Active Learning

Digging into Data: Jordan Boyd-Graber

University of Maryland

April 22, 2013

UNIVERSITY OF MARYLAND 18 56

COLLEGE OF INFORMATION STUDIES

(Slides Borrowed from John Langford)

Homework 2: Feature Engineering

- Length of the sentence
- Trope-specific information (not scored)
- Specific words
- Proper nouns
- Dictionaries
- Show metadata
  - All Shows
  - IMDB
  - Wikipedia

Homework 2: Results

- Unigram baseline: 58%
- Bigram baseline: 58%
- Adding dictionary: 52% (Huh?)
- Country: 61%
- Number of Episodes: 62%
- Length of Show (e.g. 30 min vs. 60 min): 55% (Arrested Development)
- Genre: 68%

Exploiting unlabeled data

A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

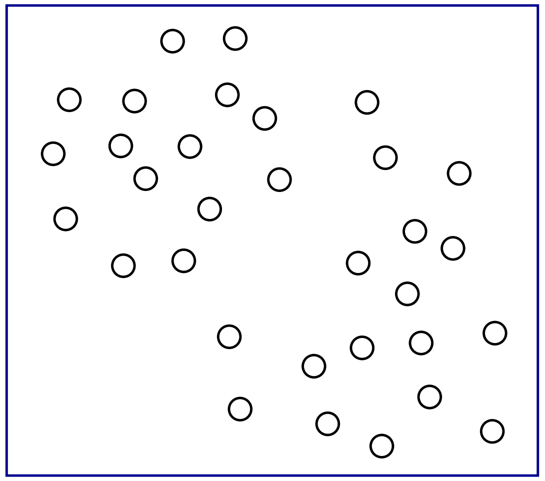
But labeling can be expensive.

Exploiting unlabeled data

A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

But labeling can be expensive.



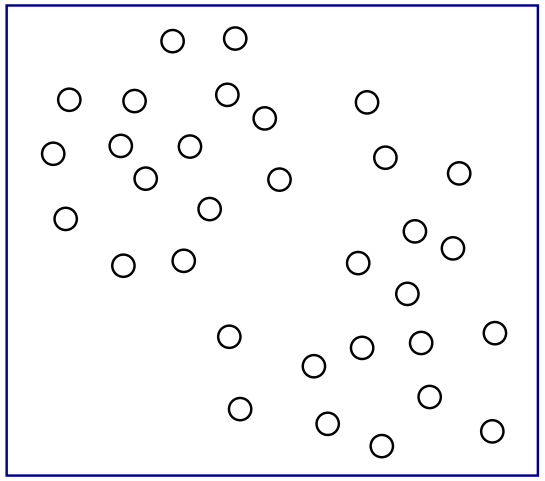
Unlabeled points

Exploiting unlabeled data

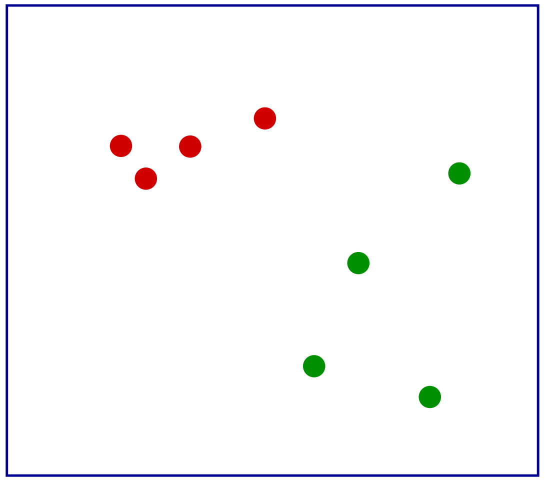
A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

But labeling can be expensive.



Unlabeled points



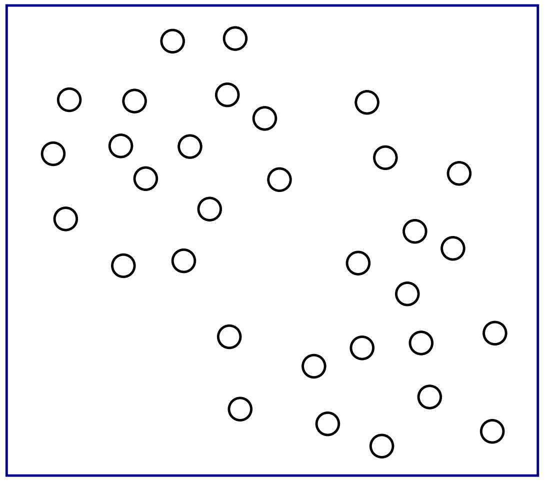
Supervised learning

Exploiting unlabeled data

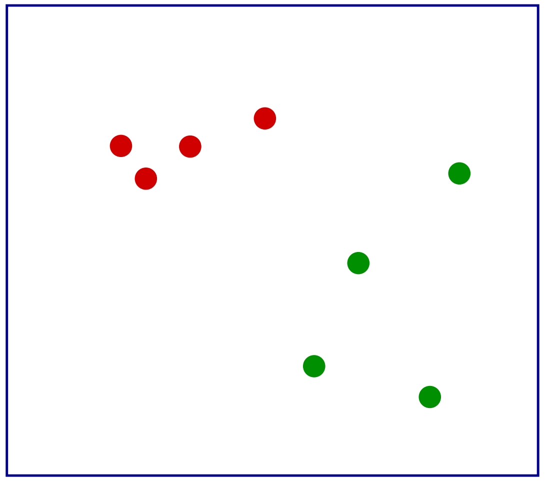
A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

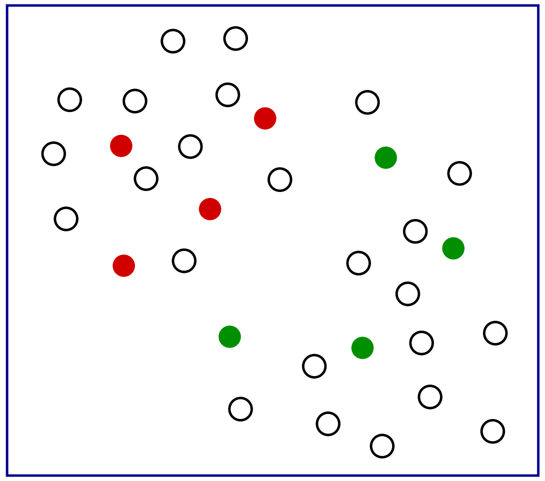
But labeling can be expensive.



Unlabeled points



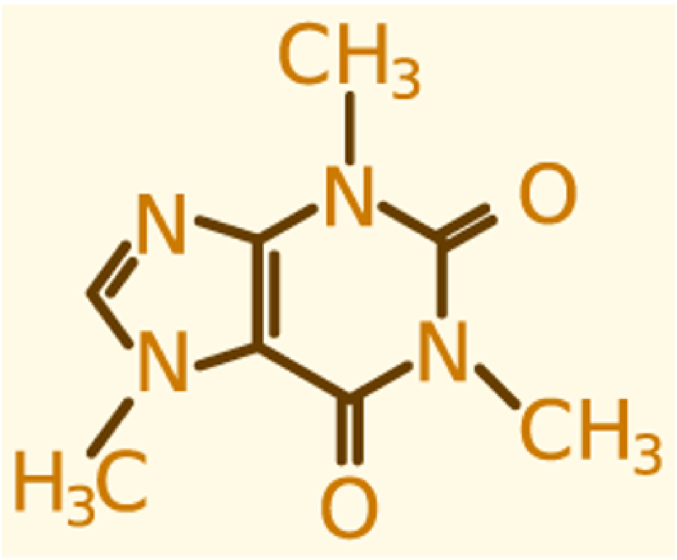
Supervised learning



Semisupervised and active learning

Active learning example: drug design [Warmuth et al 03]

Goal: find compounds which bind to a particular target

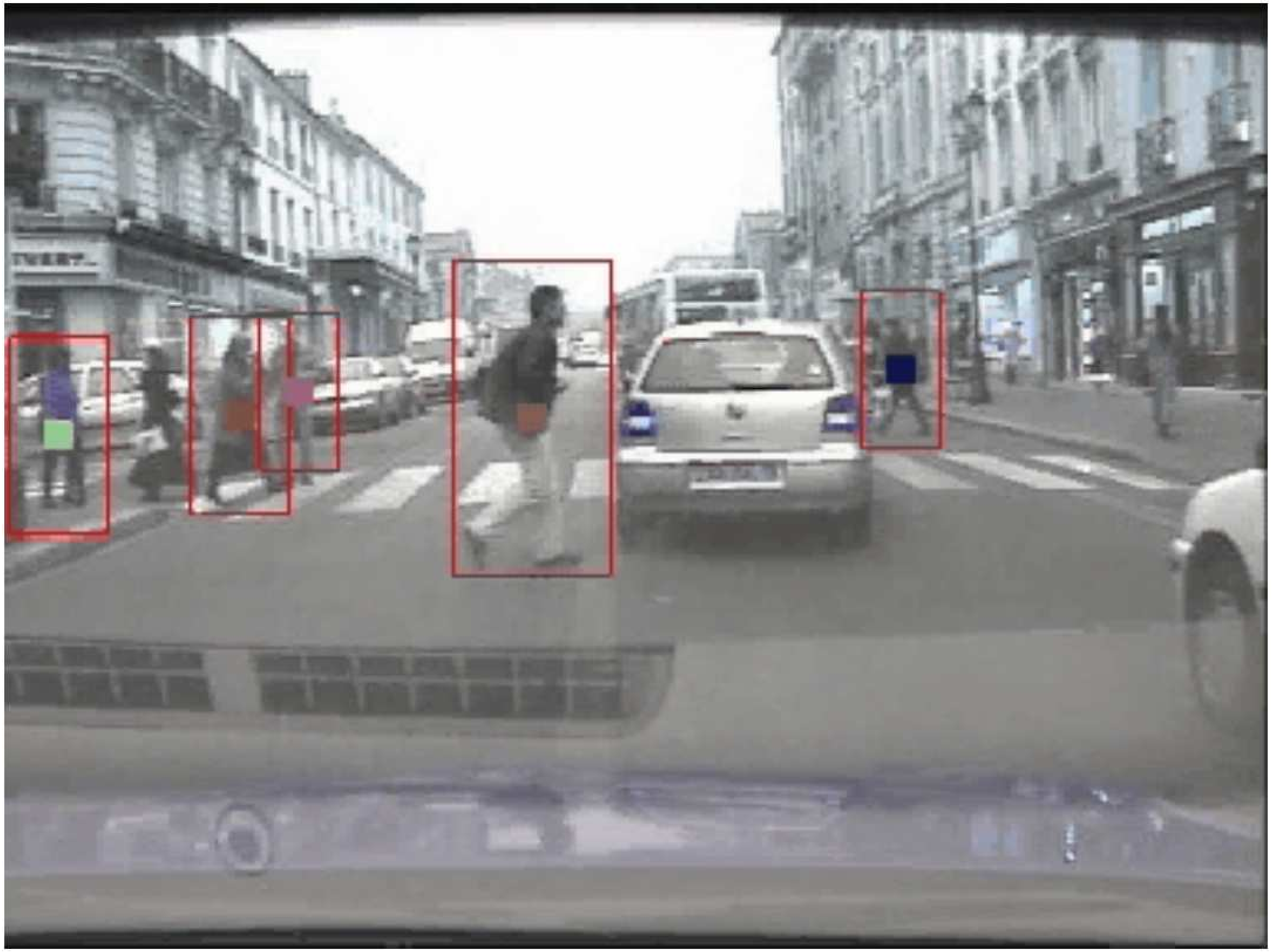


Large collection of compounds, from:

- vendor catalogs
- corporate collections
- combinatorial chemistry

unlabeled point	$\equiv$	description of chemical compound
label	$\equiv$	*active* (binds to target) vs. *inactive*
getting a label	$\equiv$	chemistry experiment

Active learning example: pedestrian detection [Freund et al 03]



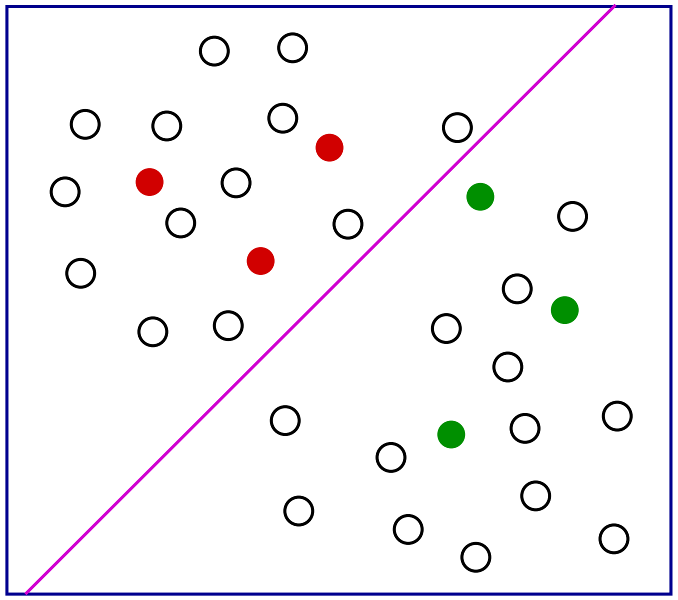
Typical heuristics for active learning

Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

- Fit a classifier to the labels seen so far
- Query the unlabeled point that is closest to the boundary (or most uncertain, or most likely to decrease overall uncertainty,...)



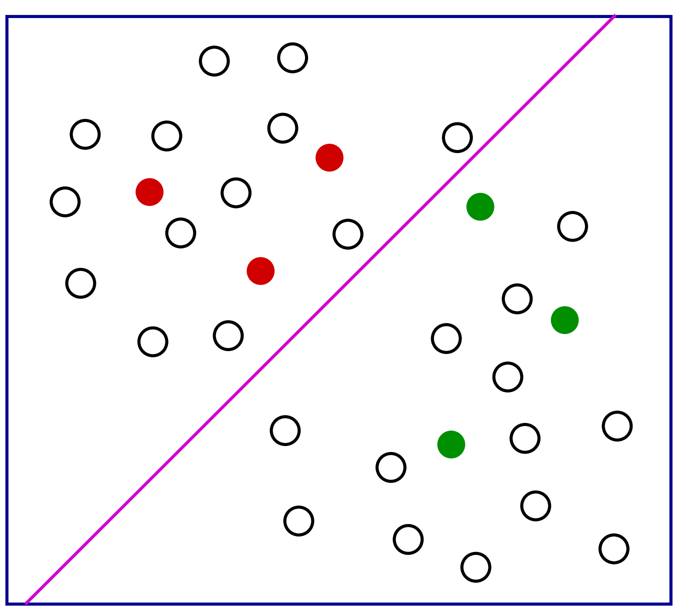
Typical heuristics for active learning

Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

Fit a classifier to the labels seen so far
Query the unlabeled point that is closest to the boundary
(or most uncertain, or most likely to decrease overall
uncertainty,...)



Biased sampling: the labeled points are not representative of the underlying distribution!

Can adaptive querying really help?

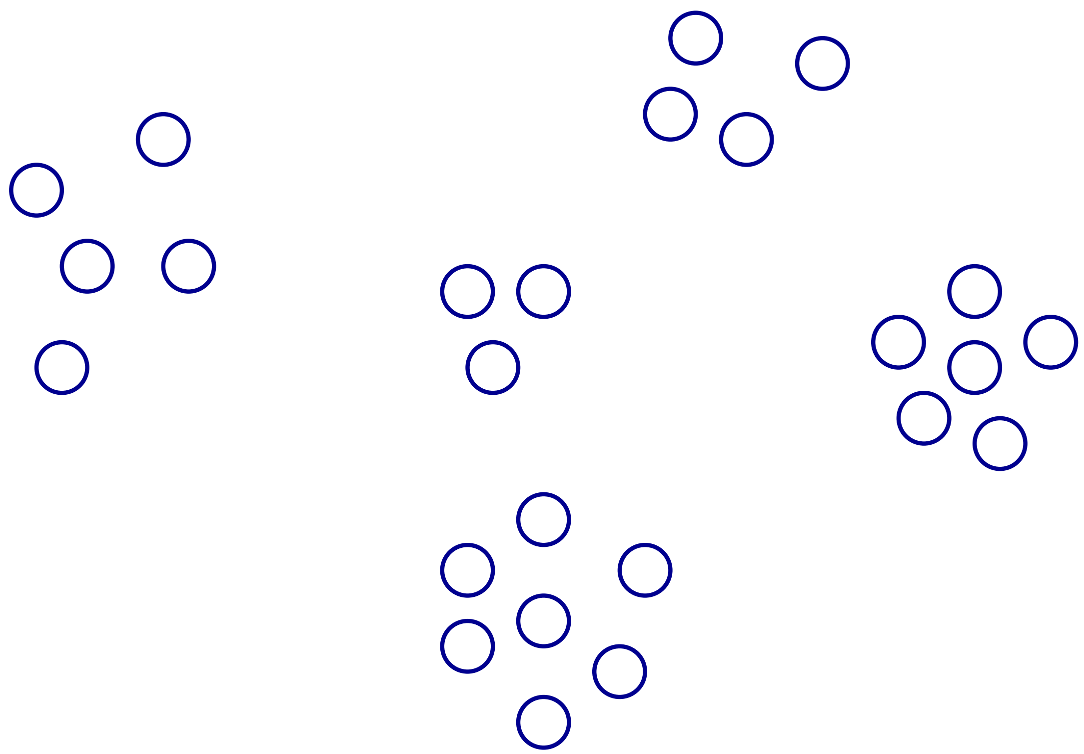
There are two distinct narratives for explaining how adaptive querying can help.

Case I: Exploiting (cluster) structure in data

Case II: Efficient search through hypothesis space

Case I: Exploiting cluster structure in data

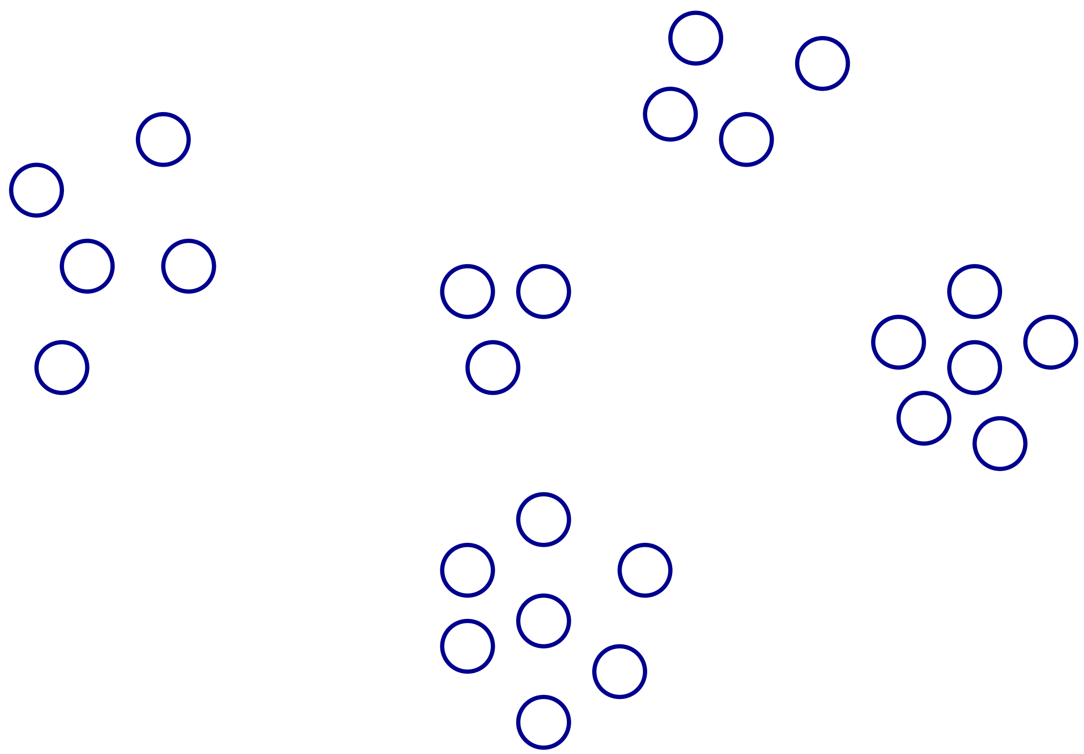
Suppose the unlabeled data looks like this.



Then perhaps we just need five labels!

Case I: Exploiting cluster structure in data

Suppose the unlabeled data looks like this.

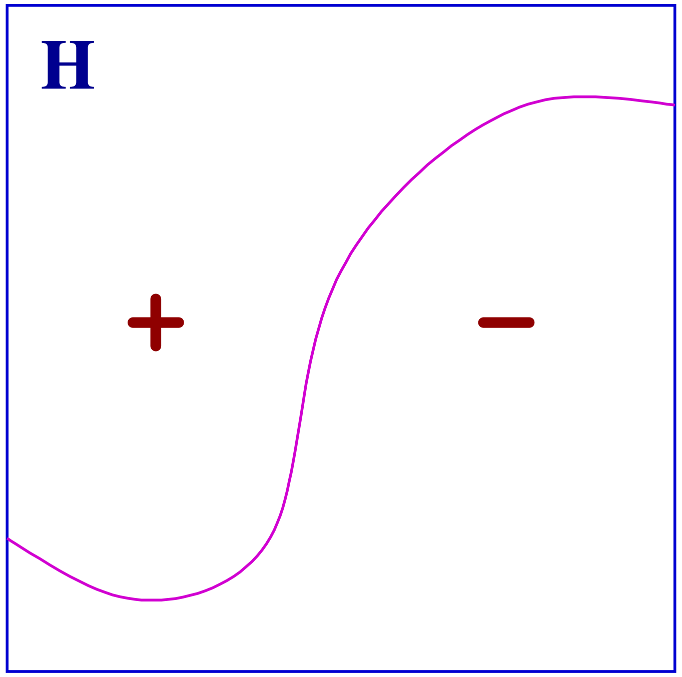


Then perhaps we just need five labels!

Challenges: In general, the cluster structure (i) is not so clearly defined and (ii) exists at many levels of granularity. And the clusters themselves might not be pure in their labels. How to exploit whatever structure happens to exist?

Case II: Efficient search through hypothesis space

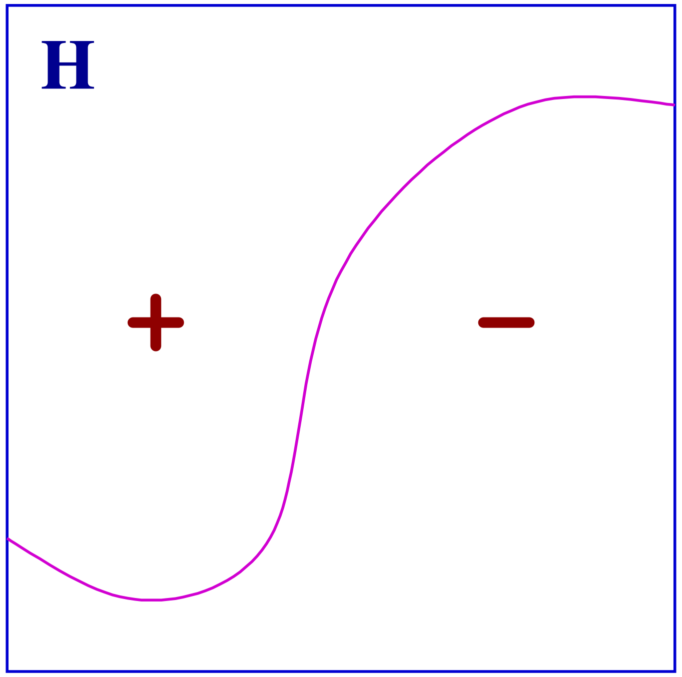
Ideal case: each query cuts the version space in two.



Then perhaps we need just $\log |H|$ labels to get a perfect hypothesis!

Case II: Efficient search through hypothesis space

Ideal case: each query cuts the version space in two.



Then perhaps we need just $\log |H|$ labels to get a perfect hypothesis!

Challenges: (1) Do there always exist queries that will cut off a good portion of the version space? (2) If so, how can these queries be found? (3) What happens in the nonseparable case?

Active Learning in Context

- Brings a lot of ideas together
  - entropy as a measure of uncertainty
  - information gain as a measure of how useful a feature is
  - can we use machine learning to guide annotation?
- Allows us to go from an **unsupervised** dataset to a **supervised** classification algorithm

Demo: Dualist

- System created by Burr Settles at Carnegie Mellon
- Under the hood (machine learning): naïve Bayes
- Under the hood (interface): play, java, and scala

Selecting documents to label

- Compute the probability of label assignments to a document
- Compute the entropy of that distribution

$$H_\theta(Y|x) = -\sum_j P_\theta(y_j|x) P_\theta(y_j|x) \tag{1}$$

- Show users the documents with **highest** entropy
- Why is this a good idea?

Selecting features to show to user

- For each class and each feature, compute the information gain of that feature

$$IG(w_k) = \sum_j P(w_k, y_j) \log \frac{P(w_k, y_j)}{P(w_k)P(y_j)} \tag{2}$$

- This looks slightly different from how we computed information gain, but it's the same thing
- Shows the features that best predict classes

What happens when you label a document?

- You have additional documents as training data
- . . . for a document the algorithm was most uncertain about
- This gives new conditional probability distributions

What happens when you label a feature?

- Instead of having "plus one" smoothing ... some features have a $1 + \alpha$ smoothing
- An extra bonus for features k that you think are important for a class j

Before

$$\theta_{jk} = \frac{1 + \sum_i P(y_j \,|\, x^{(i)}) f_k(x^{(i)})}{Z(f_k)} \tag{3}$$

After

$$\theta_{jk} = \frac{1 + \alpha + \sum_i P(y_j \,|\, x^{(i)}) f_k(x^{(i)})}{Z(f_k)} \tag{4}$$

(For the features that you've identified as important)