Active Learning

Evgeny Burnaev

Skoltech, Moscow, Russia

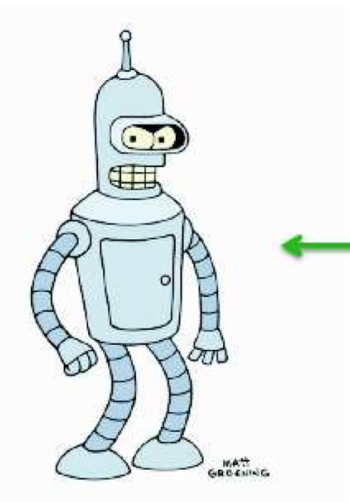
Outline

(Passive) Supervised Learning

raw unlabeled data

$x_1, x_2, x_3, \ldots$

random sample



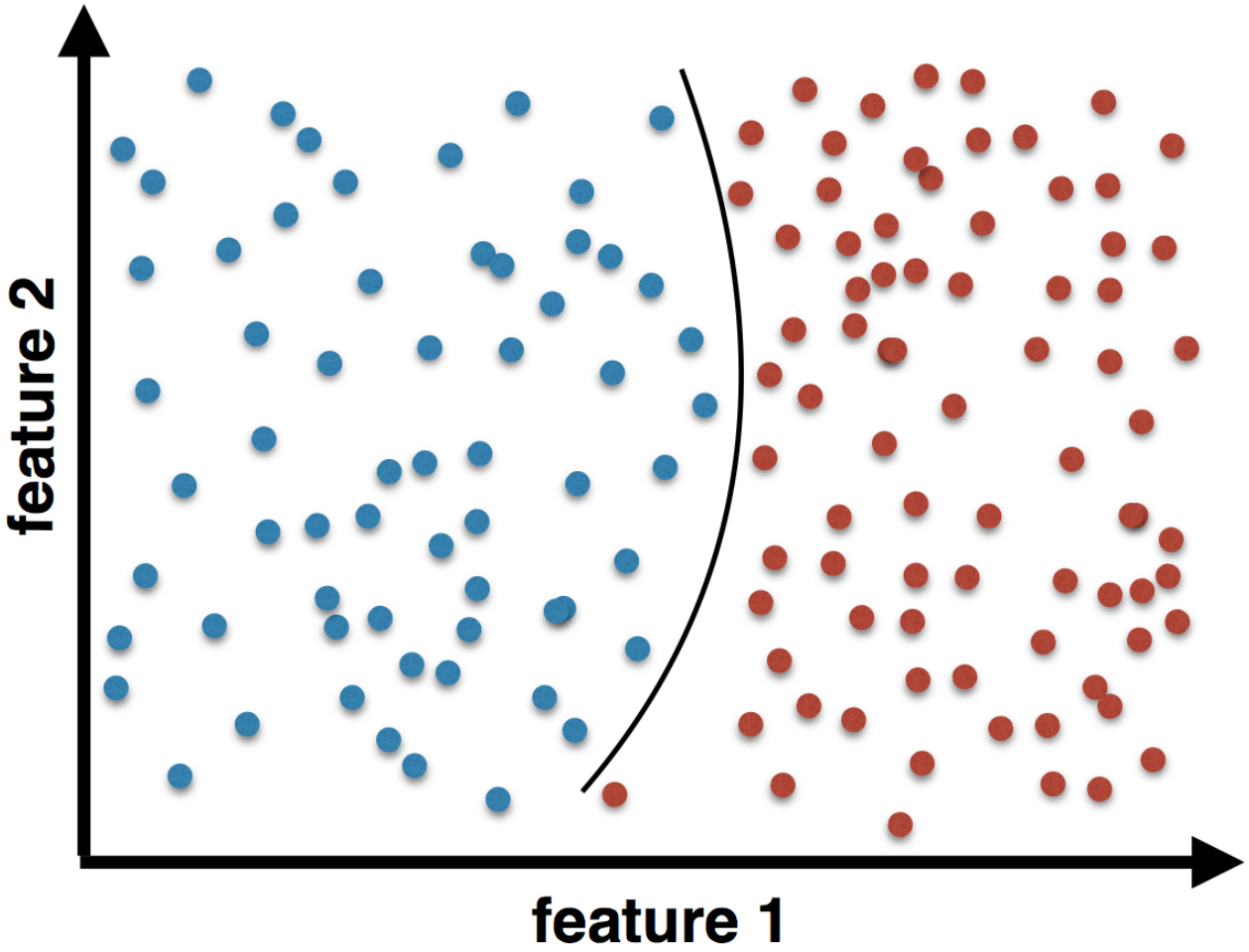
labeled training instances

$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \langle x_3, y_3 \rangle, \ldots$

supervised learner
induces a classifier

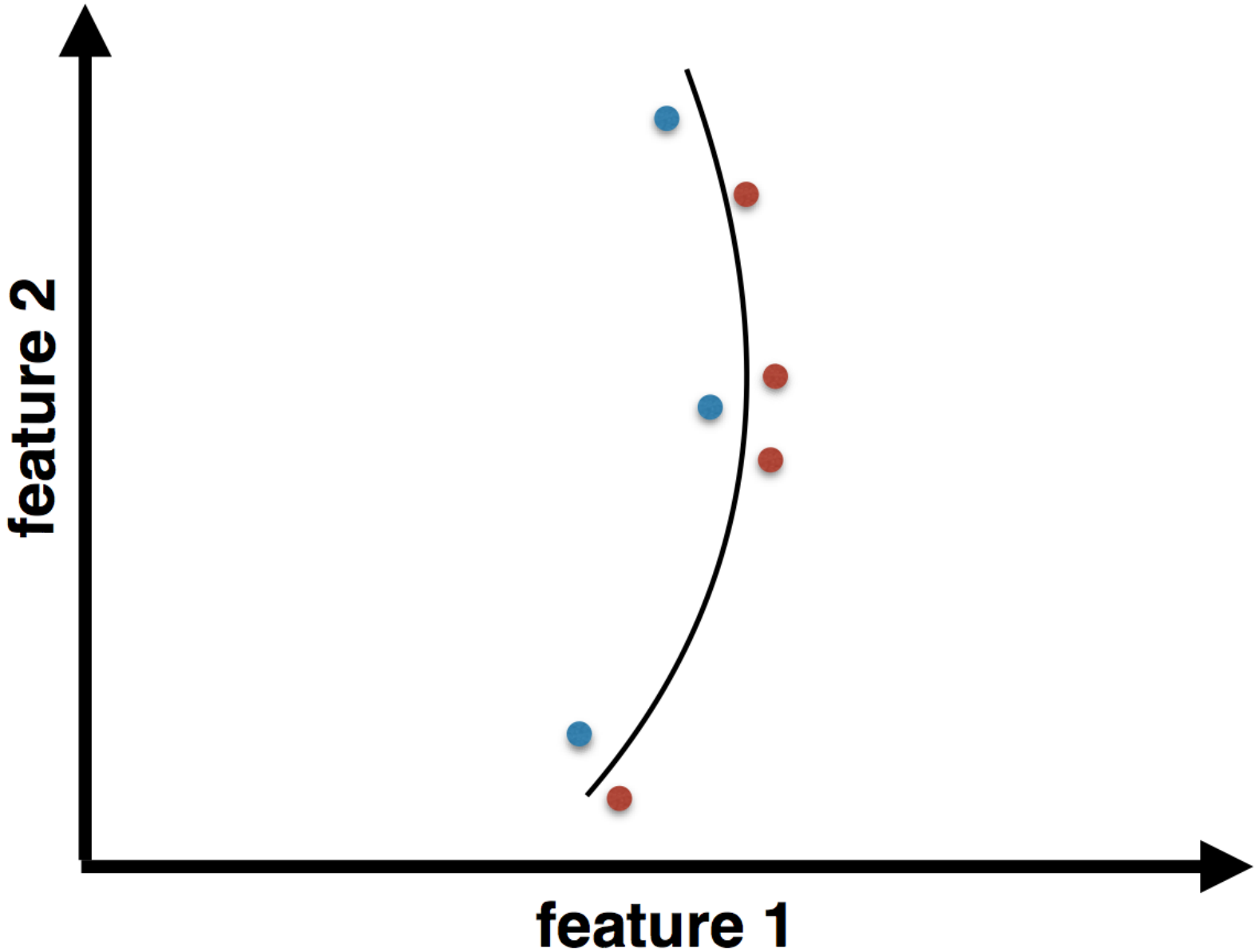
expert / oracle
analyzes experiments
to determine labels

So what's wrong with Supervised Learning

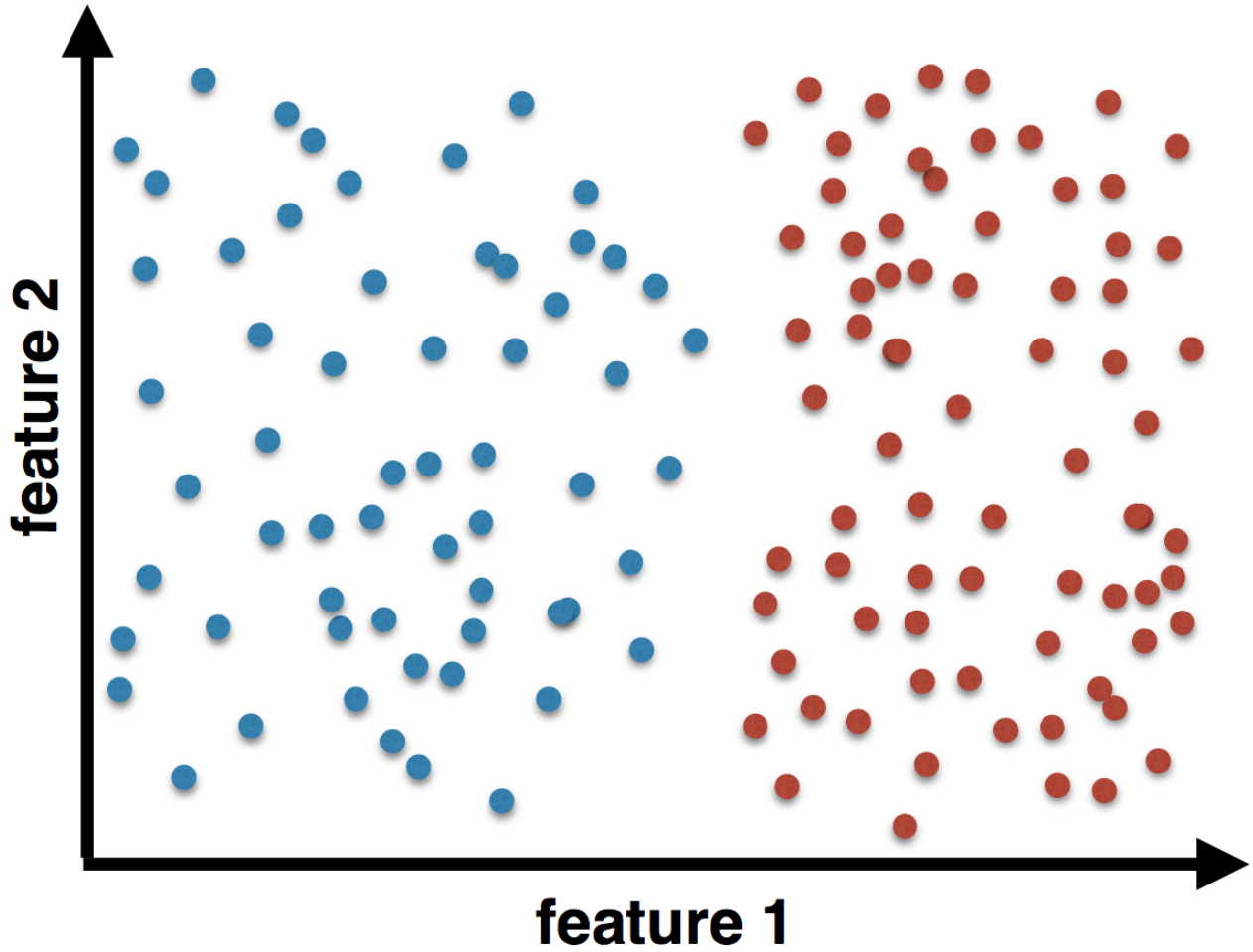


- Traditional approach almost universally adopted

Well, we actually only needed this!

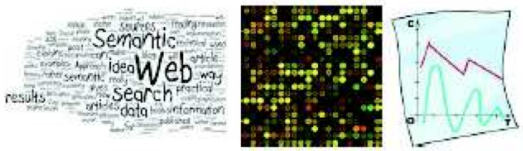


So this was a complete waste of time!



- Random sampling inevitably leads to redundancy

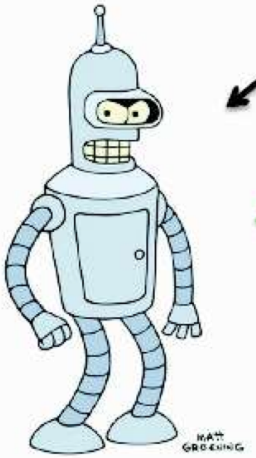
Semi-Supervised Learning



exploit the structure
in unlabeled data

raw unlabeled data
$x_1, x_2, x_3, \ldots$

random sample



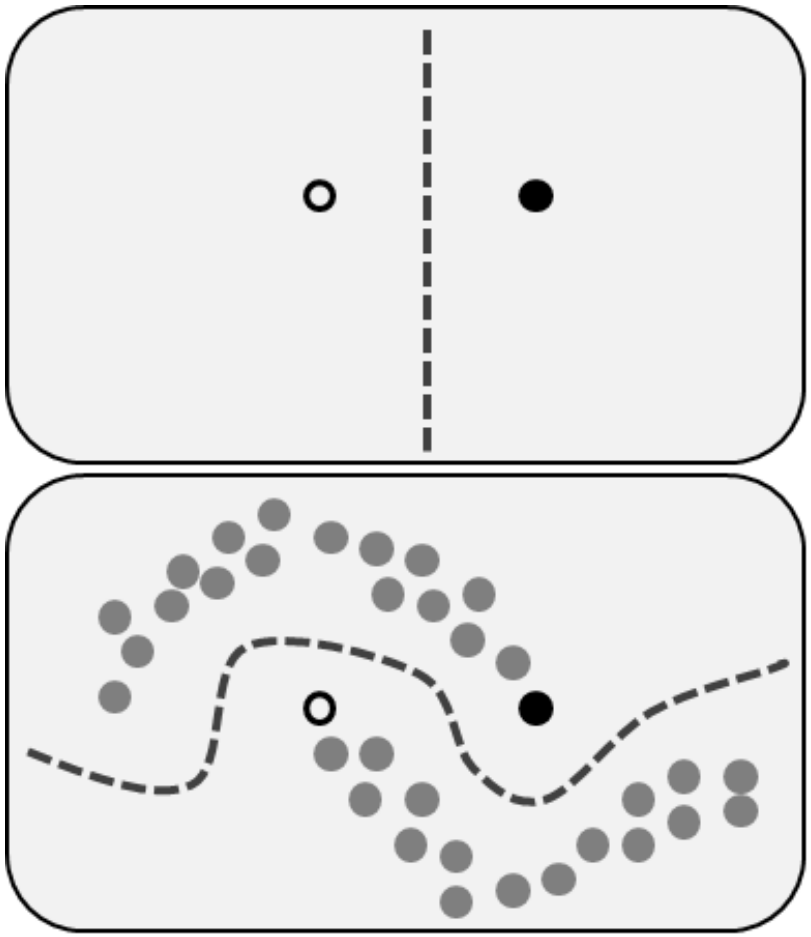
labeled training instances

$\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \langle x_3, y_3 \rangle, \ldots$

semi-supervised learner
induces a classifier

expert / oracle
analyzes experiments
to determine labels

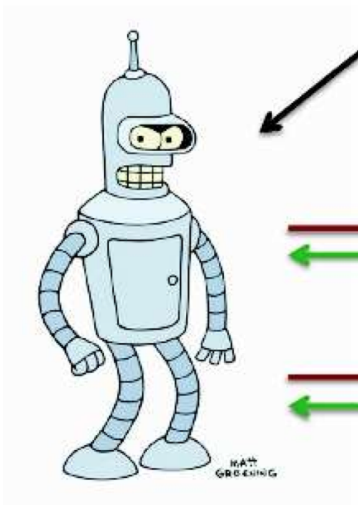
Semi-Supervised Learning



Active Learning

raw unlabeled data
$x_1, x_2, x_3, \ldots$

inspect the unlabeled data



request labels for selected data

$\langle x_1, ? \rangle$

$\langle x_1, y_1 \rangle$

$\langle x_2, ? \rangle$

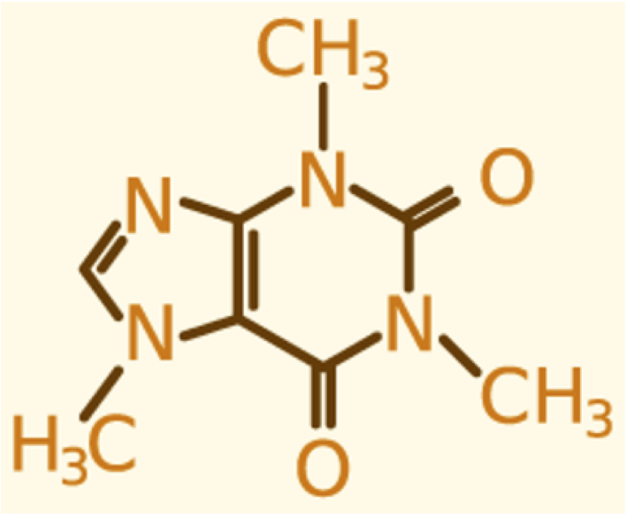
$\langle x_2, y_2 \rangle$

active learner
induces a classifier

expert / oracle
analyzes experiments
to determine labels

Active Learning example: drug design

- Goal: find compounds which bind to a particular target



Large collections 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

Labeling examples

- Example 1: Netflix Challenge
  - Concept: movies Bob would like
  - Instances: $10\,000$ movies on netflix
  - Labeling: Bob watches a movie and reports
- Example 2: Labeling phonemes
  - Concept: words labeled with phonetic alphabet
  - Instances: 1000 hours of talk radio recordings
  - Labeling: Hire linguist to annotate each syllable

TOO TIME CONSUMING/EXPENSIVE!!!

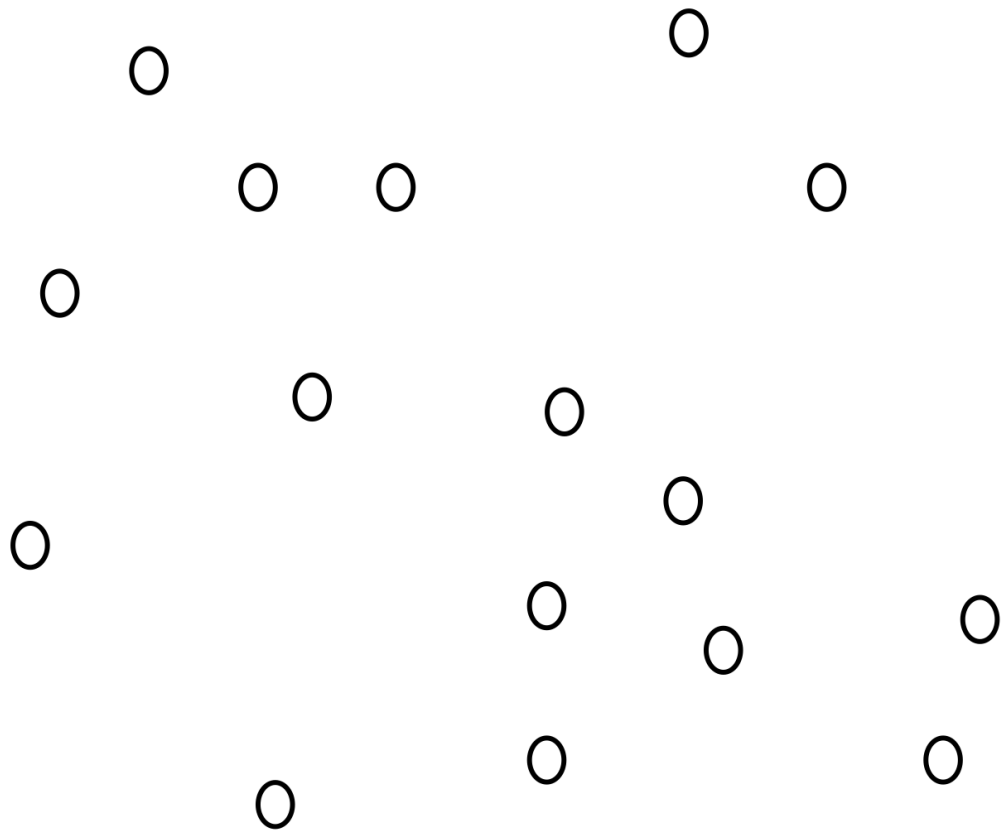
Simple Idea

- If we just pick the RIGHT examples to label, we can learn the concept from only a few labeled examples

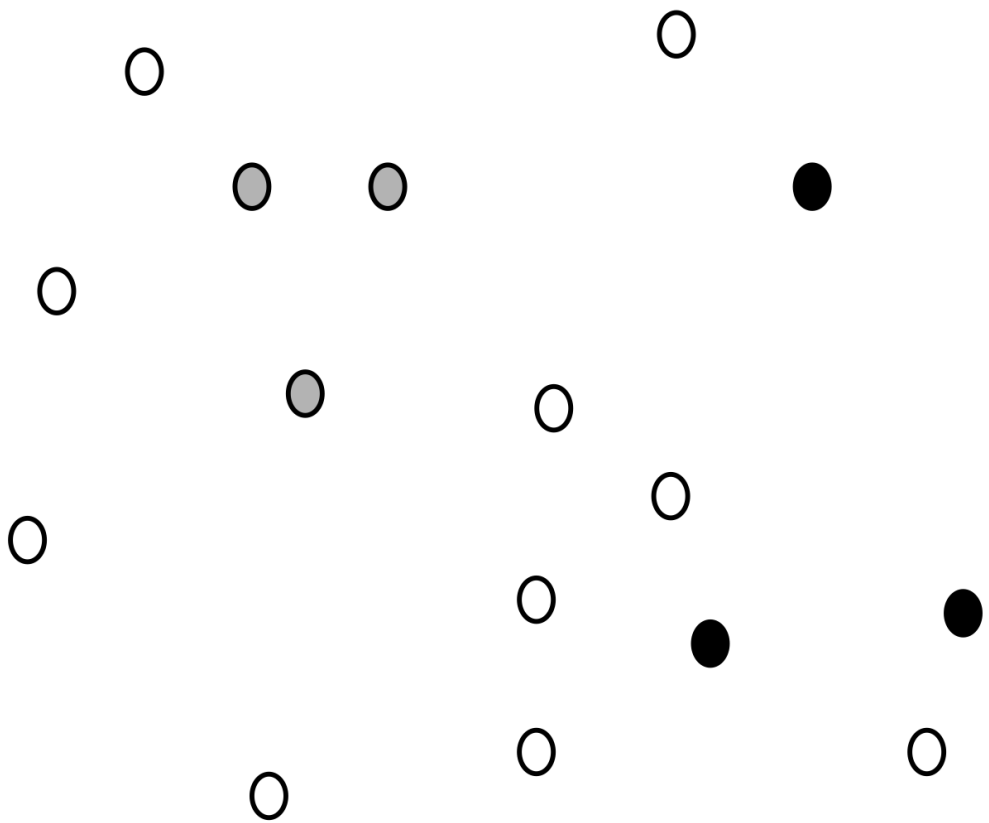
Active Learning Heuristic

- Start with a pool of unlabeled data
- Pick a few points at random and get their labels
- Repeat the following until we have budget left for getting labels
  1. Fit a classifier to the labels seen so far
  2. Pick the BEST unlabeled point to get a label for
     - (closest to the boundary?)
     - (most uncertain?)
     - (most likely to decrease overall uncertainty?)

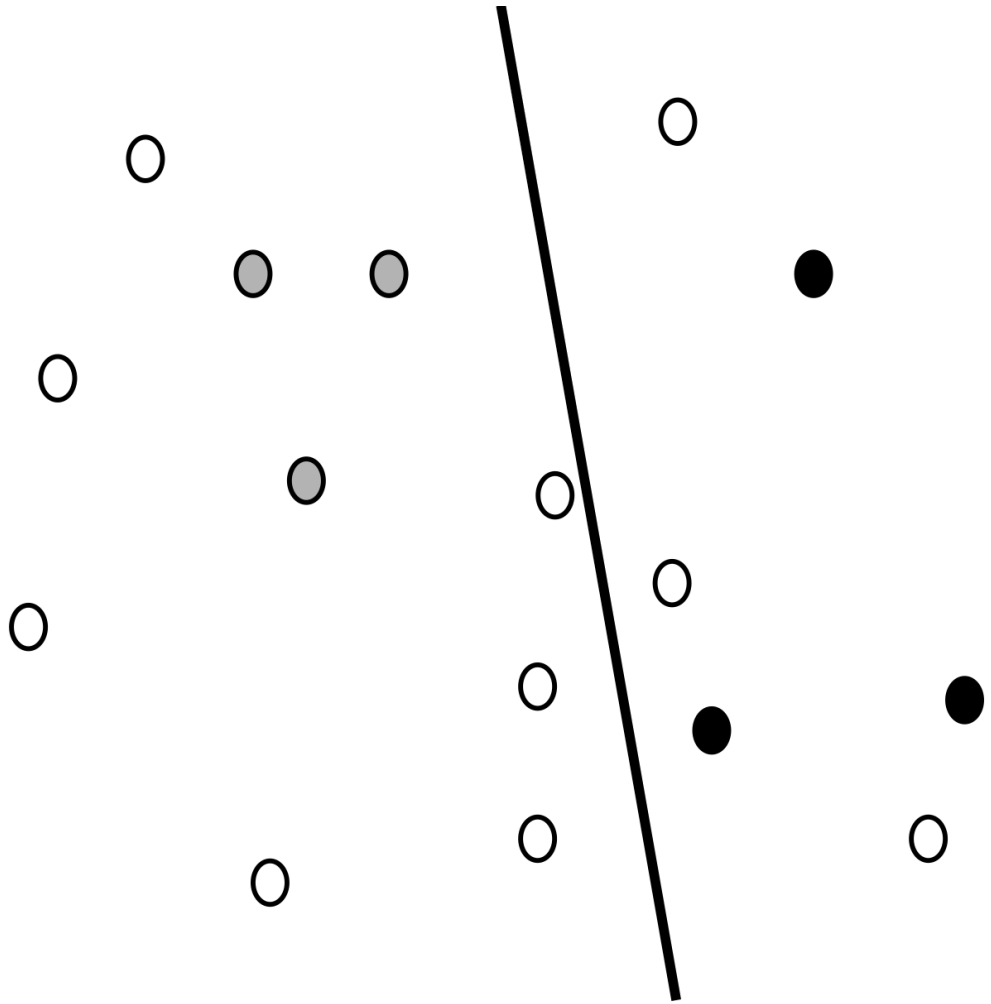
Start: Unlabeled Data



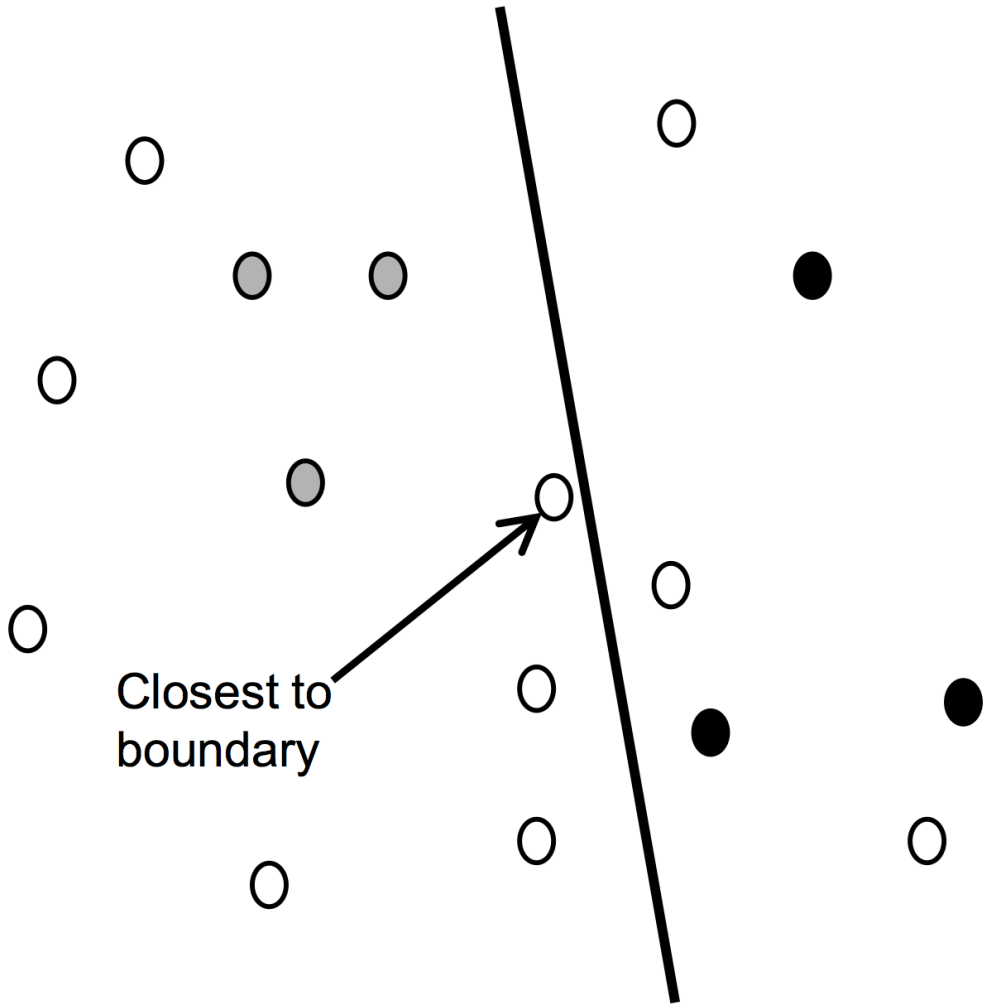
Label a Random Subset



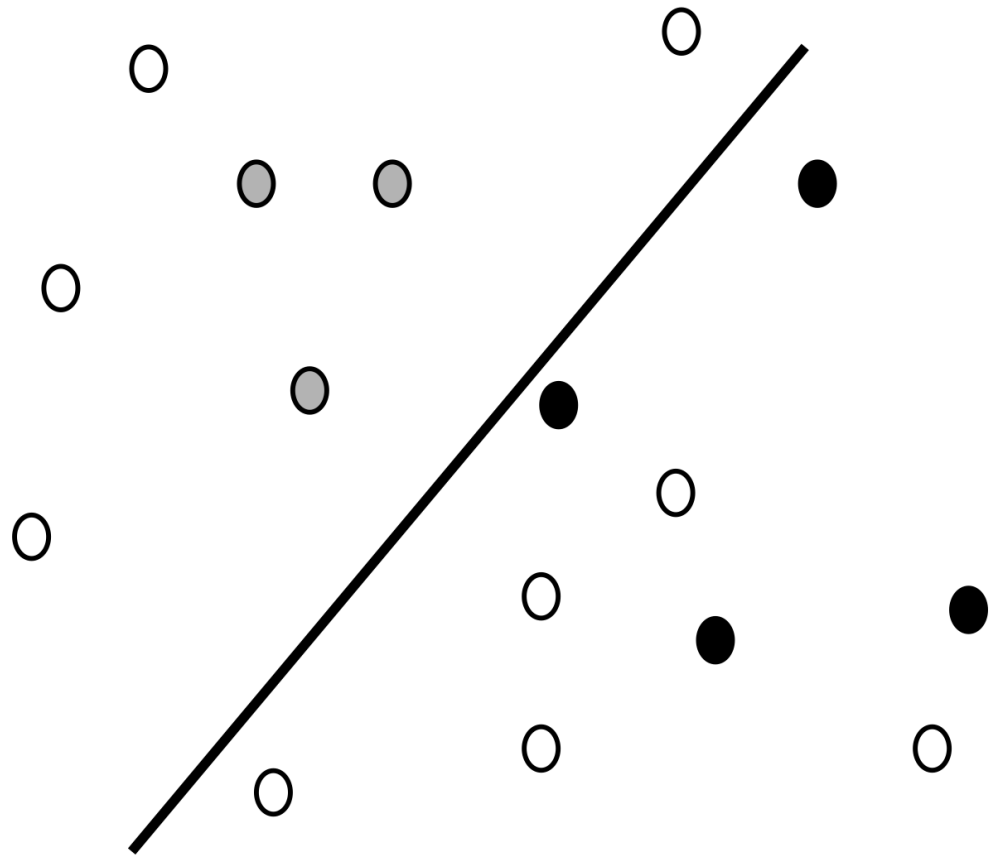
Fit a Classifier to Labeled Data



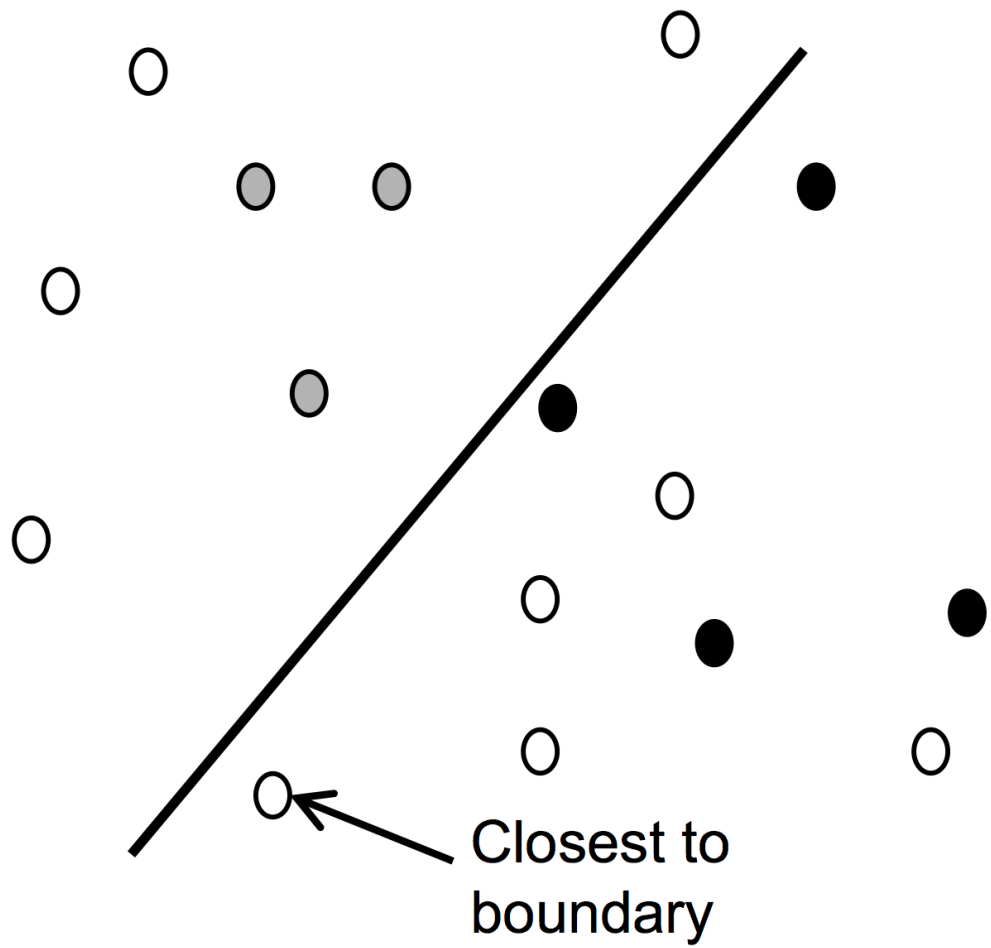
Pick the Best Next Point To Label



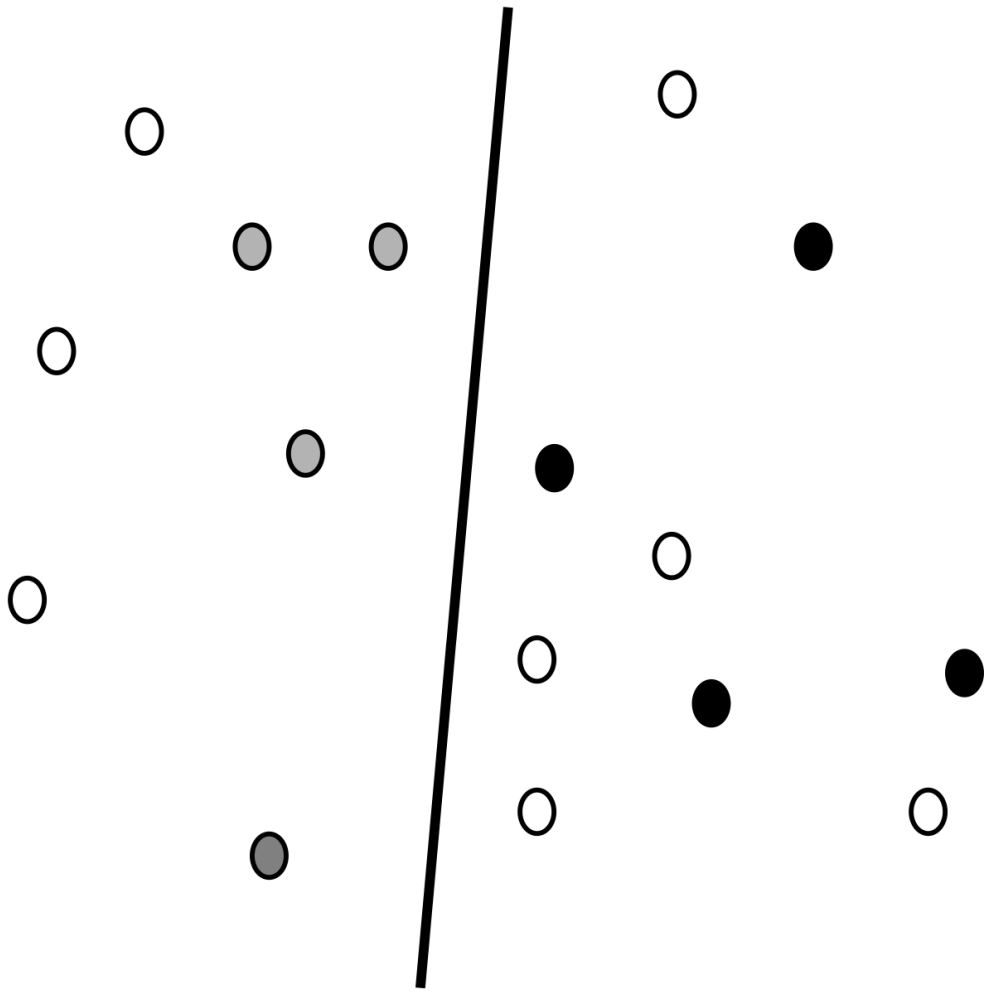
Fit a Classifier to Labeled Data



Pick the Best Next Point To Label

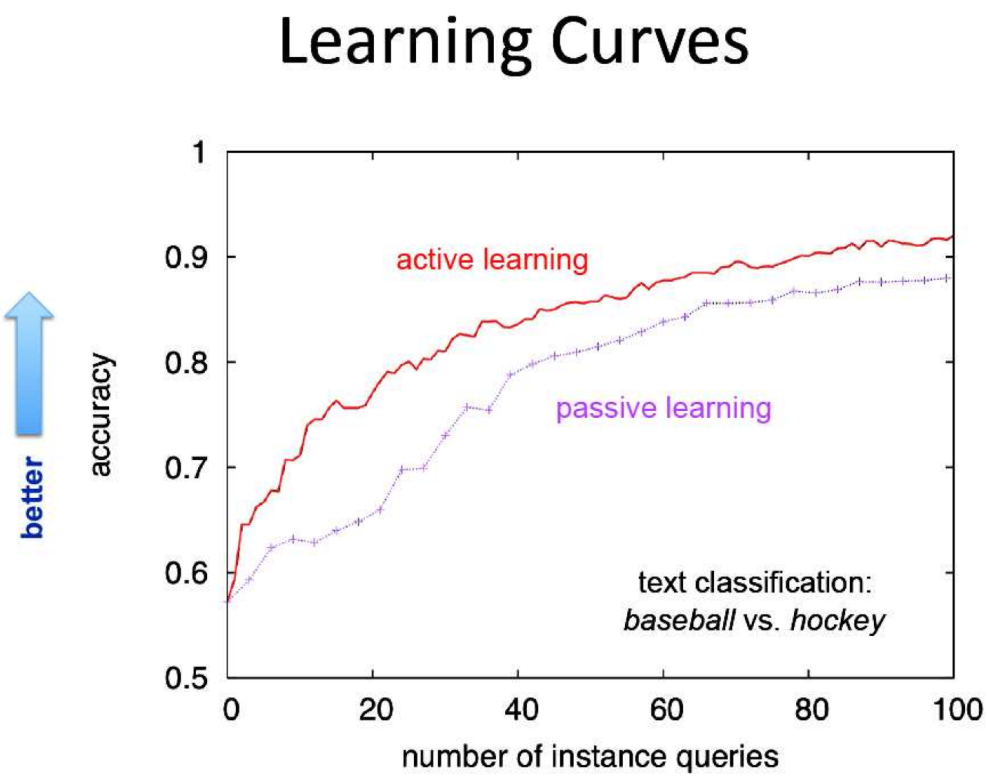


Fit a Classifier to Labeled Data



Partitional clustering

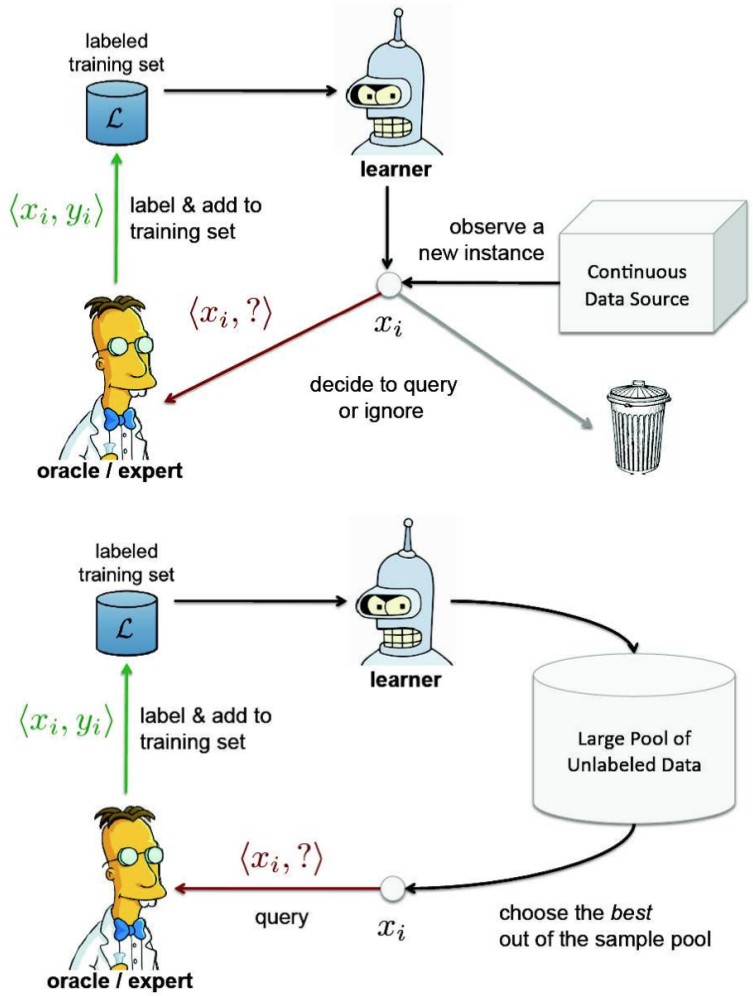
- Passive Learning curve: Randomly selects examples to get labels for
- Acctive learning curve: Active learning examples to get labels for



Types of Active Learning

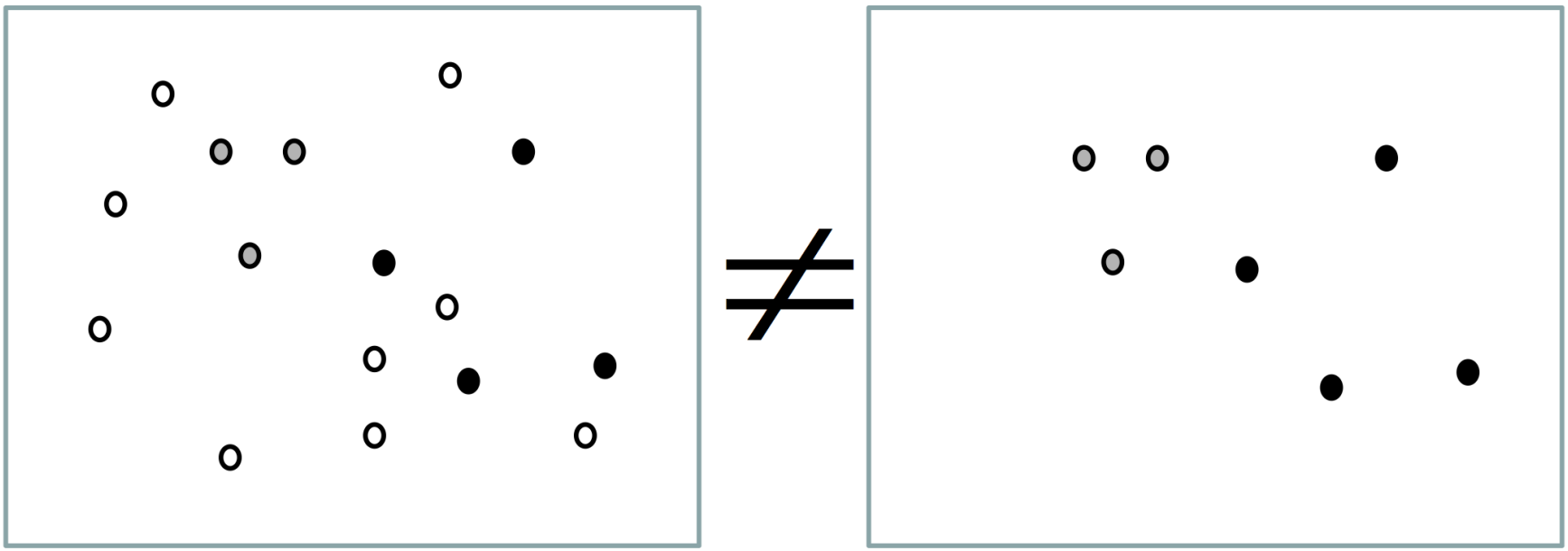
Largely falls into one of these two types:

- Stream-Based Active Learning
  - Consider one unlabeled example at a time
  - Decide whether to query its label or ignore it
- Pool-Based Active Learning
  - Given: a large unlabeled pool of examples
  - Rank examples in order of informativeness
  - Query the labels for the most informative example(s)



Biased Sampling

- The labeled points may not be representative of the underlying distribution
- This can increase error, even with infinitely many labeled samples

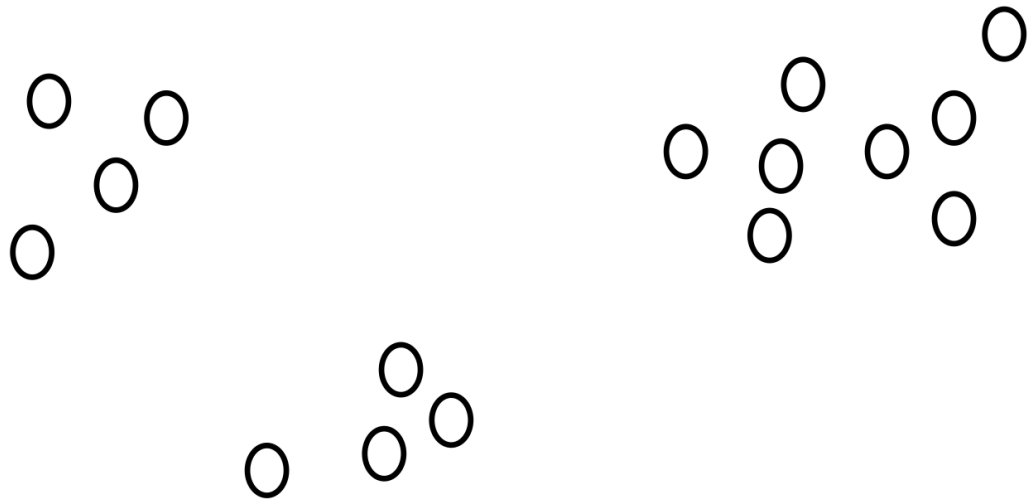


Two Rationales for Active Learning

- Rationale 1: We can exploit cluster structure in data
- Rationale 2: We can efficiently search through the hypothesis space

Exploiting structure in data

If the data looked like this ...



... then we might just need 3 labeled points

- Issues
  - Structure may not be so clearly defined
  - Structure exists at many levels of granularity
  - Clusters may not be all one label

Efficient Hypothesis Search

- If each query cuts the feature space in 2, we may need only $\log(|H|)$ to get a perfect hypothesis
- Which example should we label?
- Do there always exist queries that will cut off a good portion of the version space?
- If so, how can these queries be found?
- What happens in the nonseparable case?

Notations

- Problem: train hypothesis $h : X \to Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\ldots}$ when getting labels y_i is expensive
- Input: initial labeled sample $S_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$
- Output: hypothesis h and labeled sample $S_{m+k} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m+k}$
- General workflow:
  1. Train h using initial sample $S_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$
  2. For all $i = m+1, \ldots, m+k$
     - select an object $\mathbf{x}_i$
     - get a label y_i
     - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$
- Aim of active learning: increase accuracy of h as much as possible while using a less as possible additional labeled examples

Uncertainty sampling

- Select examples which the current model is the most uncertain about
- Various ways to measure uncertainty. For example:
  - Based on the distance from the hyperplane
  - Using the label probability $P(y|\mathbf{x})$ (for probabilistic models)
- Let us consider multi-class classification based on some probabilistic model $P(y|\mathbf{x})$
- Decision according to the model is equal to

$$h(\mathbf{x}) = \arg\max_{y \in Y} P(y|\mathbf{x})$$

- Let us denote by $p_r(\mathbf{x})$, $r = 1, 2, \ldots, |Y|$ values of $P(y|\mathbf{x})$, $y \in Y$, ranked in decreasing order

Uncertainty sampling

- Least confidence principle

$$\mathbf{x}_i = \arg\min_{\mathbf{x}\in X} p_1(\mathbf{x})$$

- Margin sampling principle

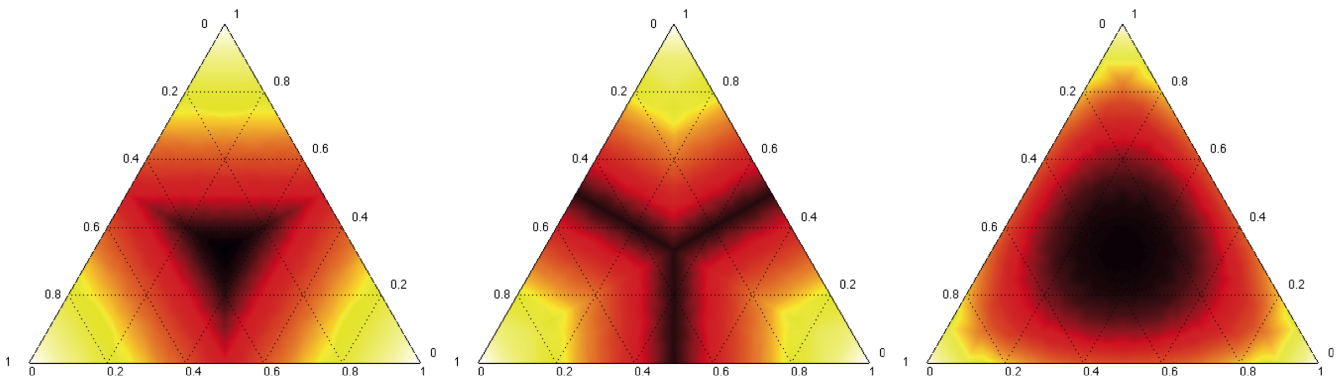
$$\mathbf{x}_i = \arg\min_{\mathbf{x}\in X} (p_1(\mathbf{x}) - p_2(\mathbf{x}))$$

- Maximum entropy principle

$$\mathbf{x}_i = \arg\min_{\mathbf{x}\in X} \sum_r p_r(\mathbf{x}) \log p_r(\mathbf{x})$$

Uncertainty sampling

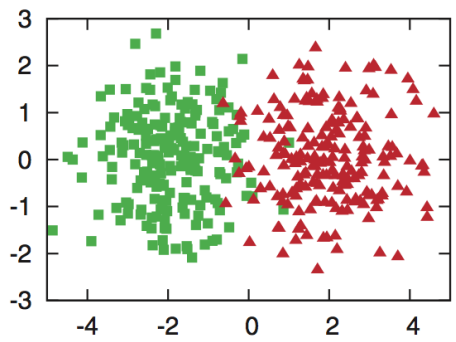
- In case of two classes these three principles are equivalent
- In a multi-class setting there are differences
- Below we show contour lines the corresponding criteria



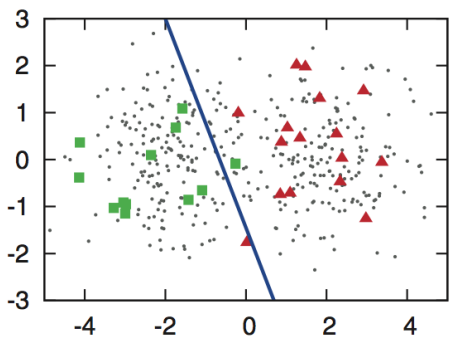
least confident margin entropy

Active vs. Passive learning

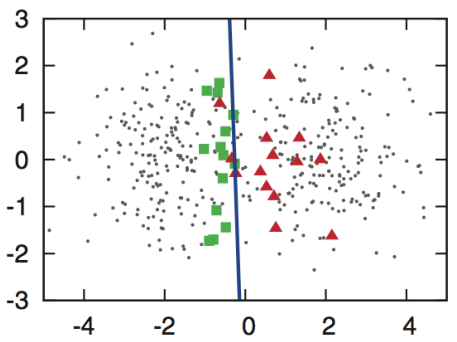
- Example 1. Synthetic dataset: $m = 30$, $m + k = 400$
  - two Gaussian density
  - logistic regression is constructed using 30 randomly selected objects
  - logistic regression is constructed using 30, adaptively selected using active learning



(a) a 2D toy data set



(b) random sampling



(c) uncertainty sampling

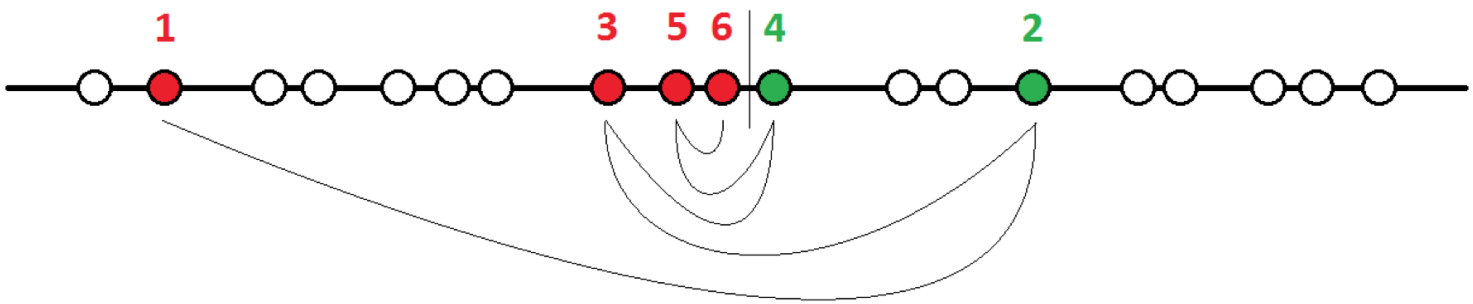
For learning a hypothesis having the same accuracy using biased non-random sample we need smaller number of data points

Active vs. Passive learning

- Example 2. 1-d problem with a threshold classifier

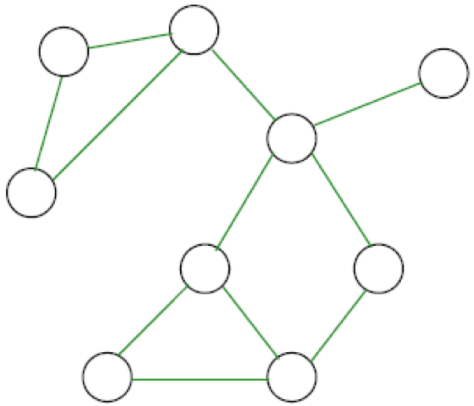
$$x_i \sim \text{Uniform}[-1, 1], \;\; y_i = 1_{x_i > 0}, \;\; h_\theta(x) = 1_{x > \theta}$$

  Let us calculate a number of steps we need to identify θ with average accuracy $\frac{1}{k}$
- Naive approach: select $x_i \sim \text{Uniform}(X) \to$ we need $O(k)$ steps
- Binary search select x_i as close as possible to the center of the margin between classes $\frac{1}{2}\left(\max_{y_j=0}(x_j) + \min_{y_j=1}(x_j)\right) \to$ we need $O(\log k)$ steps

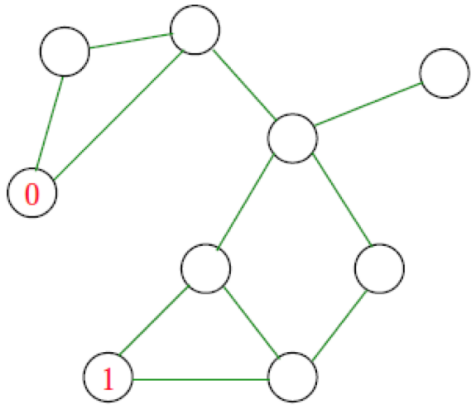


Uncertainty Sampling based on Label-Propagation

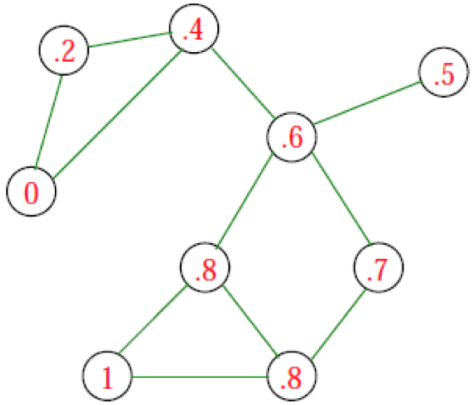
(1) Build neighborhood graph



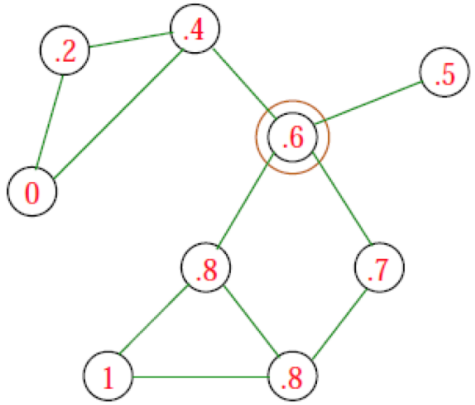
(2) Query some random points



(3) Propagate labels



(4) Make query and go to (3)



Query by committee

- Select $\mathbf{x}_i$ having the highest disagreement between decisions of a committee of models $h_t(\mathbf{x}_i) = \arg\max_{y \in Y} P_t(y|\mathbf{x})$, $t = 1, \ldots, T$
- Maximum Entropy Principle: select those $\mathbf{x}_i$ for which $h_t(\mathbf{x}_i)$ are the most different

$$\mathbf{x}_i = \arg\min_{\mathbf{z} \in X} \sum_{y \in Y} \hat{p}(y|\mathbf{z}) \log \hat{p}(y|\mathbf{z})$$

where $\hat{p}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^{T} 1_{h_t(\mathbf{z})=y}$

- Maximum of an average KL-divergence principle: select those $\mathbf{x}_i$ on which $P_t(y|\mathbf{x}_i)$ are the most different

$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in X} \sum_{t=1}^{T} \mathrm{KL}\left(P_t(y|\mathbf{z}) \parallel \overline{P}(y|\mathbf{z})\right),$$

where $\overline{P}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^{T} P_t(y|\mathbf{z})$

Expected Model Change

- Select those $\mathbf{x}_i$, which would provide the highest change of a model
- We use a parametric classification model

$$h_\theta(\mathbf{x}) = \arg\max_{y \in Y} P(y|\mathbf{x}, \theta)$$

- For $\mathbf{z} \in X$ and $y \in Y$ estimate a step of stochastic gradient descent when performing re-learning of the model with additional data point $(\mathbf{z}, y)$
- Let us denote by $\nabla_\theta \hat{R}(\theta; \mathbf{z}, y)$ is a gradient vector the loss function
- We calculate

$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in X} \sum_{y \in Y} P(y|\mathbf{z}, \theta) \left\| \nabla_\theta \hat{R}(\theta; \mathbf{z}, y) \right\|$$

Expected Error Reduction

- Select those $\mathbf{x}_i$, which after re-learning provides the most reliable classification of the non-labeled feature vectors from X
- For $\mathbf{z} \in X$ and $y \in Y$ we construct a classifier, adding to S_m additional example $(\mathbf{z}, y)$

$$h_{\mathbf{z}y}(\mathbf{x}) = \arg\max_{u \in Y} P_{\mathbf{z}y}(u|\mathbf{x})$$

- Maximum Reliability of non-labeled data principle

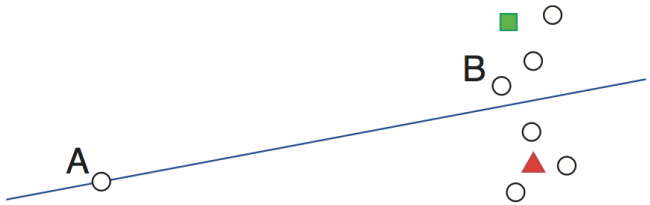
$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in X} \sum_{y \in Y} P(y|\mathbf{z}) \sum_{j=m+1}^{m+k} P_{\mathbf{z}y}(h_{\mathbf{z}y}(\mathbf{x}_j)|\mathbf{x}_j)$$

- Minimum Entropy of non-labeled data principle

$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in X} \sum_{y \in Y} P(y|\mathbf{z}) \sum_{j=m+1}^{m+k} \sum_{u \in Y} P_{\mathbf{z}y}(u|\mathbf{x}_j) \log P_{\mathbf{z}y}(u|\mathbf{x}_j)$$

Density-weighted methods

- Reduce weight of nonrepresentational objects
- Example: object A is more boundary, but it is less representative than B



- Any criterion for sampling has the form

$$\mathbf{x}_i = \arg\max_{\mathbf{x}} \phi(\mathbf{x})$$

and can be adjusted by a local density estimate

$$\mathbf{x}_i = \arg\max_{\mathbf{x}} \phi(\mathbf{x}) \left(\sum_{j=m+1}^{m+k} \mathrm{sim}(\mathbf{x}, \mathbf{x}_j) \right)^{\beta},$$

where $\mathrm{sim}(\cdot, \cdot)$ is a some similarity function (the closer the bigger)

Example of Density Weighting