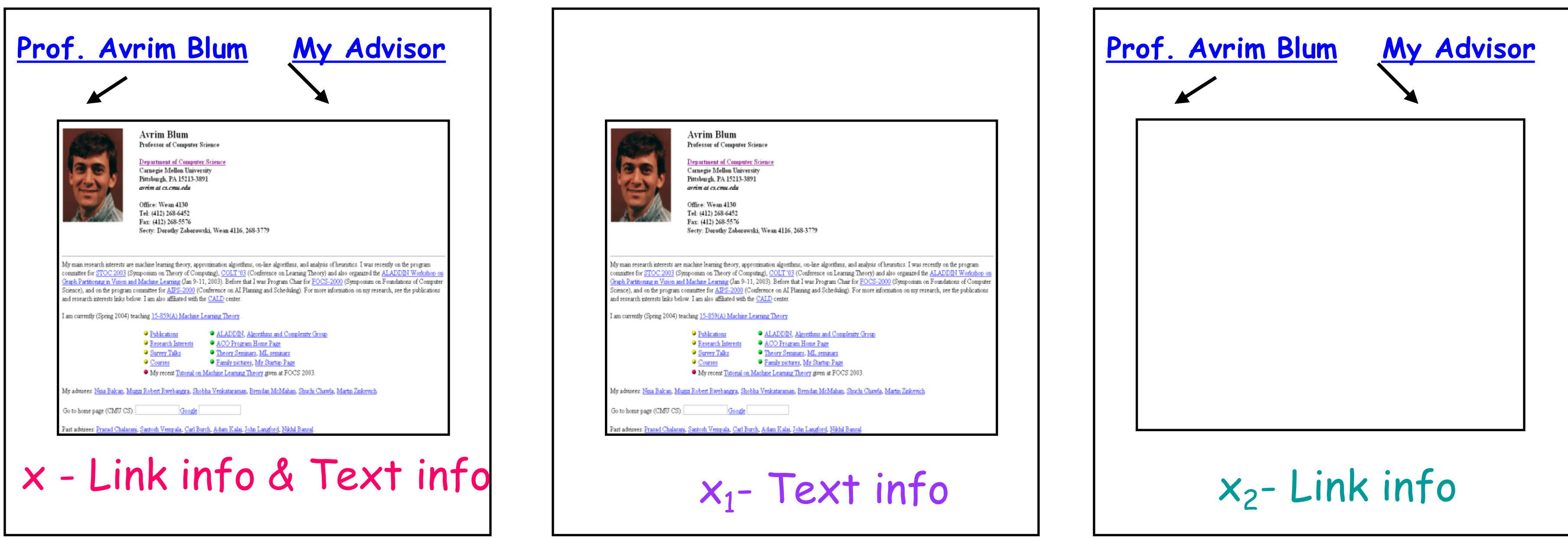
Co-training: Self-consistency

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

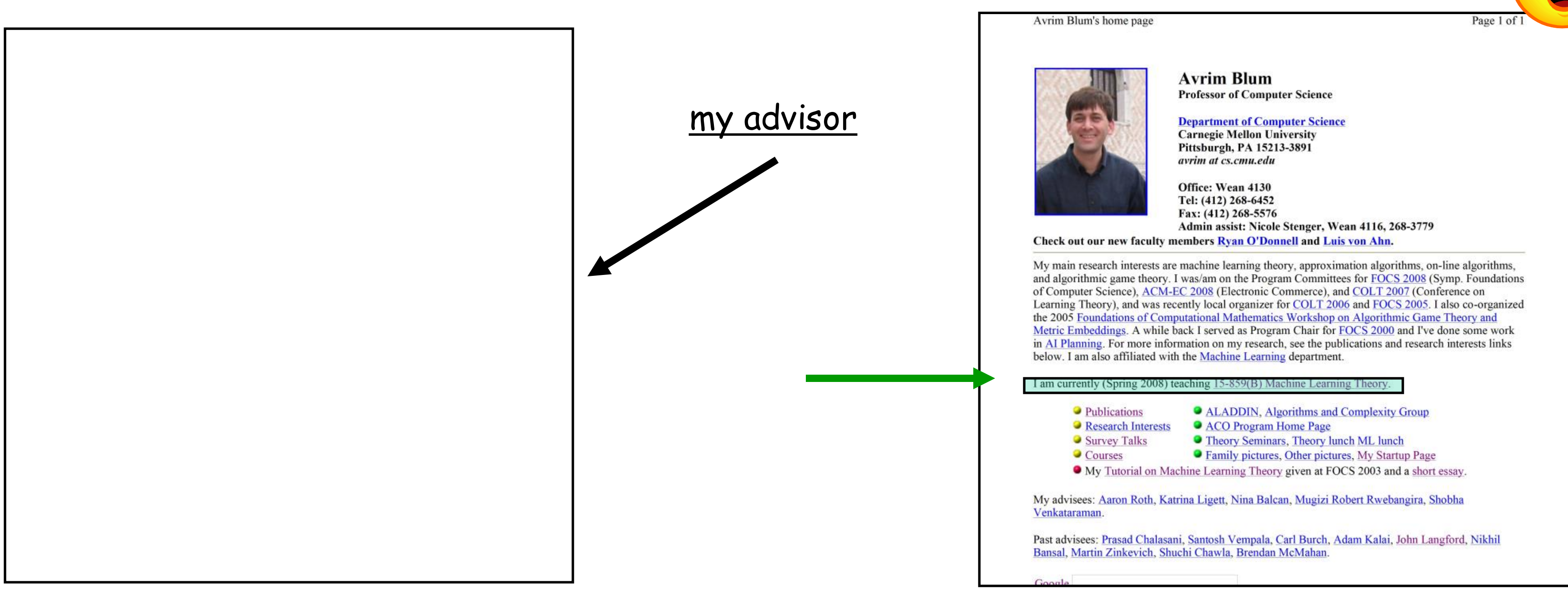


Iterative Co-Training

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.



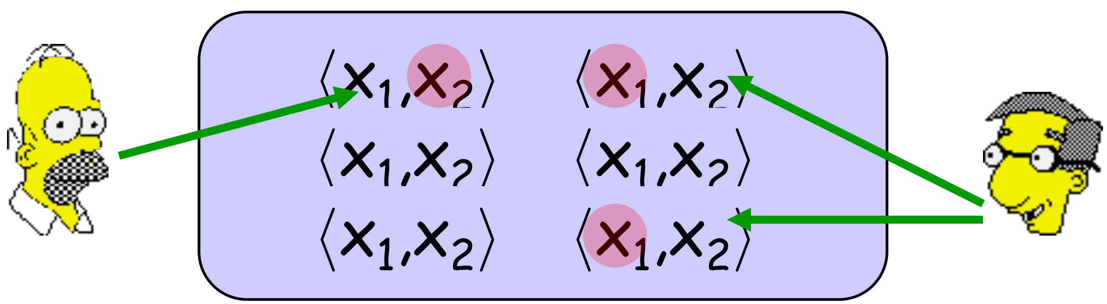
Iterative Co-Training

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.

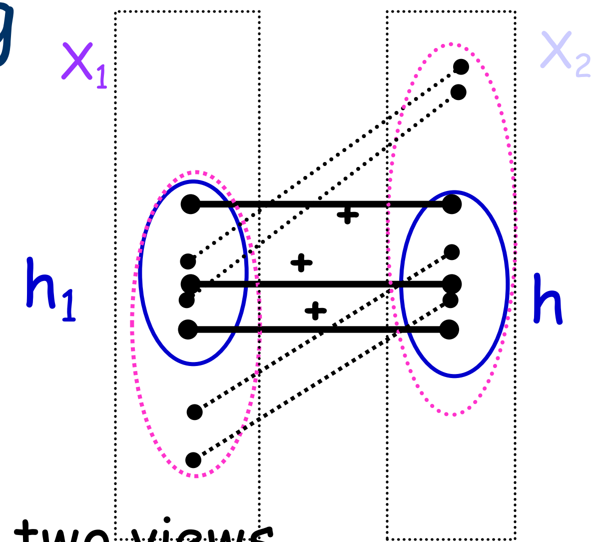
Idea: Use unlabeled data to propagate learned information.

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



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.

- 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

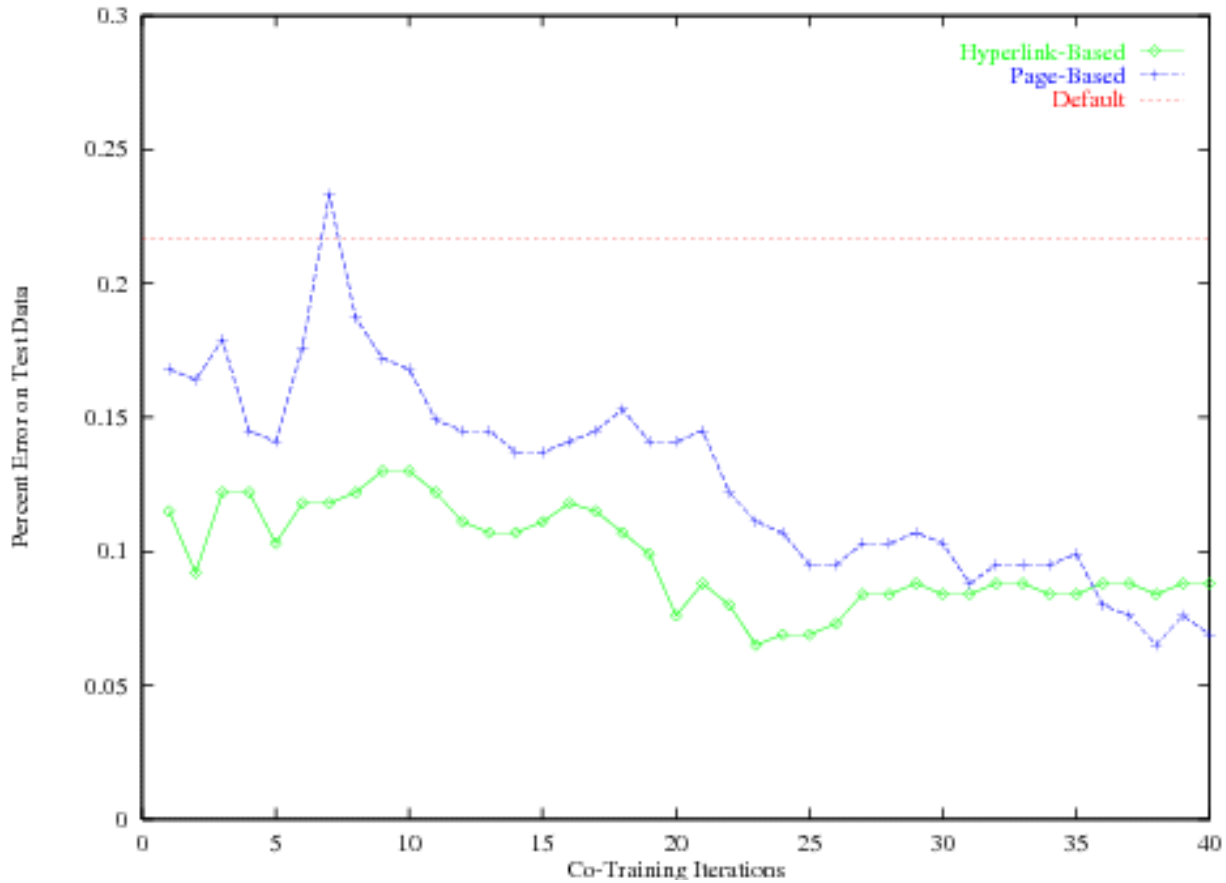
- Look through unlabeled data to find examples where one of h_i is confident but other is not.
- Have the confident h_i label it for algorithm A_{3-i}.

Original Application: Webpage classification

12 labeled examples, 1000 unlabeled

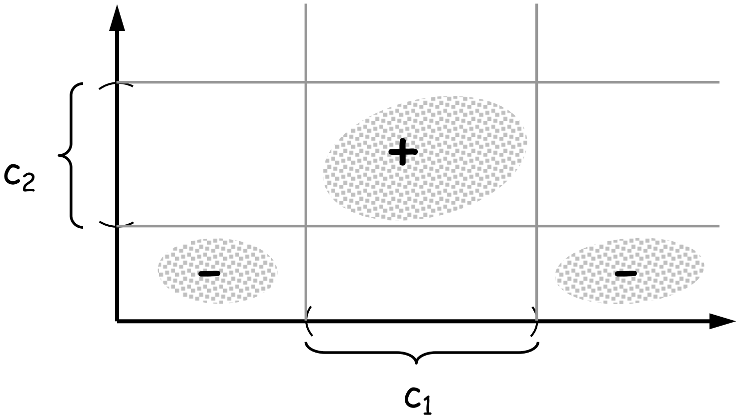
	Page-based	Hyperlink-based	Combined
Std. Supervised	12.9	12.4	11.1
Co-training	6.2	11.6	5.0
Just say neg	22	22	22

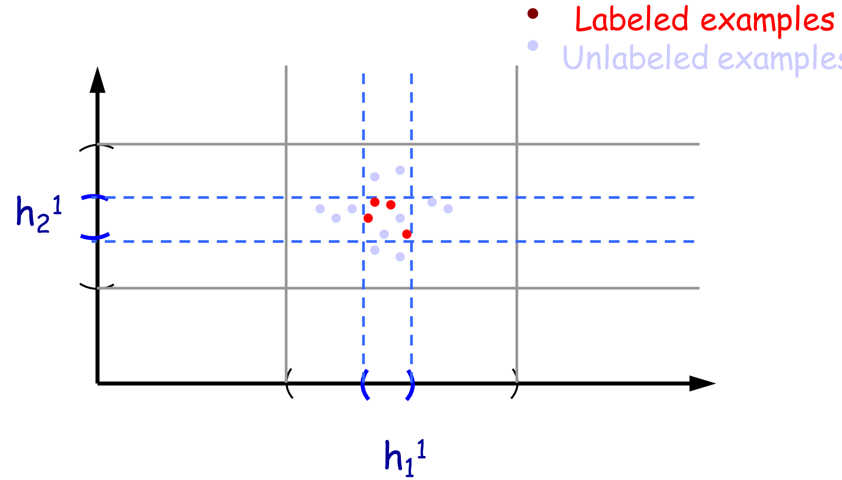
(sample run)



Iterative Co-Training

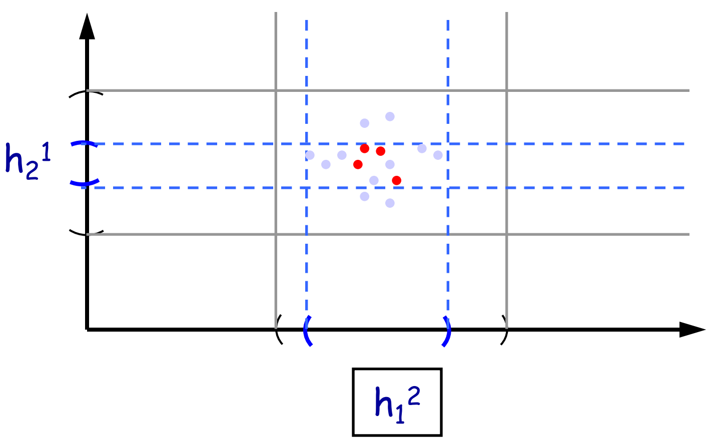
A Simple Example: Learning Intervals

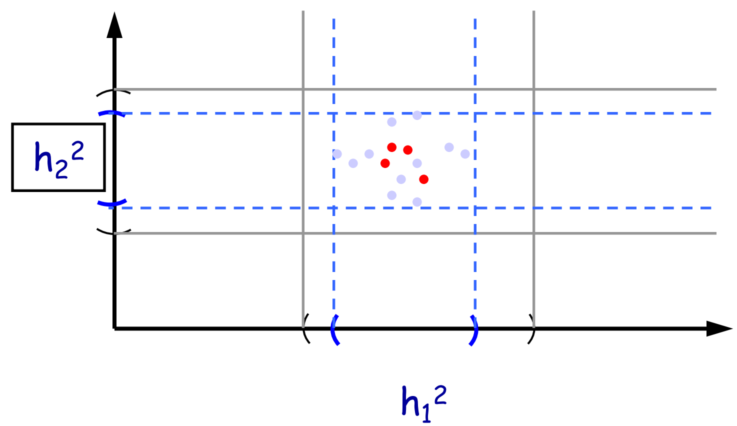




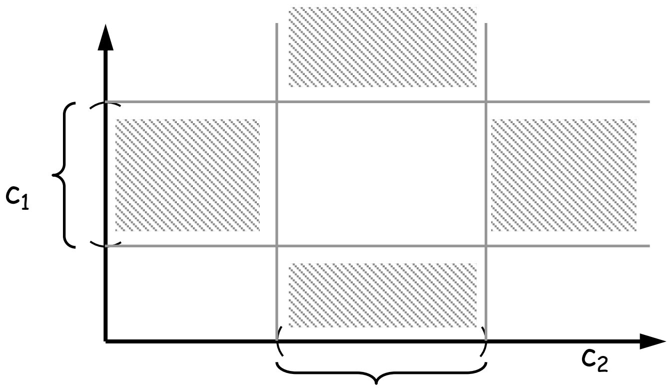
Use labeled data to learn h_1^1 and h_2^1

Use unlabeled data to bootstrap



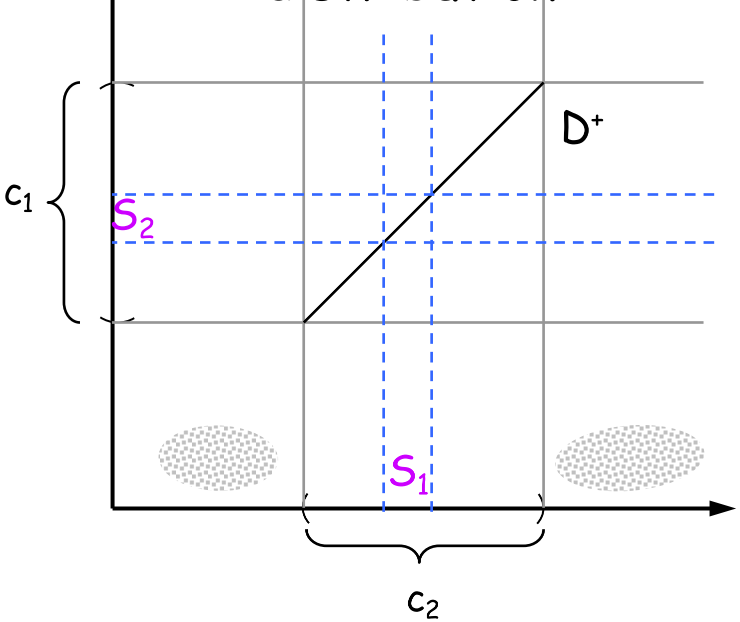


Expansion, Examples: Learning Intervals

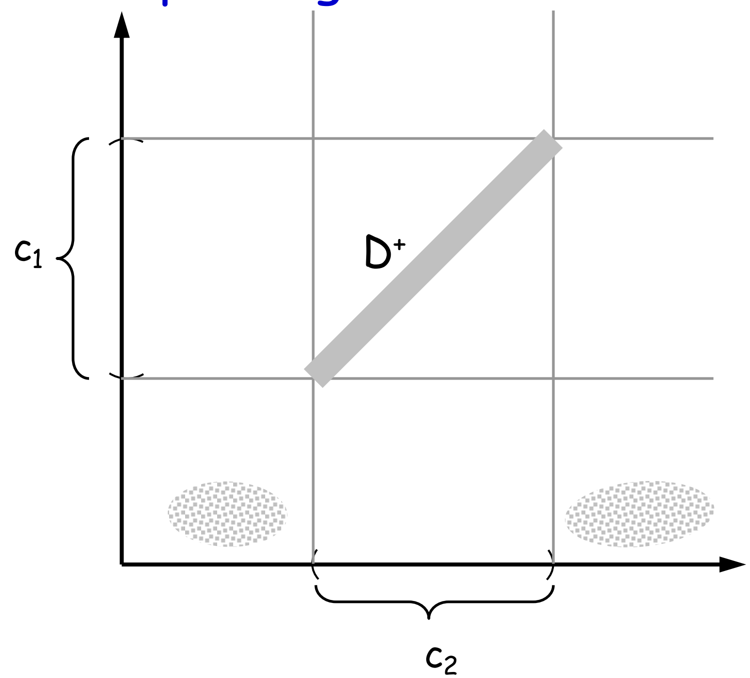


Consistency: zero probability mass in the regions

Non-expanding (non-helpful) distribution



Expanding distribution



Co-training: Theoretical Guarantees

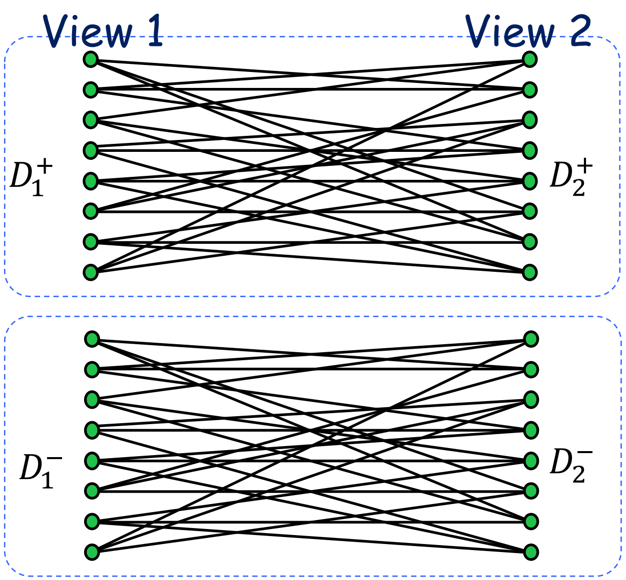
What properties do we need for co-training to work well?

We need assumptions about:

1. the underlying data distribution
2. the learning algos on the two sides

[Blum & Mitchell, COLT '98]

1. Independence given the label
2. Alg. for learning from random noise.



[Balcan, Blum, Yang, NIPS 2004]

1. Distributional expansion.
2. Alg. for learning from positve data only.

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

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

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

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

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

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

- $l(h(x_i), y_i)$ 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.,

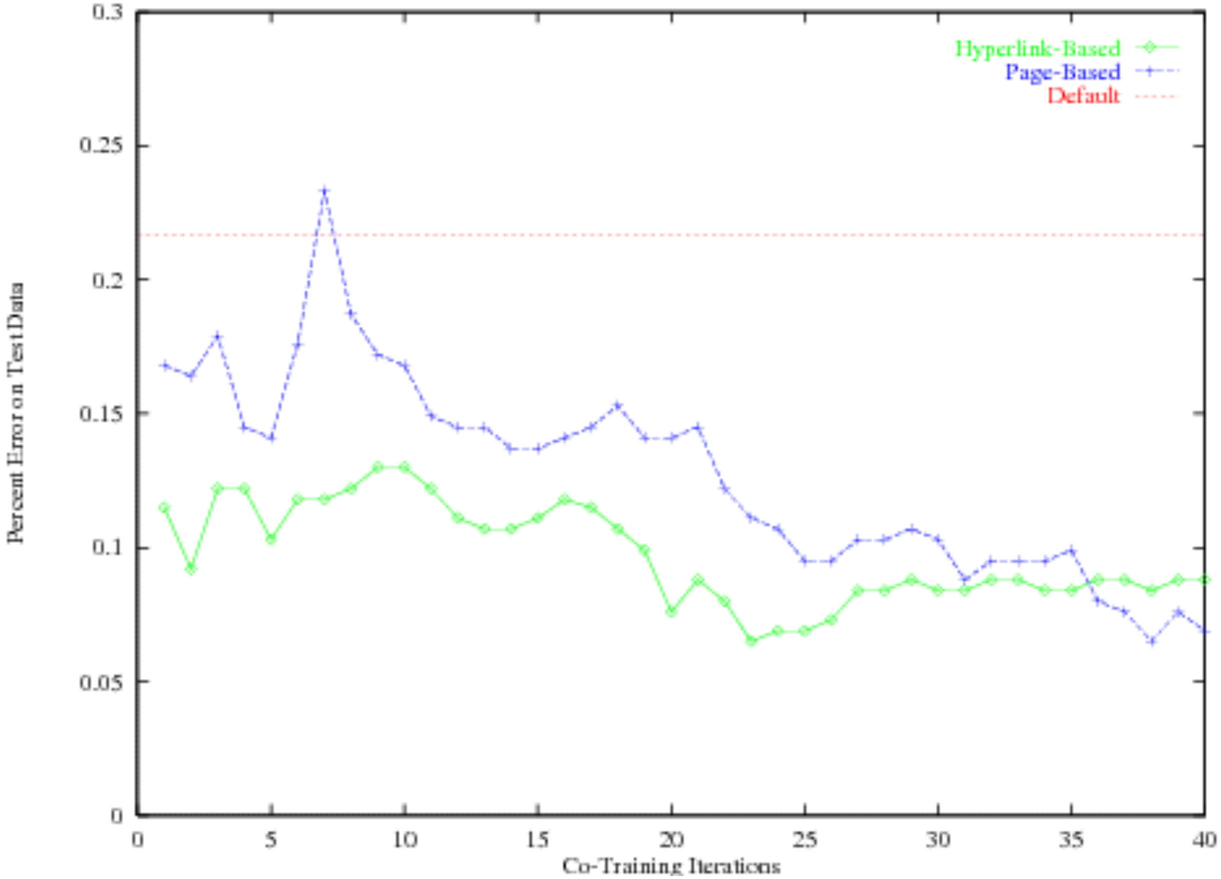
P. Bartlett, D. Rosenberg, AISTATS 2007; K. Sridharan, S. Kakade, COLT 2008

Original Application: Webpage classification

12 labeled examples, 1000 unlabeled

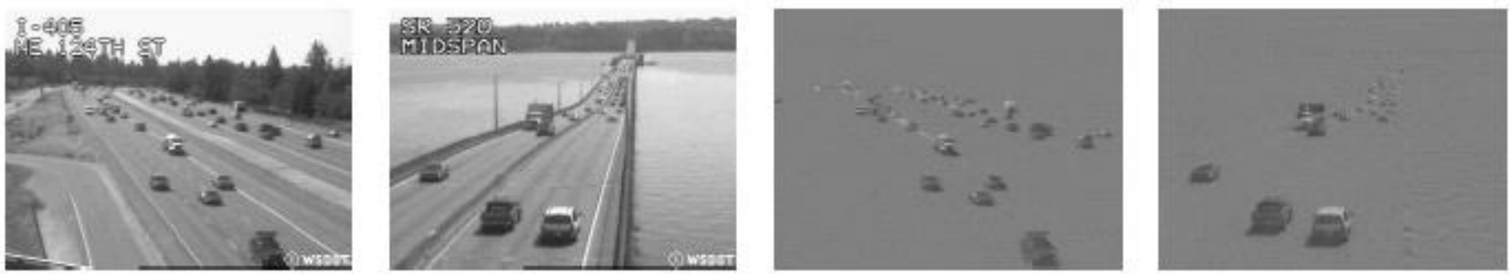
	Page-based	Hyperlink-based	Combined
Std. Supervised	12.9	12.4	11.1
Co-training	6.2	11.6	5.0
Just say neg	22	22	22

(sample run)



Many Other Applications

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"

...

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).

  - Transductive SVM [Joachims '99]
  - Co-training [Blum & Mitchell '98]
  - Graph-based methods [B&C01], [ZGL03]

Additional Material on Graph Based Methods

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]:

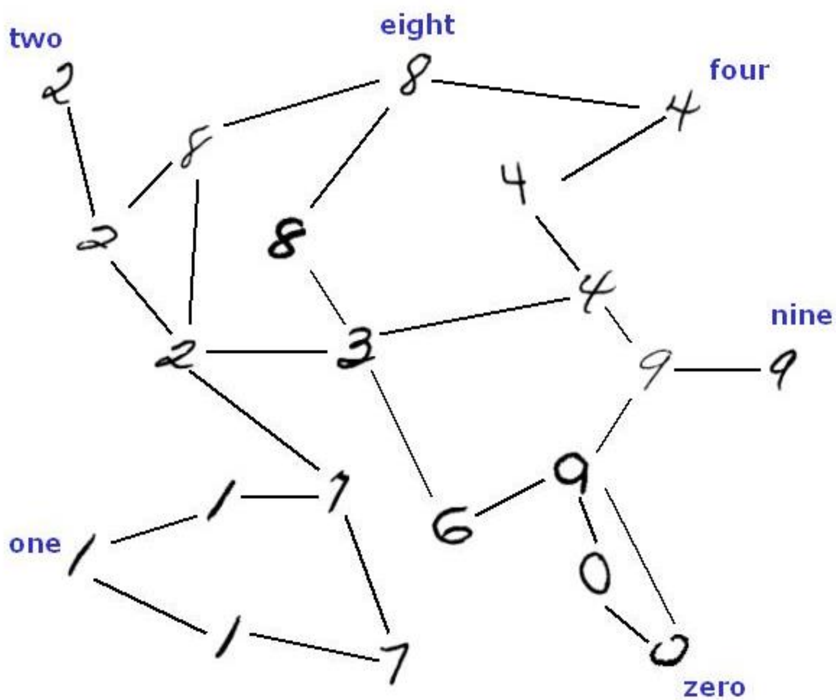
not similar

'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

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

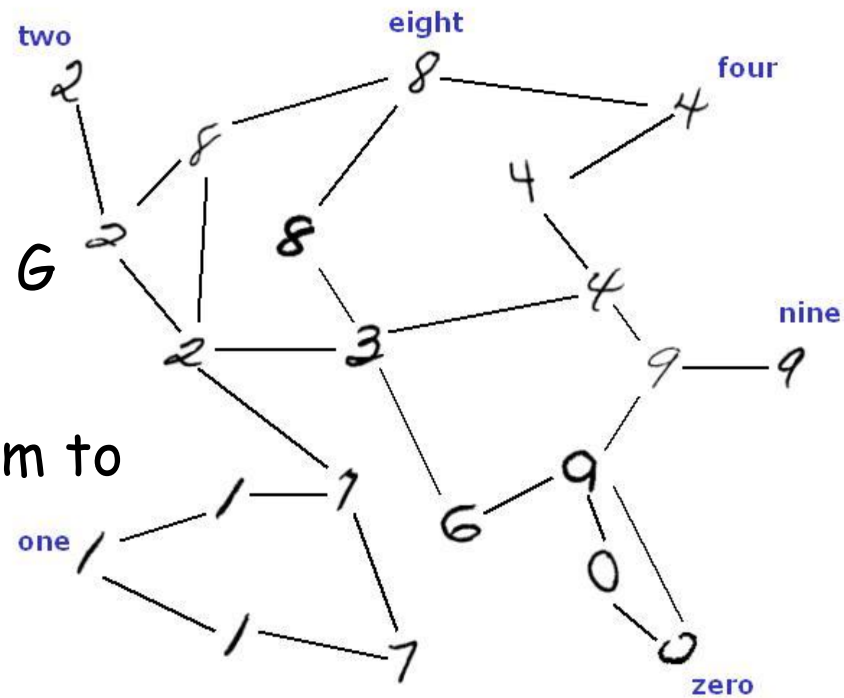
Person Identification in Webcam Images: An Application of Semi-Supervised Learning. [Balcan,Blum,Choi, Lafferty, Pantano, Rwebangira, Xiaojin Zhu], ICML 2005 Workshop on Learning with Partially Classified Training Data.

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 graph partitioning algorithm to separate the graph into pieces.

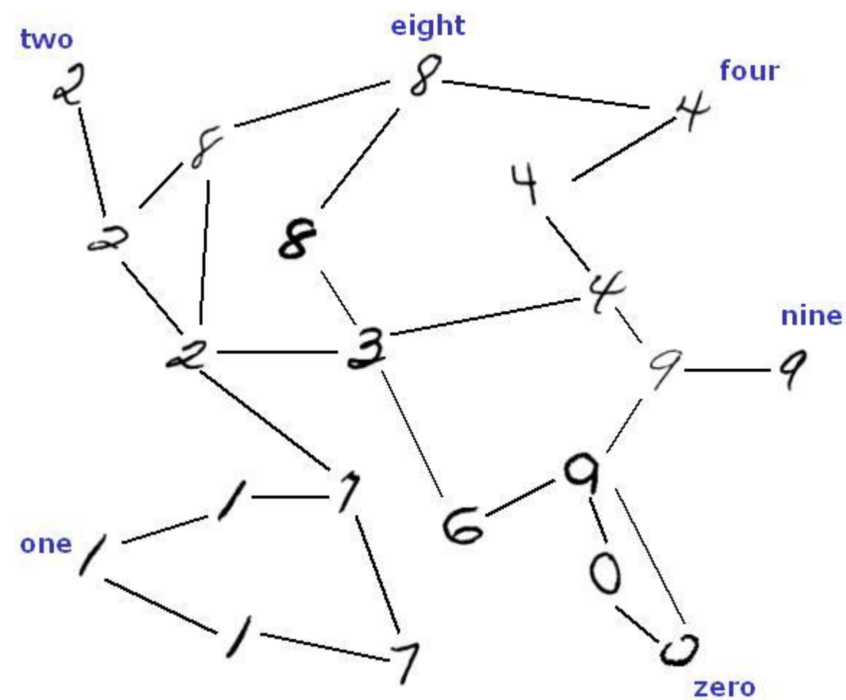
Several methods:

- Minimum/Multiway cut [Blum&Chawla01]
- Minimum "soft-cut" [ZhuGhahramaniLafferty'03]
- Spectral partitioning
- …

Minimum/Multiway Cut[Blum&Chawla01]

Objective: Solve for labels on unlabeled points that minimize total weight of edges whose endpoints have different labels.
(i.e., the total weight of bad edges)

- If just two labels, can be solved efficiently using max-flow min-cut algorithms
  - Create super-source s connected by edges of weight ∞ to all + labeled pts.
  - Create super-sink t connected by edges of weight ∞ to all $-$ labeled pts.
  - Find minimum-weight s-t cut



Minimum "soft cut"

[ZhuGhahramaniLafferty'03]

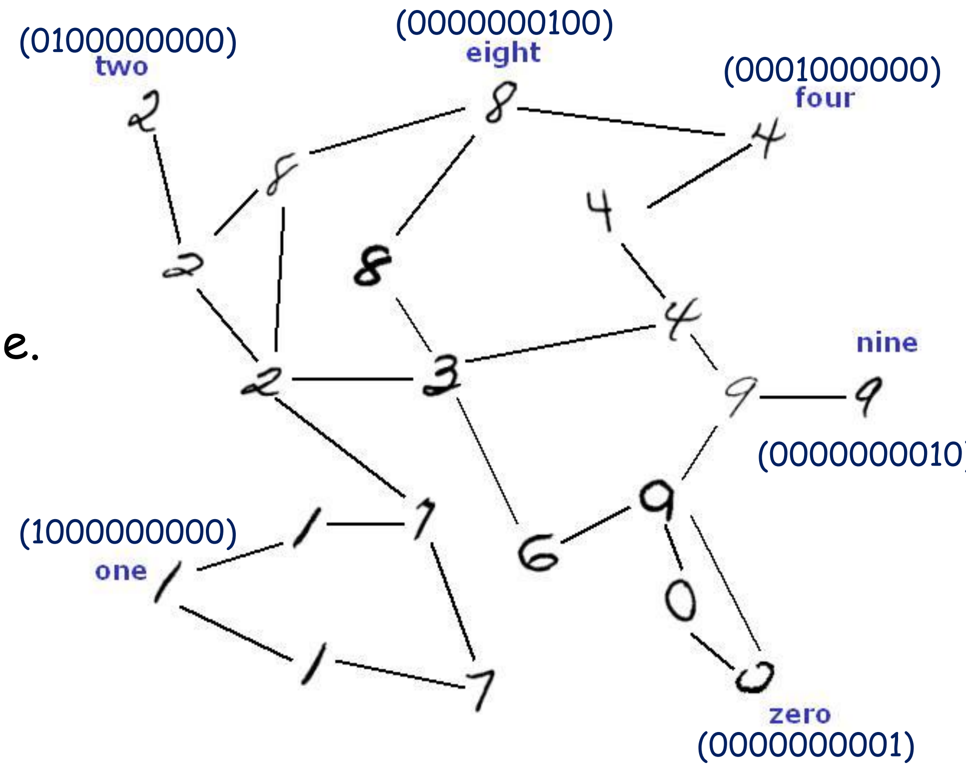
Objective Solve for probability vector over labels f_i on each unlabeled point i.

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

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

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

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



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 σ