Supervised Learning and Semi-Supervised Learning

Maria-Florina Balcan

10/30/2013

Supervised Learning: Formalization (PAC)

- X - instance space
- $S_l=\{(x_i, y_i)\}$ - labeled examples drawn i.i.d. from some distr. D over X and labeled by some target concept c^*
  - labels $\in \{-1,1\}$ - binary classification

- Algorithm A PAC-learns concept class C if for any target c* in C, any distrib. D over X, any $\varepsilon, \delta > 0$:
  - A uses at most poly($n,1/\varepsilon,1/\delta$,size(c*)) examples and running time.
  - With probab. $1-\delta$, A produces h in C of error at $\leq \varepsilon$.

Supervised Learning, Big Questions

- Algorithm Design
  - How might we automatically generate rules that do well on observed data?

- Sample Complexity/Confidence Bound
  - What kind of confidence do we have that they will do well in the future?

Sample Complexity: Uniform Convergence

Finite Hypothesis Spaces

Realizable Case

Theorem After

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

examples, with probab. $1-\delta$, all $h \in C$ with $err(h) \geq \varepsilon$ have $\hat{err}(h) > 0$.

Agnostic Case

- What if there is no perfect h?

Theorem After m examples, with probab. $\geq 1-\delta$, all $h \in C$ have $|err(h) - \hat{err}(h)| < \varepsilon$, for

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

Sample Complexity: Uniform Convergence

Infinite Hypothesis Spaces

- C[S] – the set of splittings of dataset S using concepts from C.
- C[m] - maximum number of ways to split m points using concepts in C; i.e. $C[m] = \max_{|S|=m} |C[S]|$

- C[m,D] - expected number of splits of m points from D with concepts in C.

- Fact #1: previous results still hold if we replace |C| with C[2m].
- Fact #2: can even replace with C[2m,D].

Sample Complexity: Uniform Convergence

Infinite Hypothesis Spaces

For instance:

Theorem For any class C, distrib. D, if the number of labeled examples seen m_l satisfies

$$m_l \geq \frac{2}{\varepsilon}\left[\log_2(2C[2m]) + \log_2\left(\frac{1}{\delta}\right)\right]$$

then with probab. $1-\delta$, all $h \in C$ with $err(h) \geq \varepsilon$ have $\hat{err}(h) > 0$.

Sauer's Lemma, $C[m]=O(m^{VC\text{-}dim(C)})$ implies:

Theorem

$$m_l = O\left(\frac{1}{\varepsilon}\left[VCdim(C)\log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right]\right)$$

labeled examples are sufficient so that with probab. $1-\delta$, all $h \in C$ with $err(h) \geq \varepsilon$ have $\hat{err}(h) > 0$.

Sample Complexity: ε-Cover Bounds

- C_ε is an ε-cover for C w.r.t. D if for every $h \in C$ there is a $h' \in C_\varepsilon$ which is ε-close to h.

- To learn, it's enough to find an ε-cover and then do empirical risk minimization w.r.t. the functions in this cover.

- In principle, in the realizable case, the number of labeled examples we need is

$$O\left(\frac{1}{\varepsilon}\left[\ln(|C_{\epsilon/4}|) + \ln\left(\frac{1}{\delta}\right)\right]\right)$$

Usually, for fixed distributions.

Sample Complexity: ε-Cover Bounds

Can be much better than Uniform-Convergence bounds!

Simple Example (Realizable case)

- $X=\{1, 2, \ldots, n\}$, $C = C_1 \cup C_2$, D= uniform over X.
- C_1 - the class of all functions that predict positive on at most $\varepsilon \cdot n/4$ examples.
- C_2 - the class of all functions that predict negative on at most $\varepsilon \cdot n/4$ examples.

If the number of labeled examples $m_l < \varepsilon \cdot n/4$, don't have uniform convergence yet.

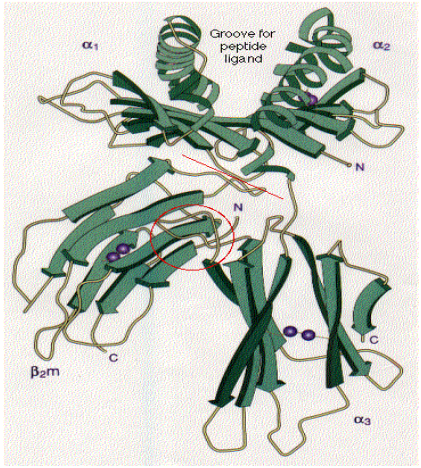
The size of the smallest $\varepsilon/4$-cover is 2, so we can learn with only $O(1/\varepsilon)$ labeled examples.

In fact, since the elements of this cover are far apart, much fewer examples are sufficient.

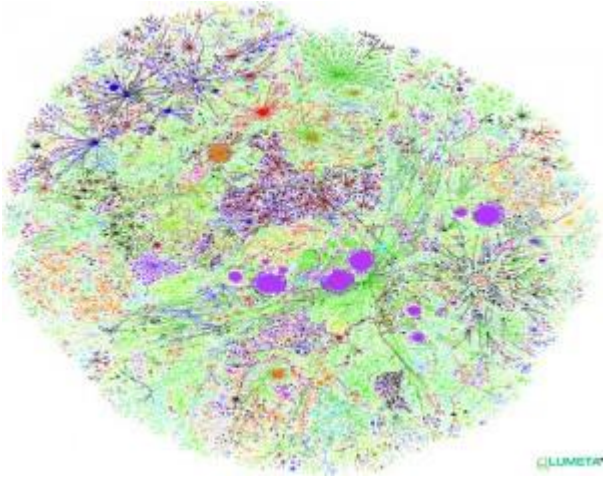
Classic Paradigm Insufficient Nowadays

Modern applications: massive amounts of raw data.

Only a tiny fraction can be annotated by human experts.



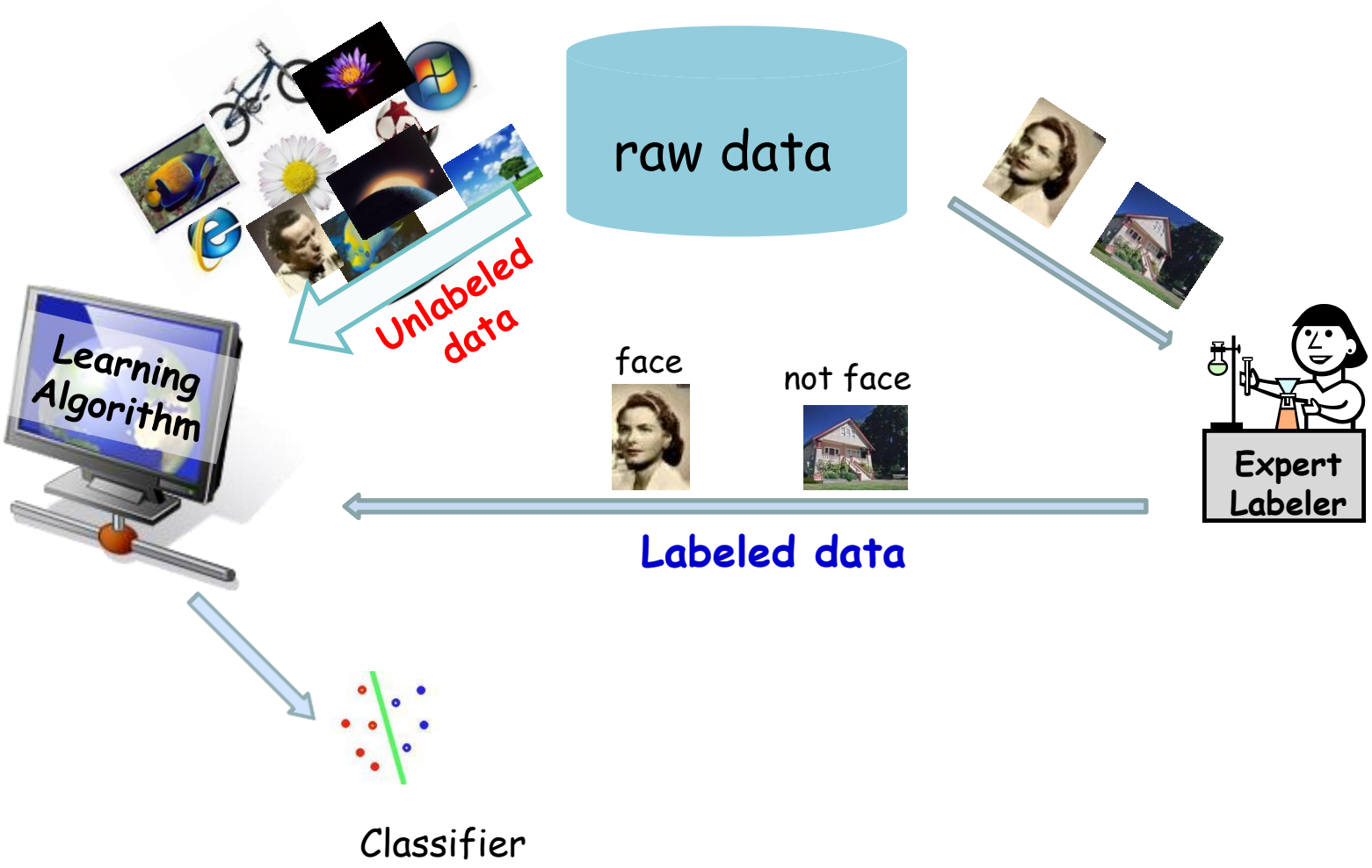
Protein sequences



Billions of webpages

Images

Semi-Supervised Learning



Semi-Supervised Learning

Hot topic in recent years in Machine Learning.

- Many applications have lots of unlabeled data, but labeled data is rare or expensive:
  - Web page, document classification
  - OCR, Image classification

Workshops [ICML '03, ICML' 05]

Books: Semi-Supervised Learning, MIT 2006

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

Combining Labeled and Unlabeled Data

- Several methods have been developed to try to use unlabeled data to improve performance, e.g.:
  - Transductive SVM [Joachims '98]
  - Co-training [Blum & Mitchell '98], [BBY04]
  - Graph-based methods [Blum & Chawla01], [ZGL03]

- Augmented PAC model for SSL [Balcan & Blum '05]

  $S_u=\{x_i\}$ - unlabeled examples drawn i.i.d. from D

  $S_l=\{(x_i, y_i)\}$ – labeled examples drawn i.i.d. from D and labeled by some target concept c^*.

Different model: the learner gets to pick the examples to Labeled – Active Learning.

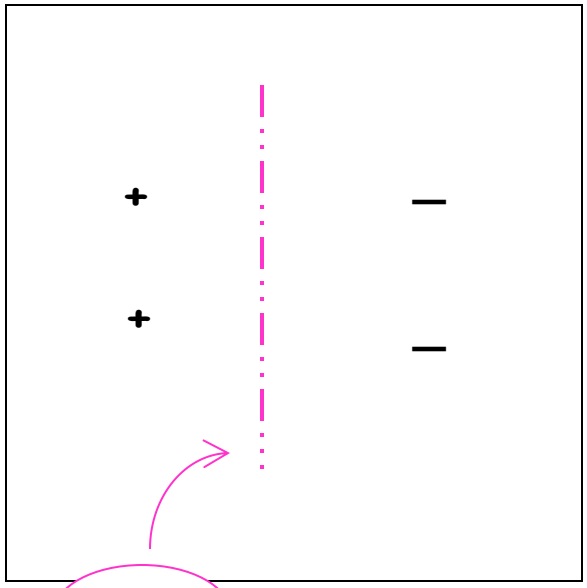
Can we extend the PAC/SLT models to deal with Unlabeled Data?

- PAC/SLT models – nice/standard models for learning from labeled data.

- Goal – extend them naturally to the case of learning from both labeled and unlabeled data.
  - Different algorithms are based on different assumptions about how data should behave.
  - Question – how to capture many of the assumptions typically used?

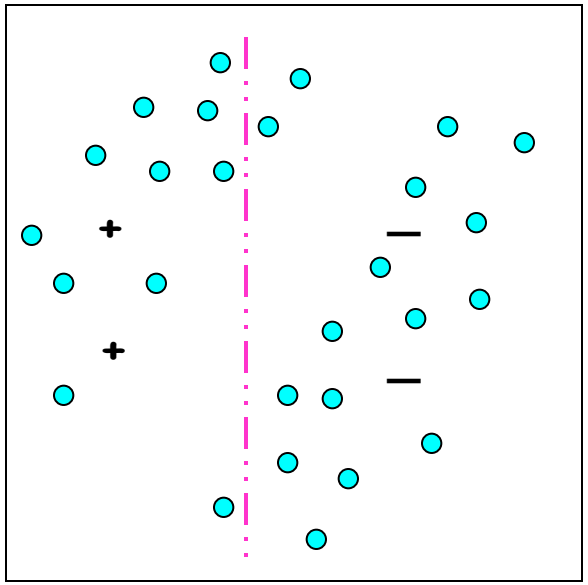
Example of "typical" assumption: Margins

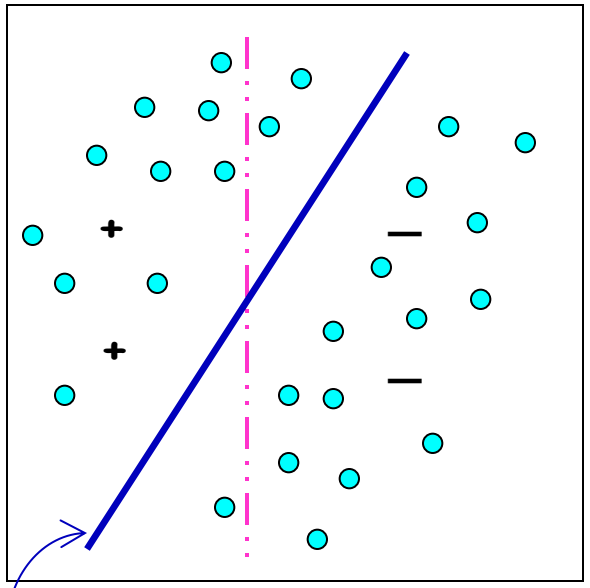
- The separator goes through low density regions of the space/large margin.
  - assume we are looking for linear separator
  - belief: should exist one with large separation



SVM

Labeled data only

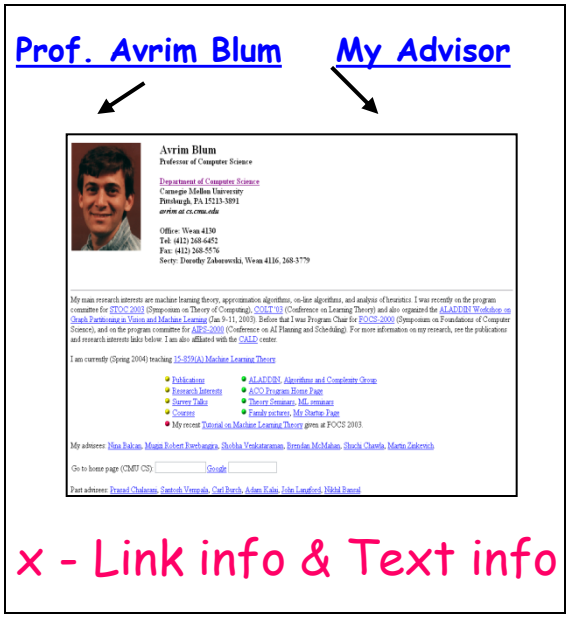


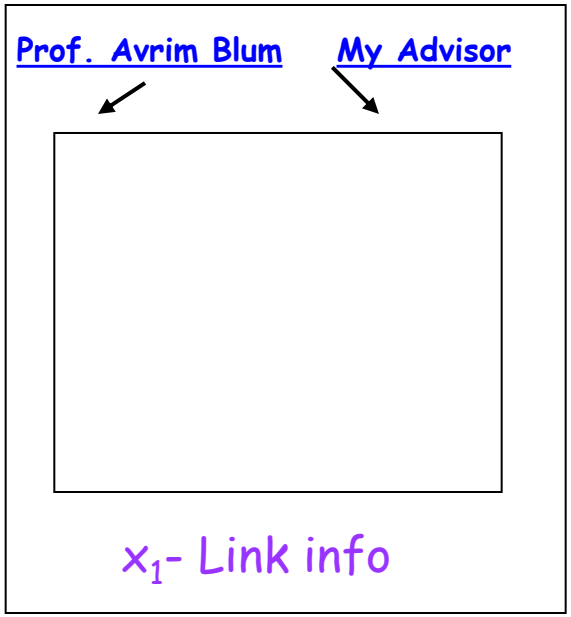


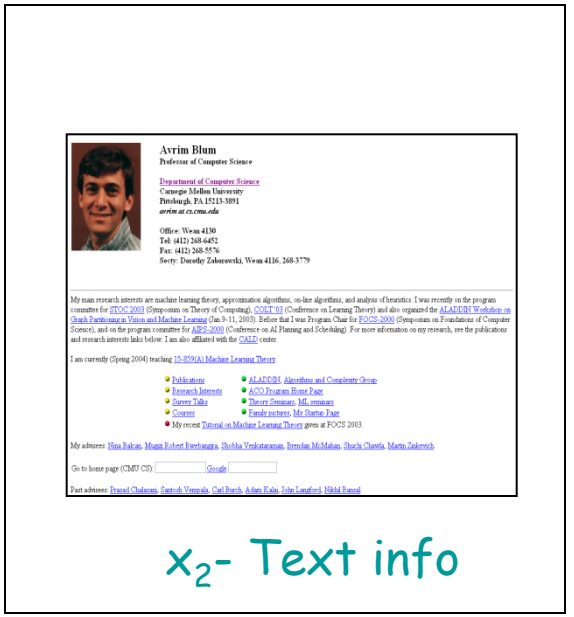
Transductive SVM

Another Example: Self-consistency

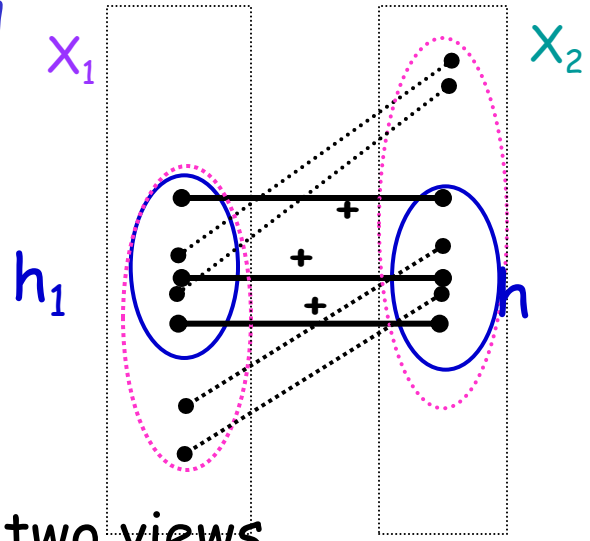
- Agreement between two parts : co-training.
  - examples contain two sufficient sets of features, i.e. an example is $x=\langle x_1, x_2 \rangle$ and the belief is that the two parts of the example are consistent, i.e. $\exists\ c_1, c_2$ such that $c_1(x_1)=c_2(x_2)=c^*(x)$
  - for example, if we want to classify web pages: $x = \langle x_1, x_2 \rangle$







Iterative Co-Training



Works by using unlabeled data to propagate learned information.

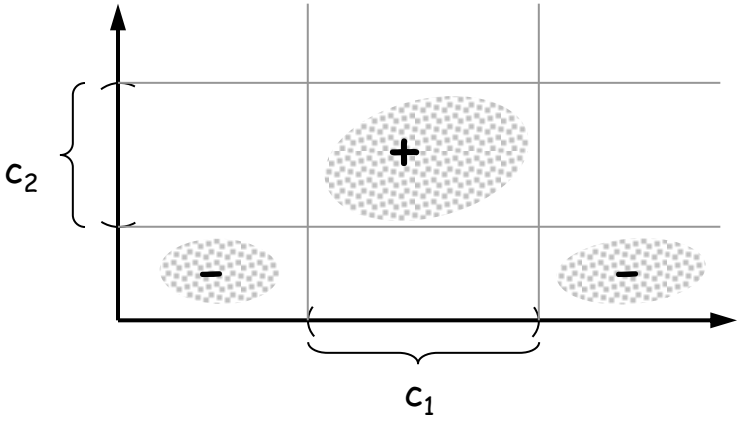
- Have learning algos A_1, A_2 on each of the two views.
- Use labeled data to learn two initial hyp. h_1, h_2.

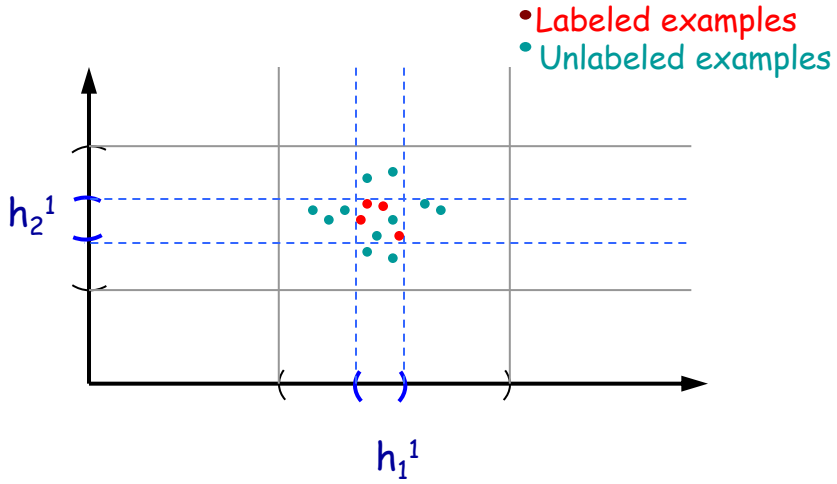
Repeat

- 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}.

Iterative Co-Training

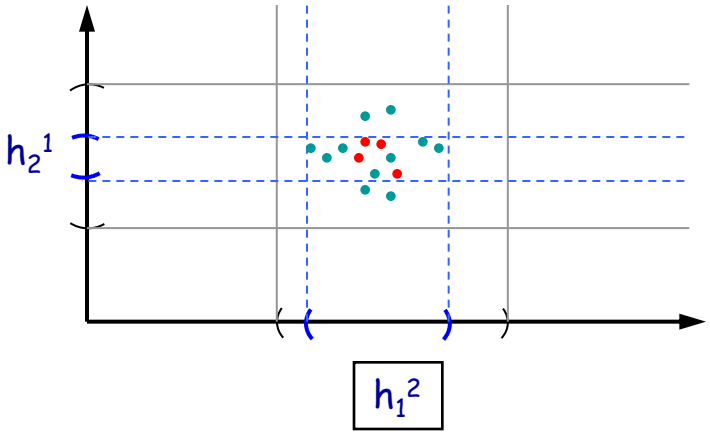
A Simple Example: Learning Intervals

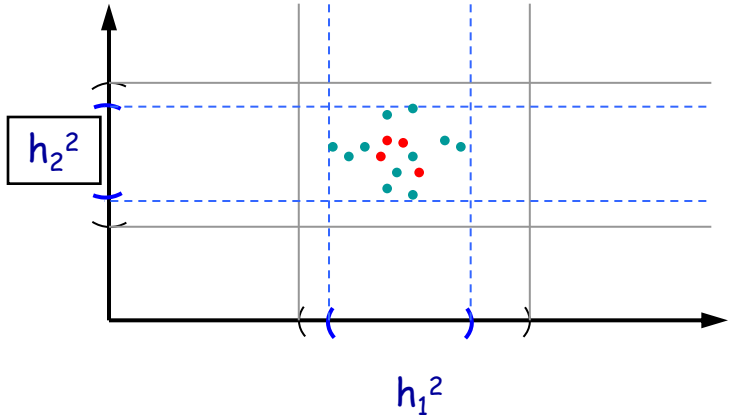




Use labeled data to learn h_1^1 and h_2^1

Use unlabeled data to bootstrap





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 algorithms 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.

Problems thinking about SSL in the PAC model

- PAC model talks of learning a class C under (known or unknown) distribution D.
  - Not clear what unlabeled data can do for you.
  - Doesn't give you any info about which $c \in C$ is the target function.

- Can we extend the PAC model to capture these (and more) uses of unlabeled data?

  - Give a unified framework for understanding when and why unlabeled data can help.

New discriminative model for SSL

$S_u=\{x_i\}$ - x_i i.i.d. from D and $S_l=\{(x_i, y_i)\}$ -x_i i.i.d. from D, $y_i = c^*(x_i)$.

Problems with thinking about SSL in standard WC models

- PAC or SLT: learn a class C under (known or unknown) distribution D.
  - a complete disconnect between the target and D
- Unlabeled data doesn't give any info about which $c \in C$ is the target.

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.

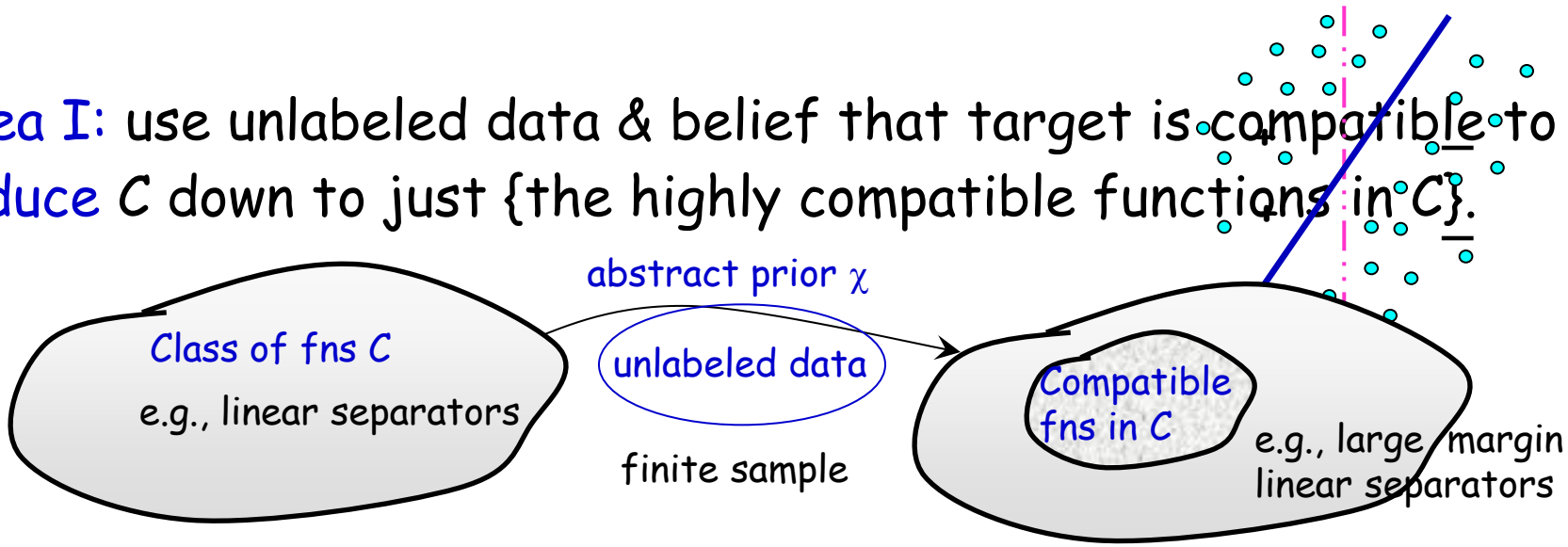
New model for SSL, Main Ideas

Augment the notion of a concept class C with a notion of compatibility χ between a concept and the data distribution.

"learn C" becomes "learn (C,χ)" (learn class C under χ)

Express relationships that target and underlying distr. possess.

Idea I: use unlabeled data & belief that target is compatible to reduce C down to just {the highly compatible functions in C}.



Idea II: degree of compatibility estimated from a finite sample.