Active Learning. Adaptive Design of Experiments

Evgeny Burnaev

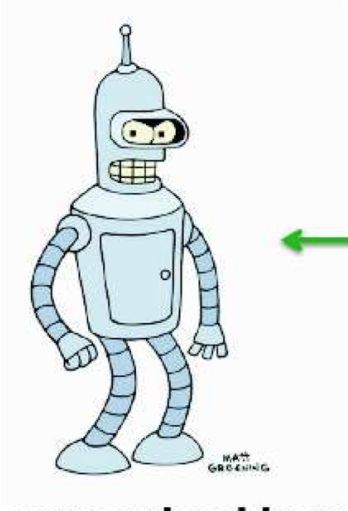
Skoltech, Moscow, Russia

Outline

(Passive) Supervised Learning

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

random sample



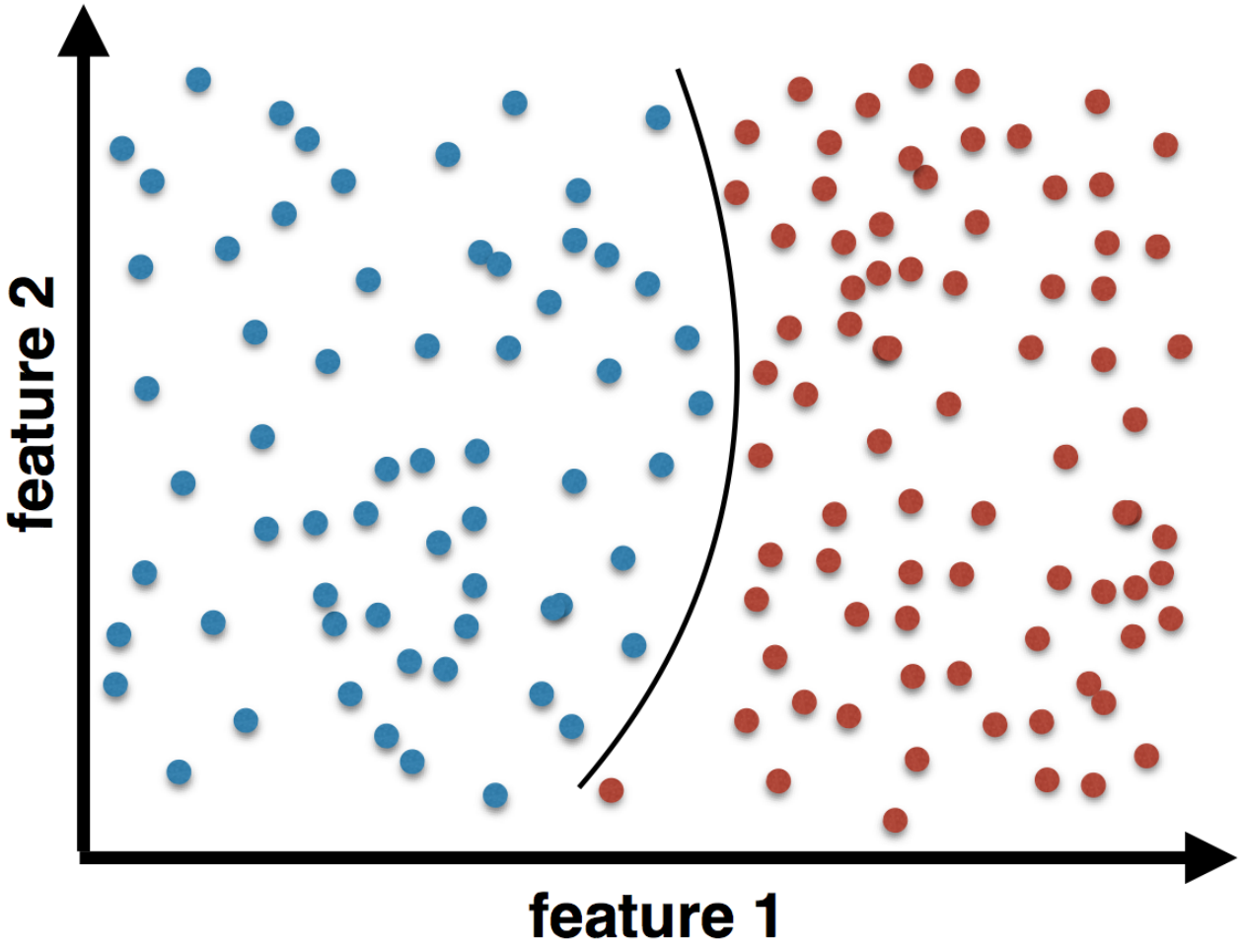
labeled training instances

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

supervised learner
induces a classifier

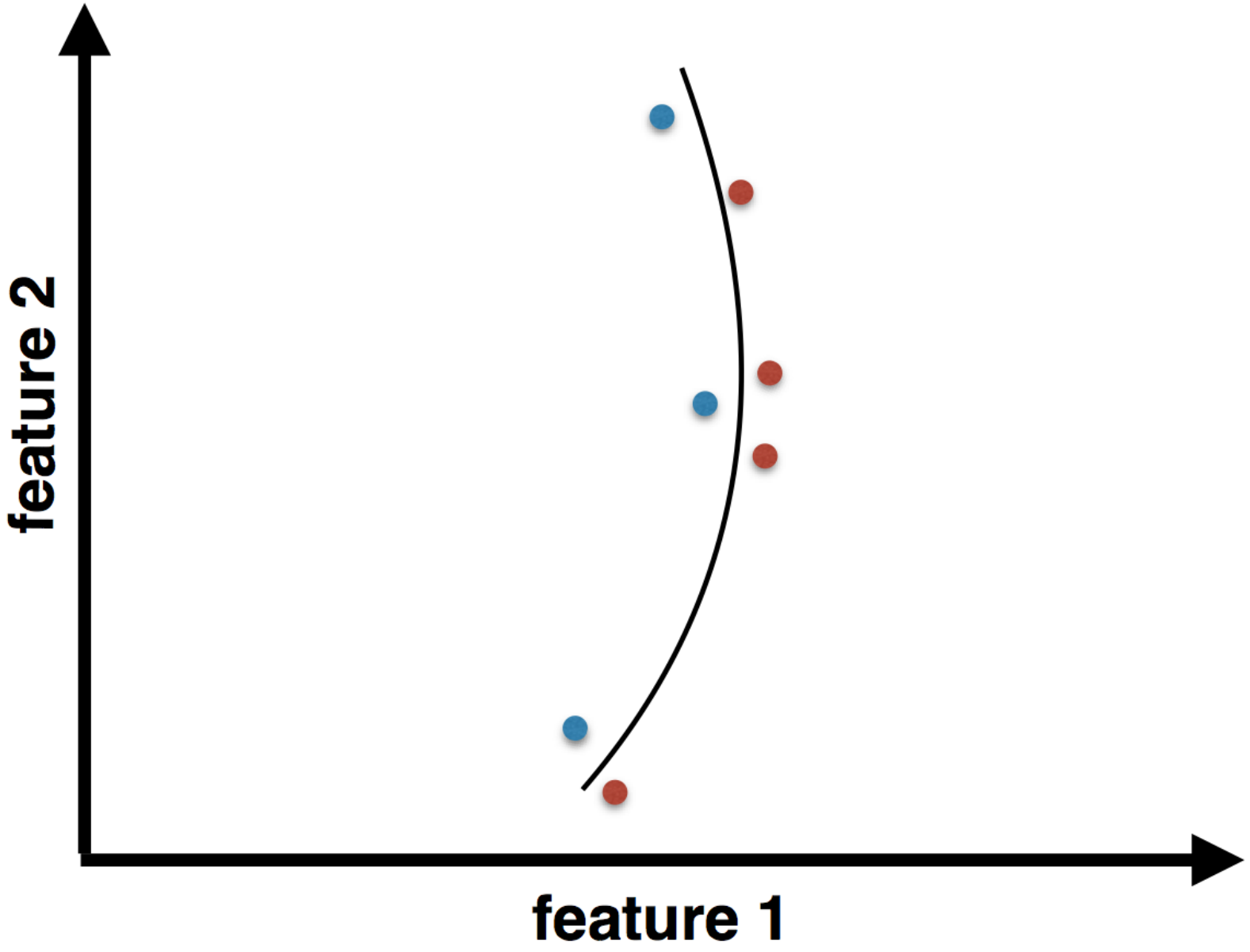
expert / oracle
analyzes experiments
to determine labels

So what's wrong with Supervised Learning

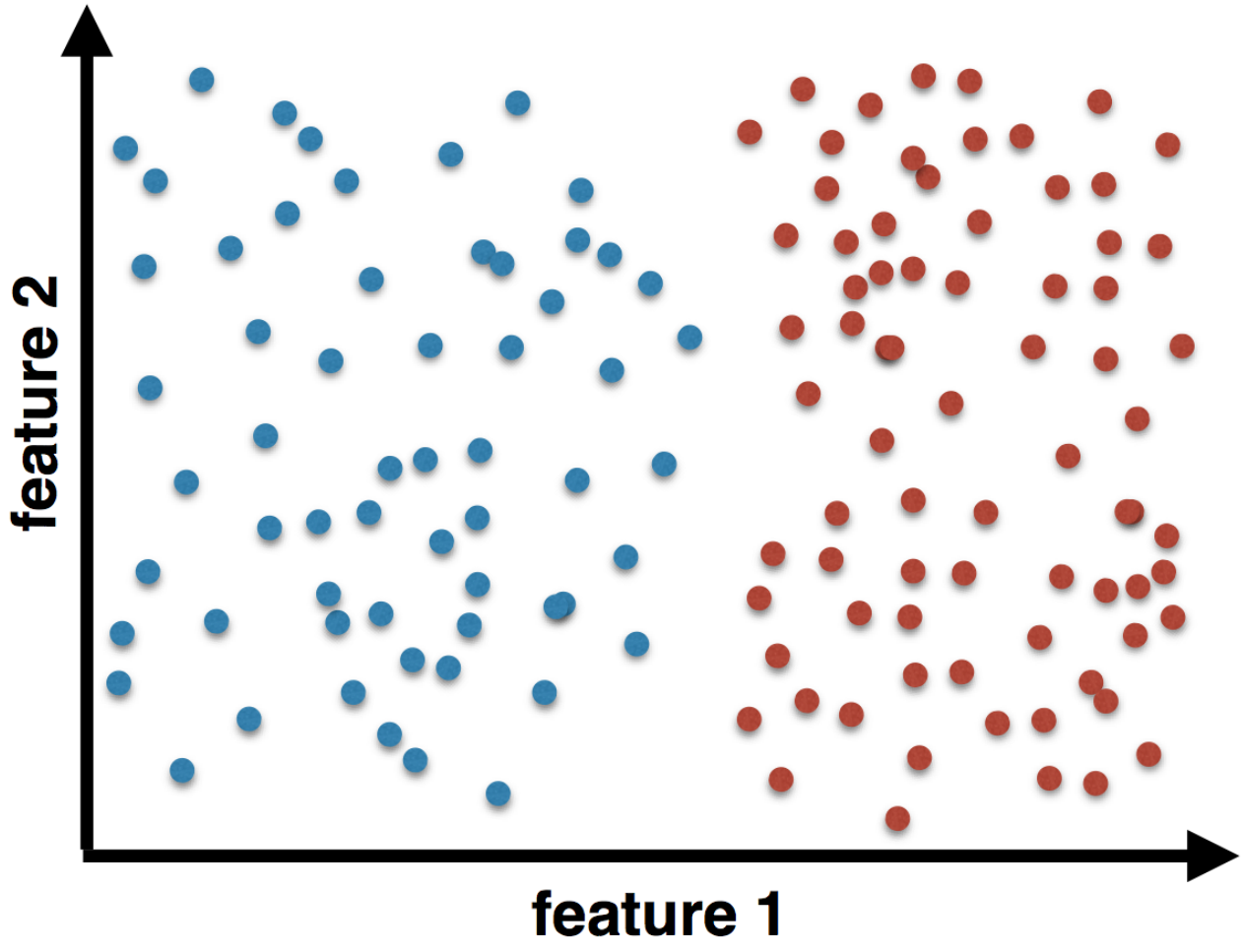


- Traditional approach almost universally adopted

Well, we actually only needed this!

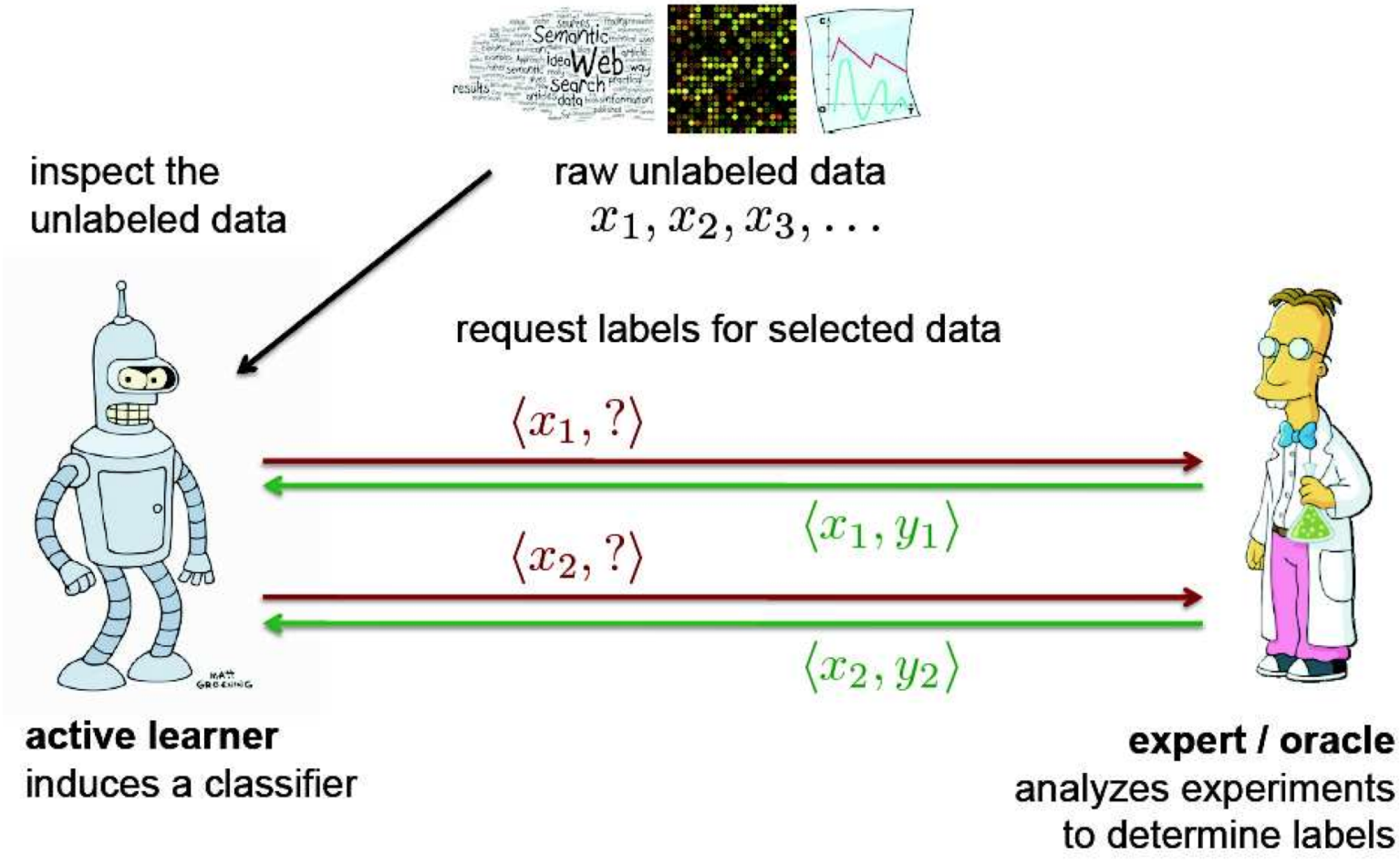


So this was a complete waste of time!



- Random sampling inevitably leads to redundancy

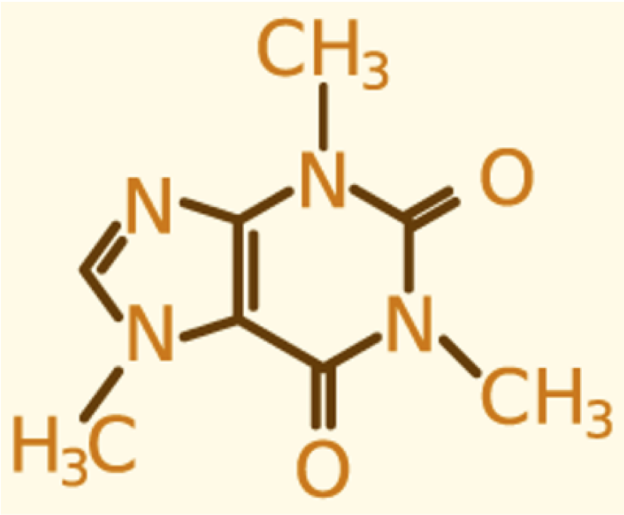
Active Learning



Aim of Active Learning: increase accuracy as much as possible while using a less as possible additional labeled examples

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

Definition

Adaptive DoE is an approach that allows to iteratively add points to the training set minimizing the error of the model.

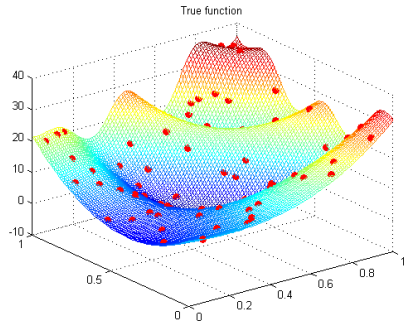


Figure – Unknown function

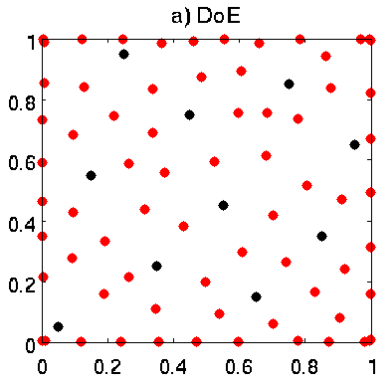


Figure – Initial and added points

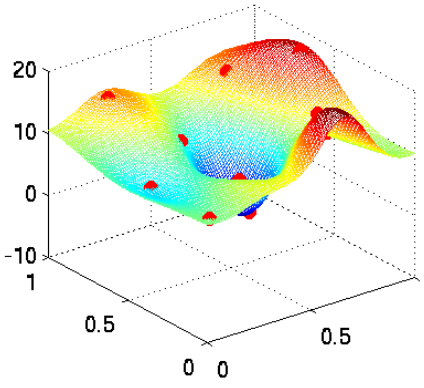


Figure – Initial model

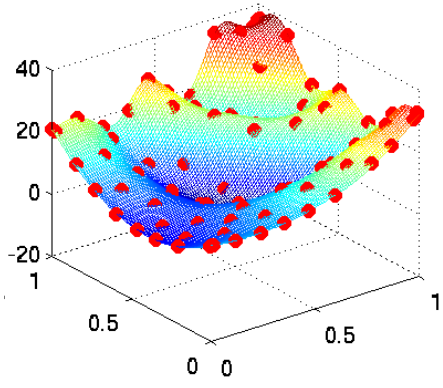


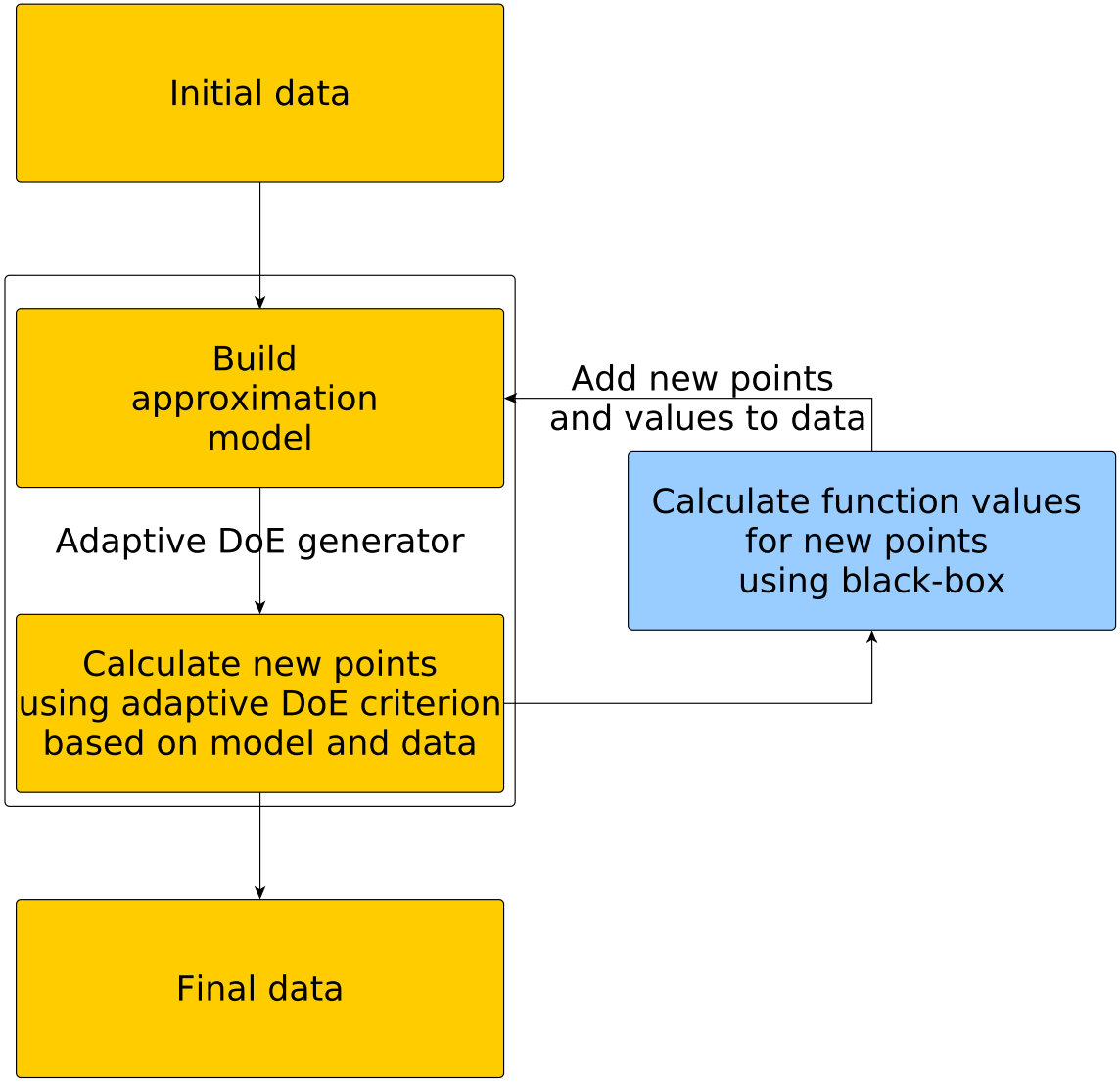
Figure – Final model

Adaptive DoE allows to control the process of modelling by adaptive sampling that improves the quality of the model

Adaptive DoE:

- Generate initial sample using space-filling technique
- Build a model using the obtained training set
- Update the training set by iteratively adding points which improves the model the most

Adaptive DoE



Reasons for such approach:

1. Approximation contains information about the dependency

$$\Downarrow$$

Approximation can give clue on how to select new point in order to increase model quality

2. Adaptive update of training set gives flexibility in sample size

$$\Downarrow$$

If the desired quality is reached generation of DoE can be stopped

Sampling criterion

The procedure of choosing new point can be formulated as maximization of some criterion:

$$\mathbf{x}_{m+1} = \arg\max_{\mathbf{x}\in\mathbb{X}} \mathcal{I}(\mathbf{x}|S_m, \widehat{f}_m, \widehat{\sigma}^2_m),$$

where $\mathcal{I}(\mathbf{x}|S, \widehat{f}, \widehat{\sigma}^2)$ is a criterion of point selection based on

- current training set S,
- current approximation model $\widehat{f}$ and
- current error of approximation $\widehat{\sigma}^2$

Maximum variance criterion

- Maximum variance criterion:

$$\mathcal{I}_{MV}(\mathbf{x}) = \widehat{\sigma}^2(\mathbf{x}|S),$$

  where $\widehat{\sigma}^2(\mathbf{x}|S)$ is an error at point $\mathbf{x}$ of the model trained on $S = (\mathbf{X}, \mathbf{y})$
- As $\widehat{\sigma}^2(\mathbf{x}|S)$ we can use GP-based posterior variance $\sigma_*^2(\mathbf{x})$
- **NB**!: for GP

$$\sigma_*^2(\mathbf{x}) = \sigma_*^2(\mathbf{x}|\mathbf{X}),$$

  i.e. it depends on S explicitly only through $\mathbf{X}$

Advantages

- Easy to calculate
- Takes into account information about current approximation

Disadvantages

- Takes into account only local behavior
- Tends to sample points that are close to the boundary $\mathbb{X}$

Minimization of mean prediction error

- GP-based criterion
- Minimum mean squared error on next iteration:

$$\mathcal{I}_{\rho_2}(\mathbf{x}) = \frac{1}{|\mathbb{X}|}\int_{\mathbb{X}} \big(\sigma_*^2(\mathbf{v}|\mathbf{X}) - \sigma_*^2(\mathbf{v}|\mathbf{X}\cup\mathbf{x})\big)\mathrm{d}\mathbf{v},$$

  where
  - $\sigma_*^2(\mathbf{v}|\mathbf{X})$ is a GP posterior variance at point $\mathbf{v}$ of the model built using $S = (\mathbf{X}, \mathbf{y})$
  - $\sigma_*^2(\mathbf{v}|\mathbf{X}\cup\mathbf{x})$ is a GP posterior variance for the input sample $\mathbf{X}^{ext} = \mathbf{X}\cup\mathbf{x}$

- GP-based criterion
- Integrated MSE Gain-Maximum Variance:

$$\mathcal{I}_{IGMV}(\mathbf{x}) = \mathcal{I}_{\rho_2}(\mathbf{x}) \cdot \mathcal{I}_{MV}(\mathbf{x})$$

Advantages

- Takes into account information about model behavior in all the design space

Disadvantages

- Relatively hard to compute

The criterion can be rewritten in the following form:

$$\mathcal{I}_{\rho_2}(\mathbf{x}) = \frac{1}{|\mathbb{X}|}\int_{\mathbb{X}} \frac{k_*^2(\mathbf{x},\mathbf{v})}{\sigma_*^2(\mathbf{x}|\mathbf{X})}\mathrm{d}\mathbf{v},$$

where

$$k_*(\mathbf{x},\mathbf{v}) = K(\mathbf{x},\mathbf{v}) - \mathbf{k}(\mathbf{x})^\top[\mathbf{K}+\sigma^2\mathrm{I}]^{-1}\mathbf{k}(\mathbf{v})$$

is a posterior covariance of GP between values $f(\mathbf{x})$ and $f(\mathbf{v})$

Criterion can be unstable if $\sigma_*^2(\mathbf{x}|\mathbf{X})$ is small

$$\Downarrow$$

It leads to noisy values and multiple non-robust minima

We can solve the problem by combining two criteria: $\mathcal{I}_{\rho_2}(\mathbf{x})$ and $\mathcal{I}_{MV}(\mathbf{x})$

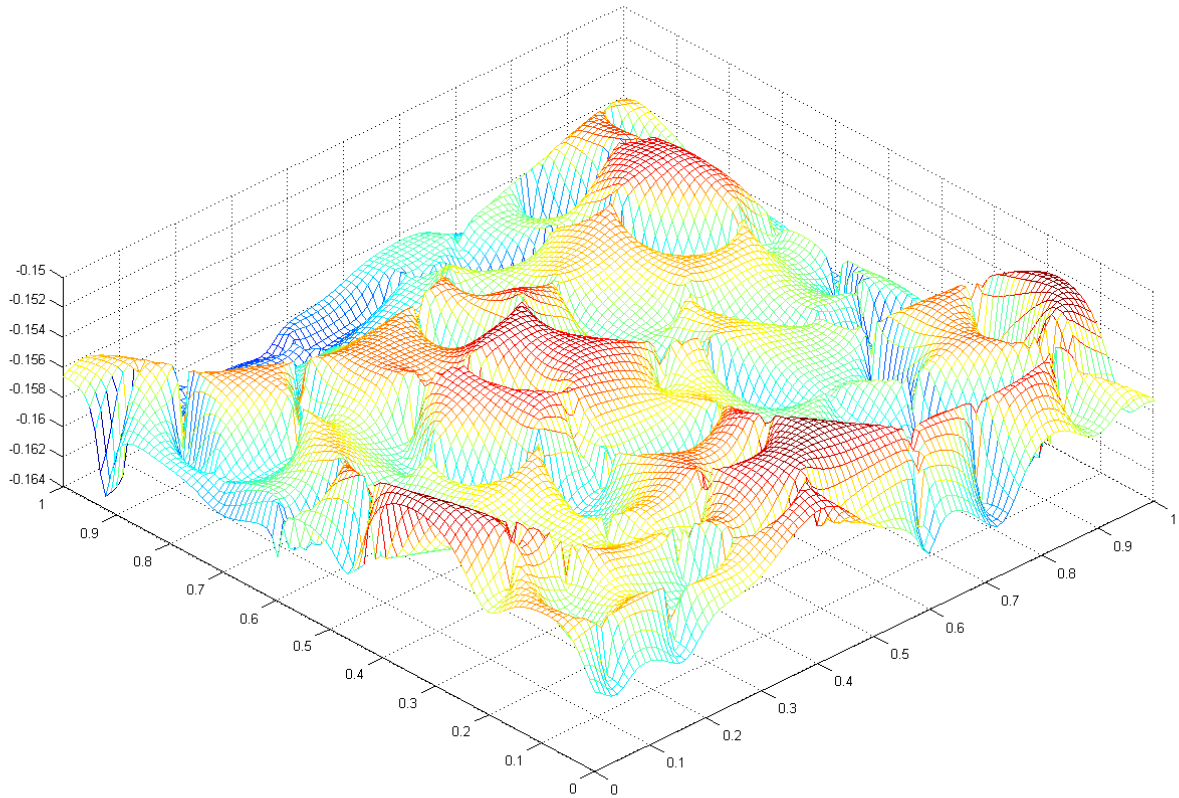
$$\mathcal{I}_{IGMV}(\mathbf{x}) = \mathcal{I}_{\rho_2}(\mathbf{x}) \cdot \mathcal{I}_{MV}(\mathbf{x}) = \frac{1}{|\mathbb{X}|} \int_{\mathbb{X}} k_*^2(\mathbf{x}, \mathbf{v}) \mathrm{d}\mathbf{v}$$

Advantages

- No issues with denominator
- Still takes into account all the design space

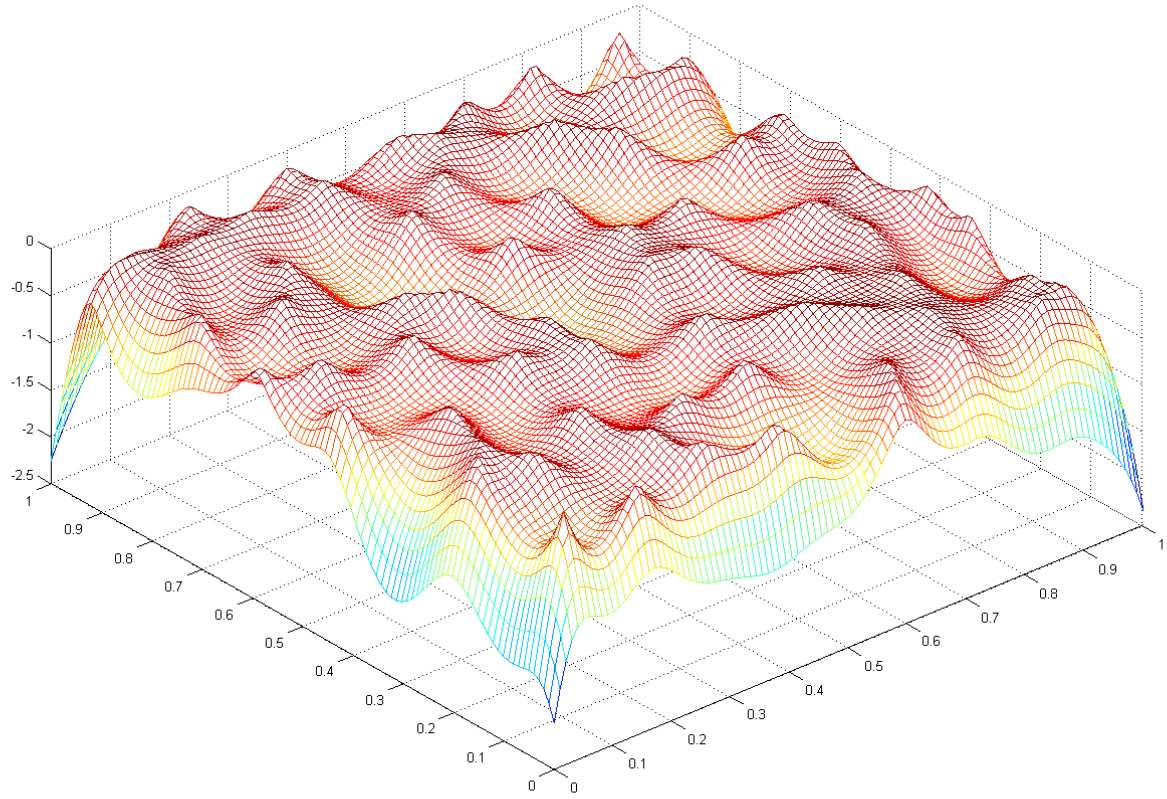
$\mathcal{I}_{\rho 2}$ surface

- High modality
- Narrow and non-robust local minima



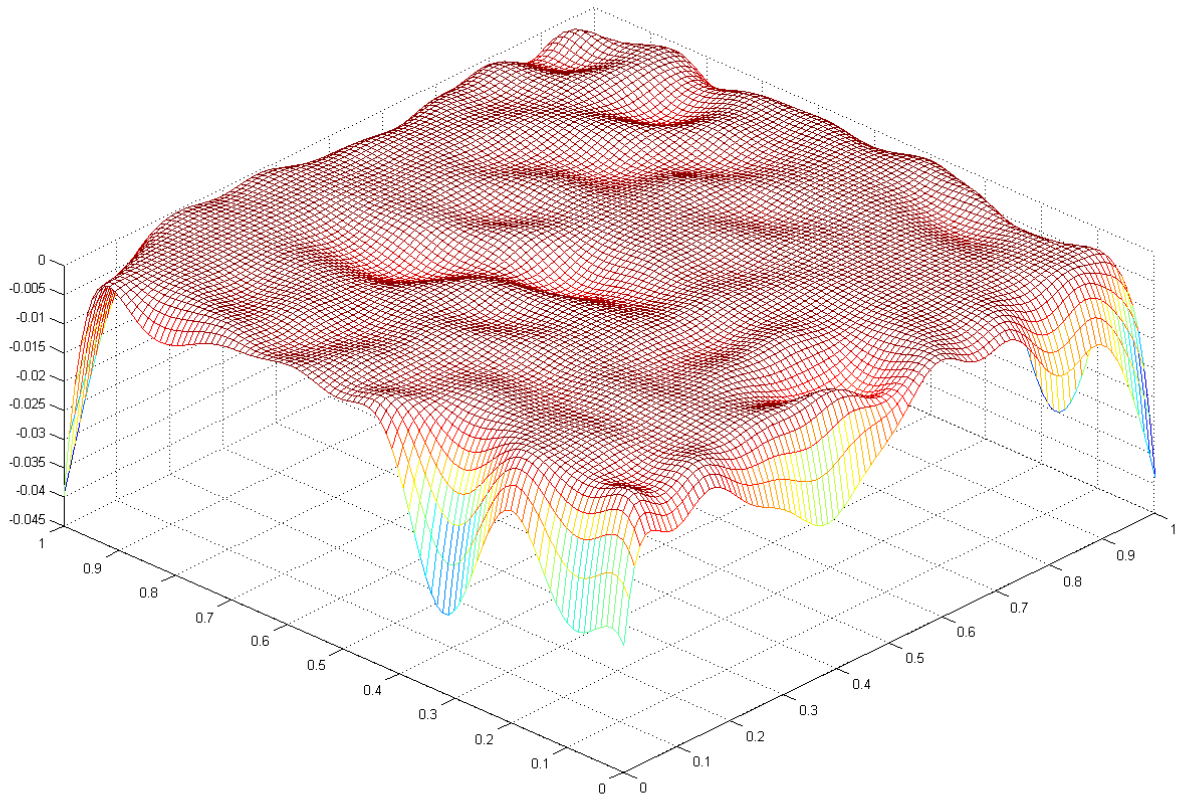
Surface of MaxVar criterion

- High modality
- More regular behavior



Surface of ImseGain-MaxVar

- Less local maxima
- Regular behavior



MaxMin criterion

$$\mathcal{I}_{MM}(\mathbf{x}) = \min_{\mathbf{v} \in \mathbb{X}} d^2(\mathbf{v}, \mathbf{x}),$$

where $d(\mathbf{v}, \mathbf{x})$ is a Euclidean distance between $\mathbf{v}$ и $\mathbf{x}$

Advantages

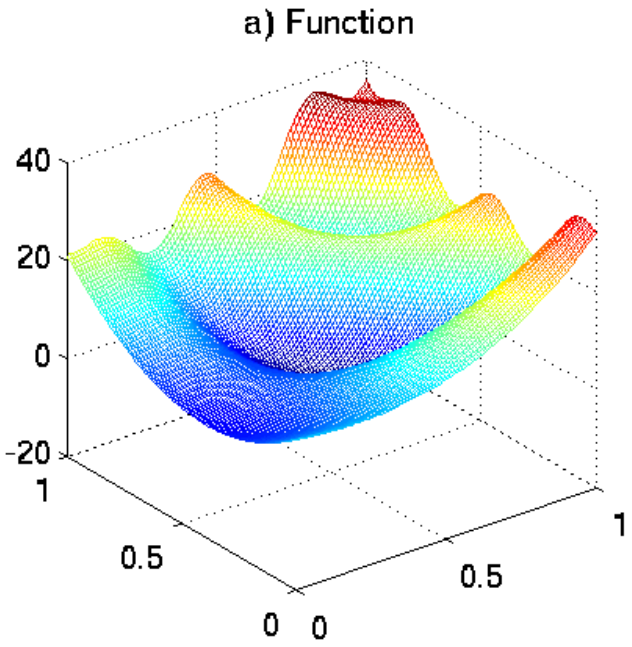
- Very fast computation
- Doesn't depend on the modely type
- Samples point uniformly in the design space

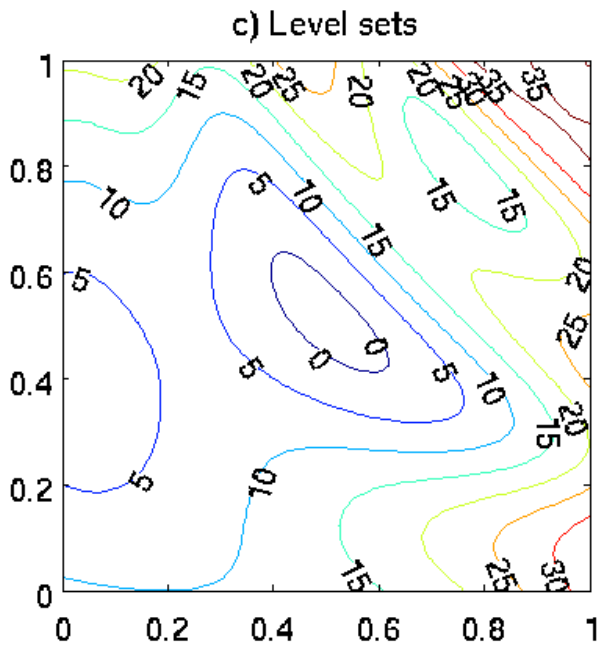
Disadvantages

- Doesn't take into account the model

Example: Toy function

$$y = 2 + 0.25(x_2 - 5x_1^2)^2 + (1 - 5x_1)^2 + 2(2 - 5x_2)^2 + 7\sin(2.5x_1)$$





Example: initial approximation

Initial experimental design contains 10 points
Let us build GP based approximation

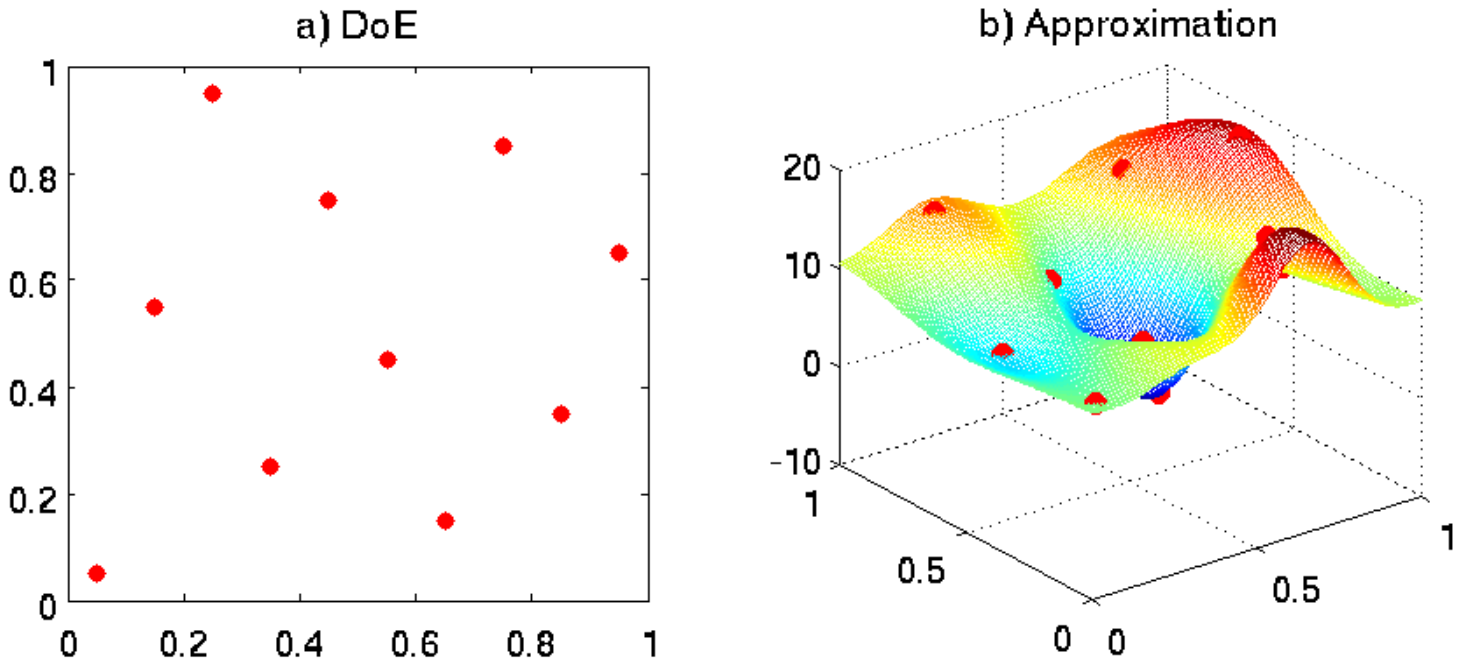


Figure – Initial training set and approximation

Example: Adaptive DoE

70 points were added using MaxVar criterion

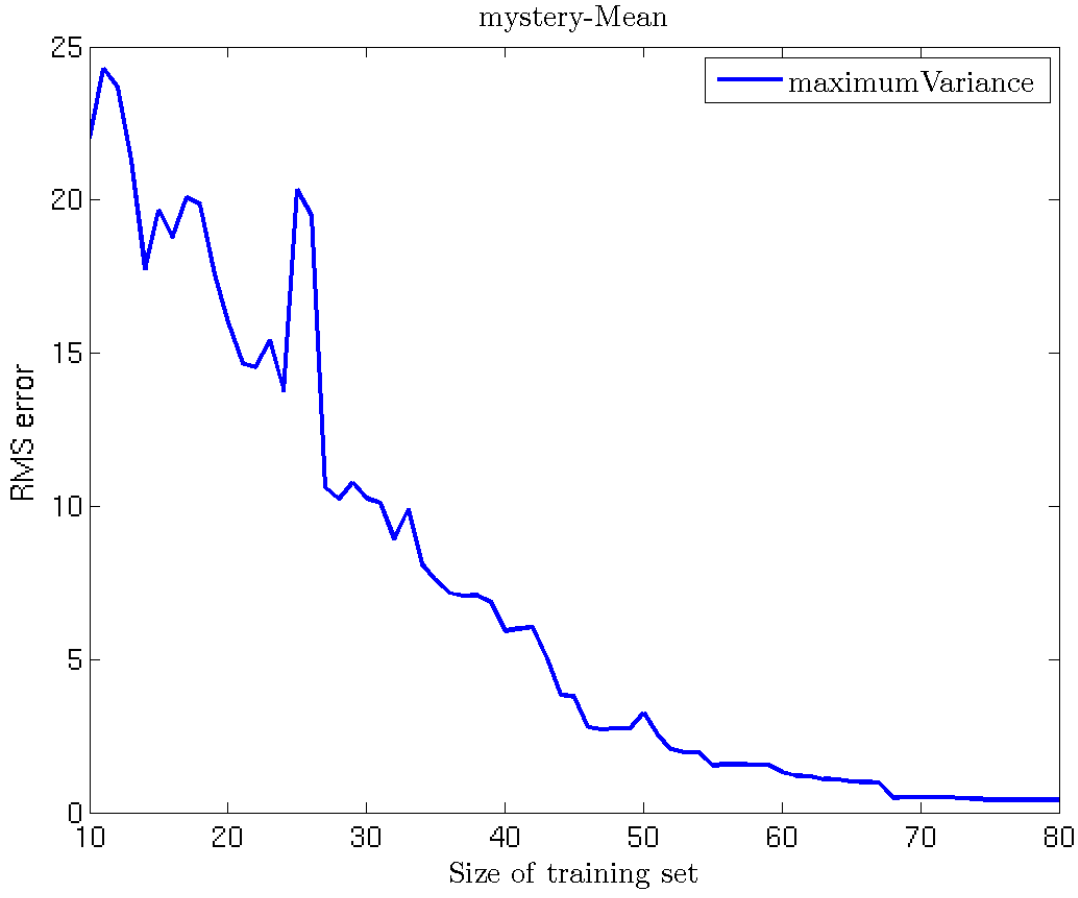


Figure – Approximation error vs size of the training set

Example: final approximation

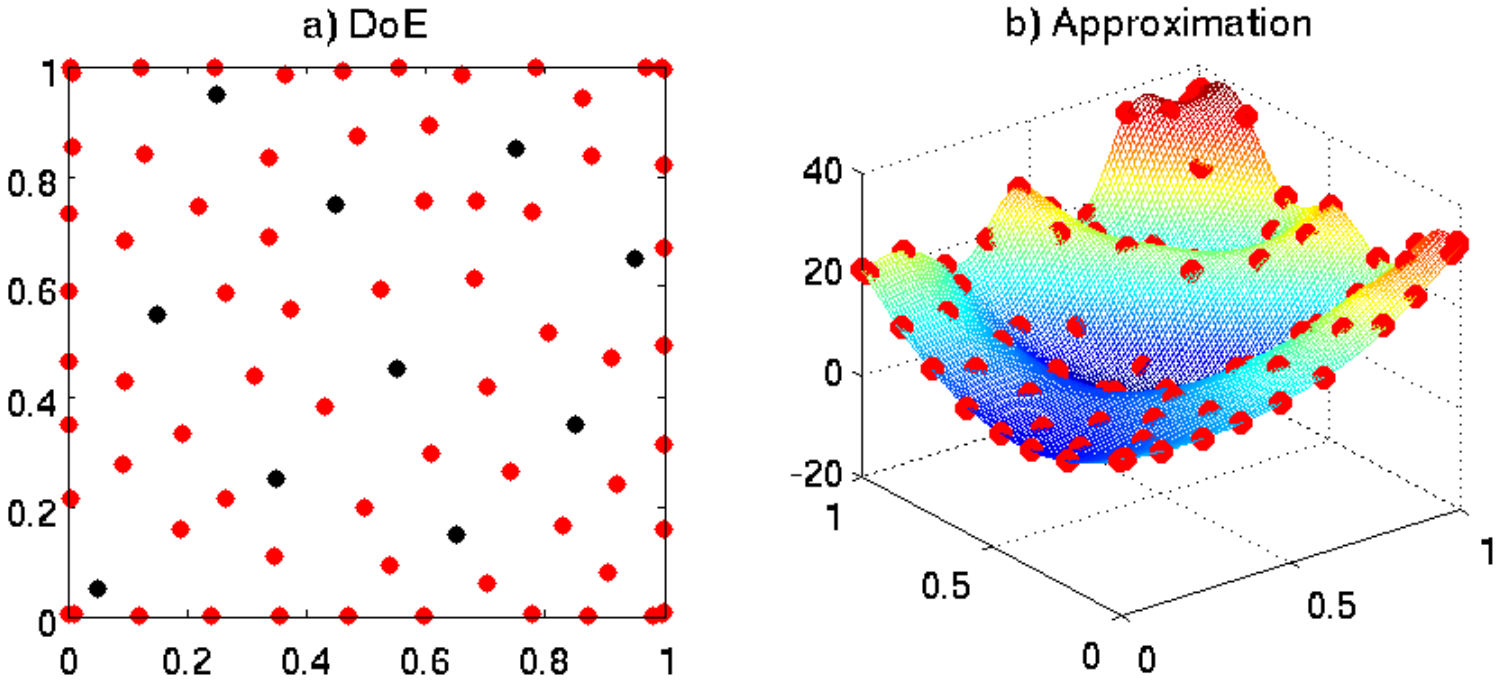


Figure – Final training set and approximation

Example: results

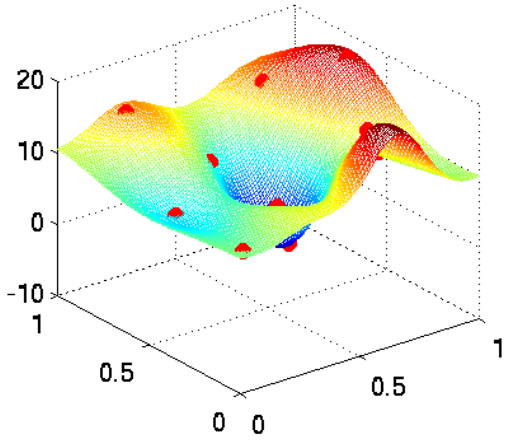


Figure – Initial model

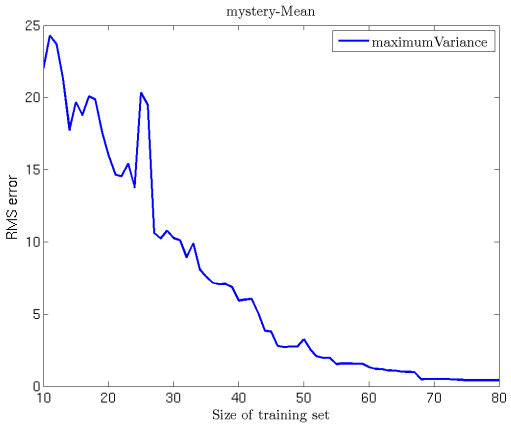


Figure – Error vs training set size

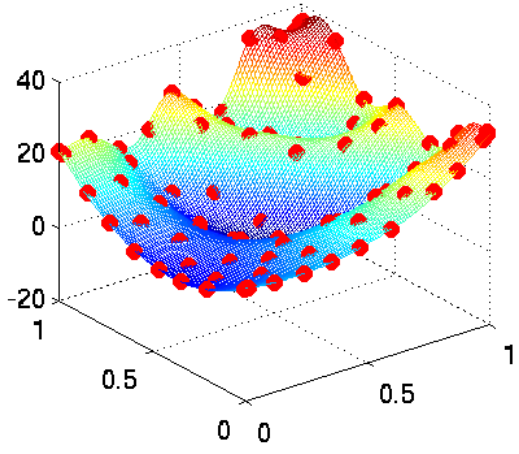


Figure – Final model

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!!!

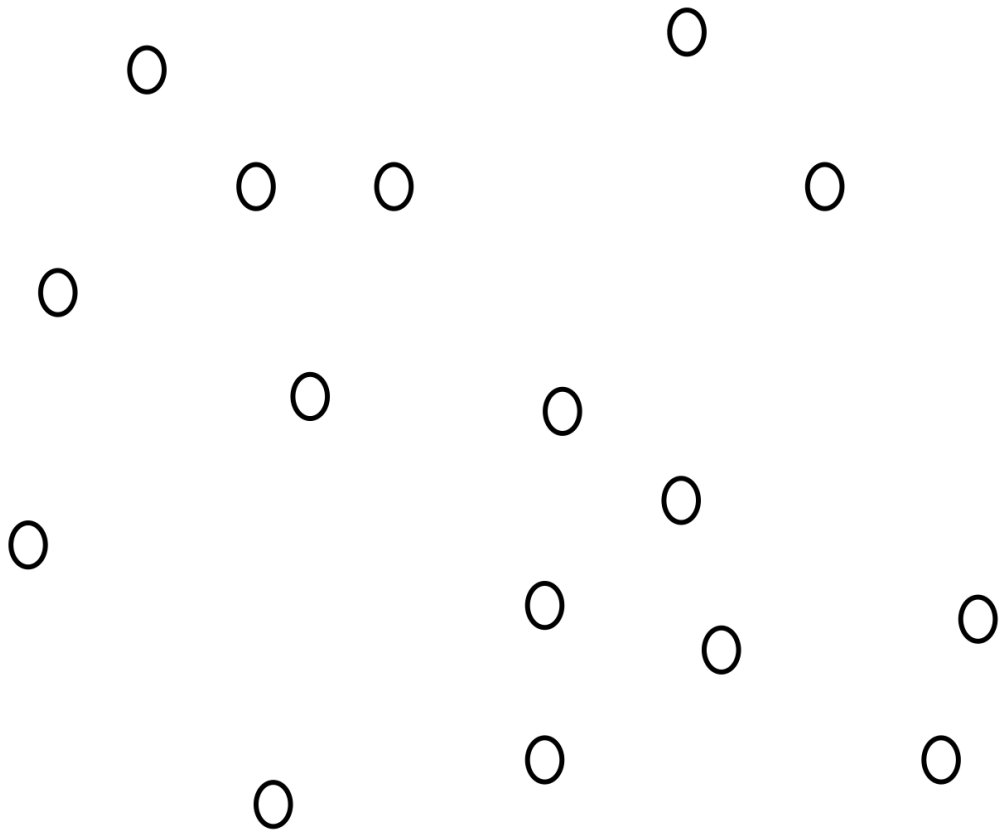
Simple Idea

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

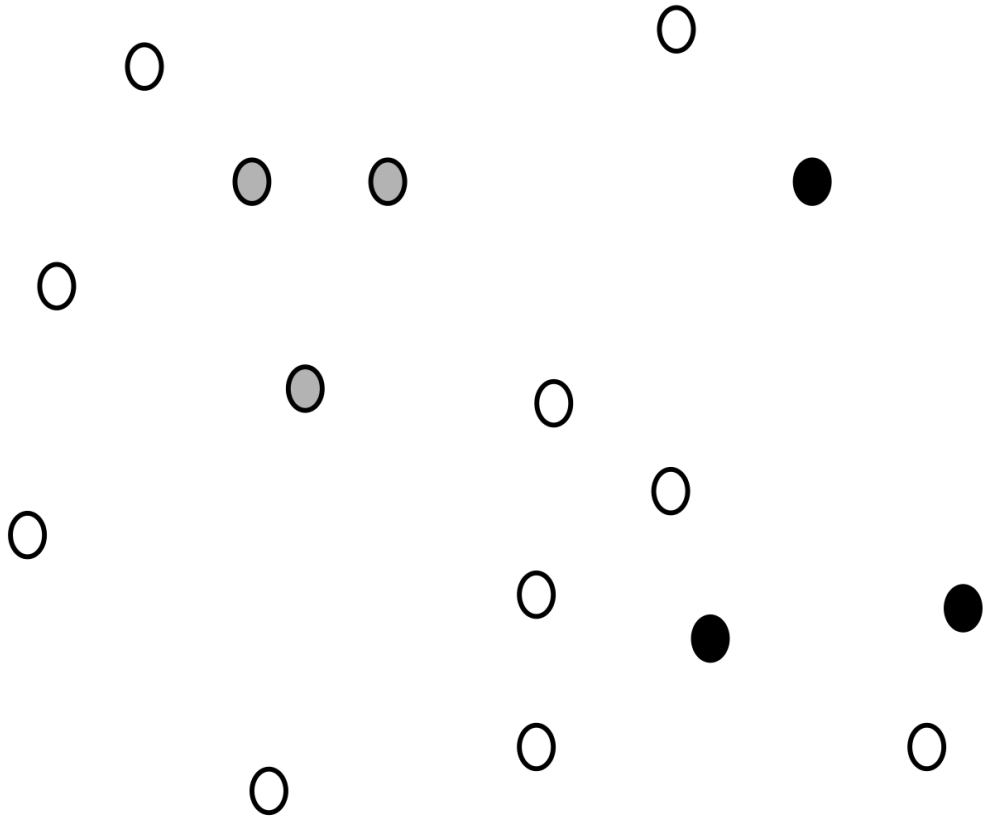
Active Learning Heuristic

- 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?)

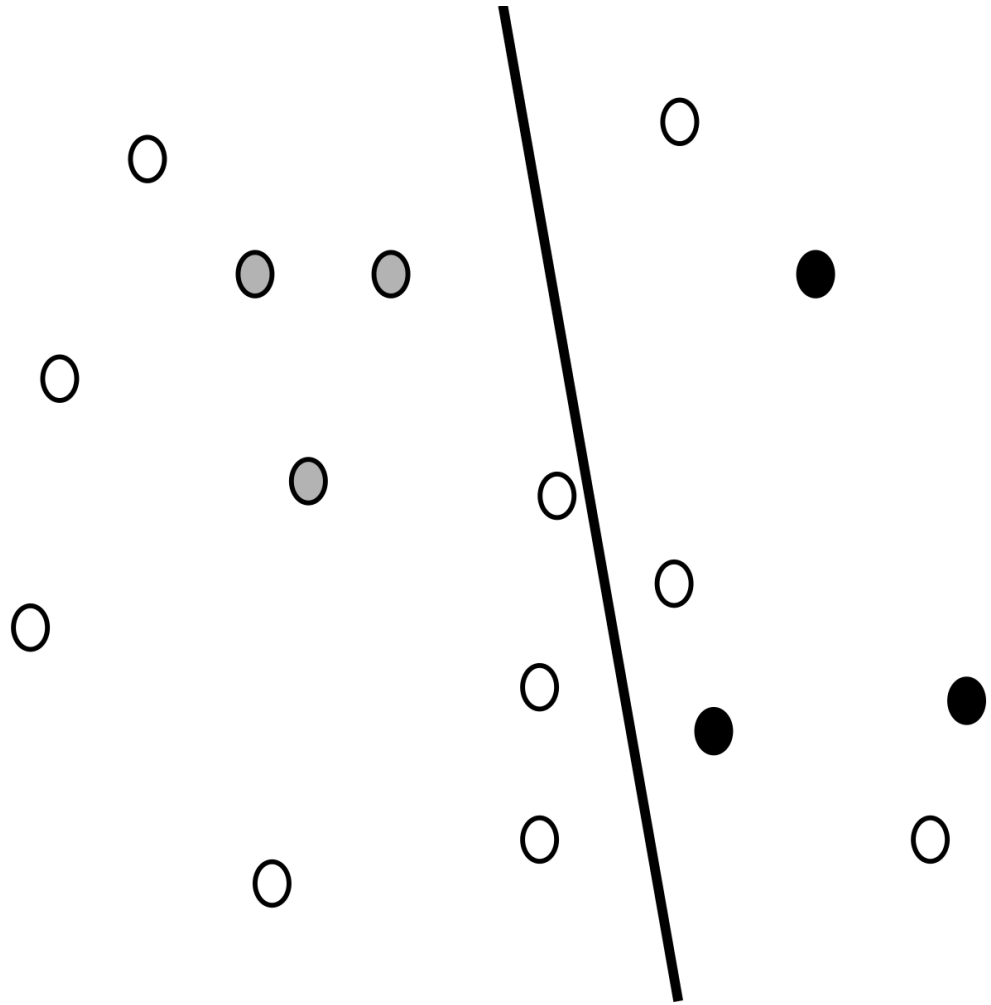
Start: Unlabeled Data



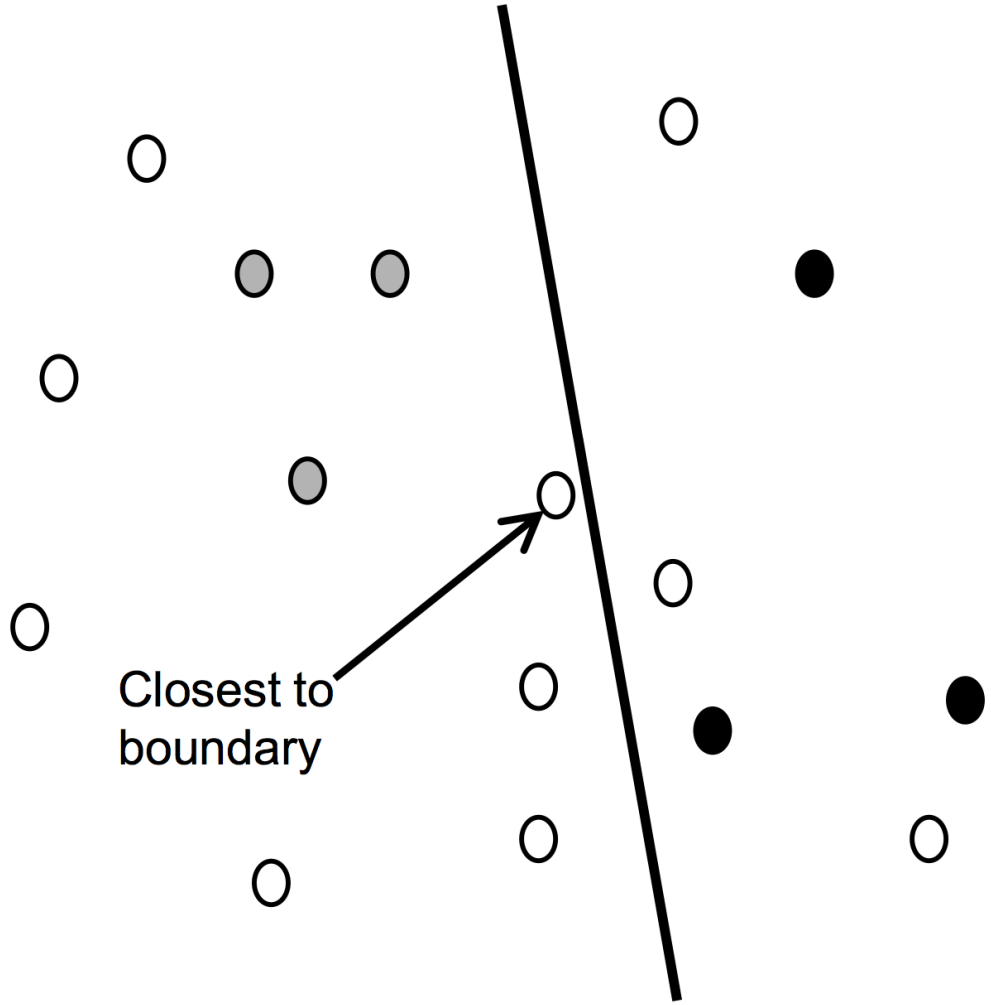
Label a Random Subset



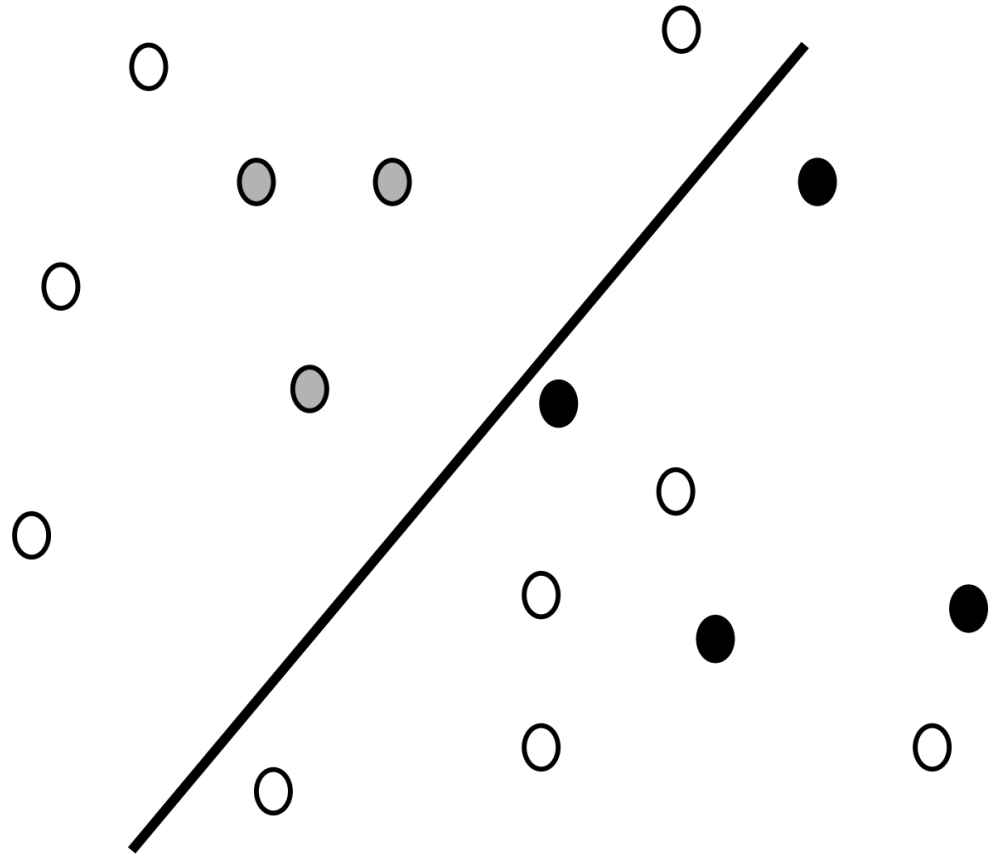
Fit a Classifier to Labeled Data



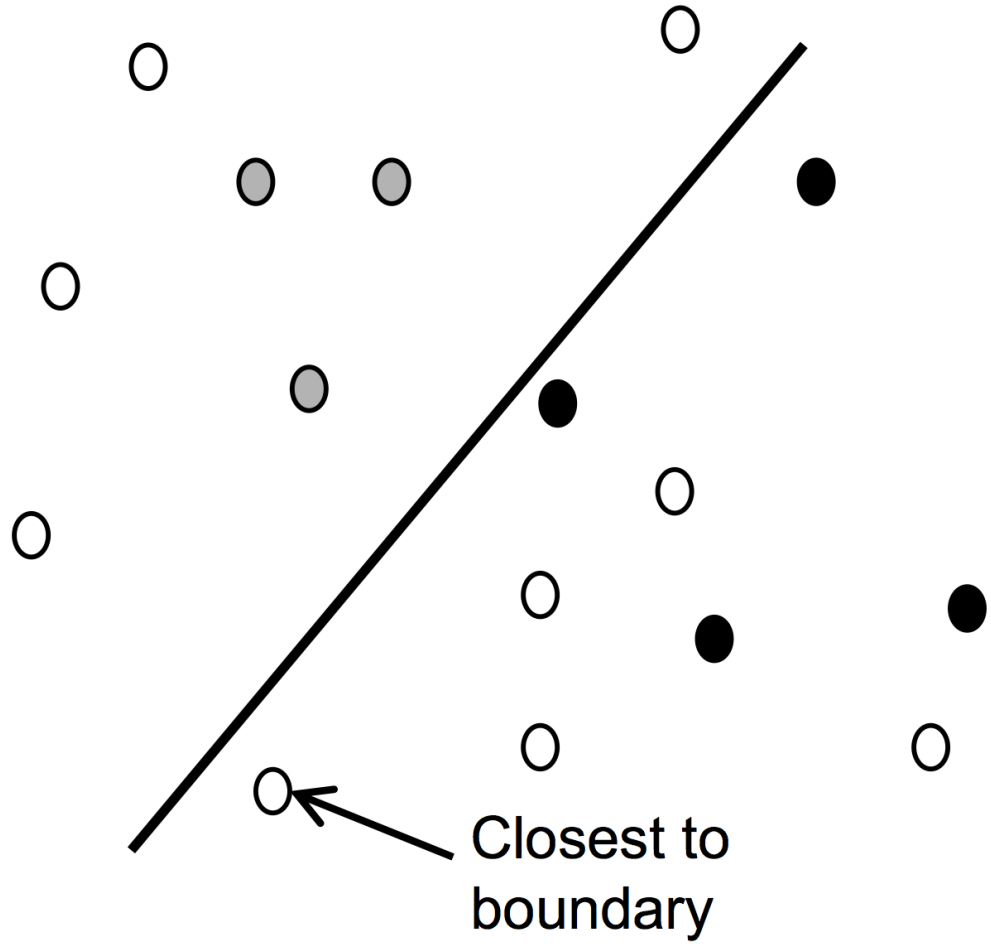
Pick the Best Next Point To Label



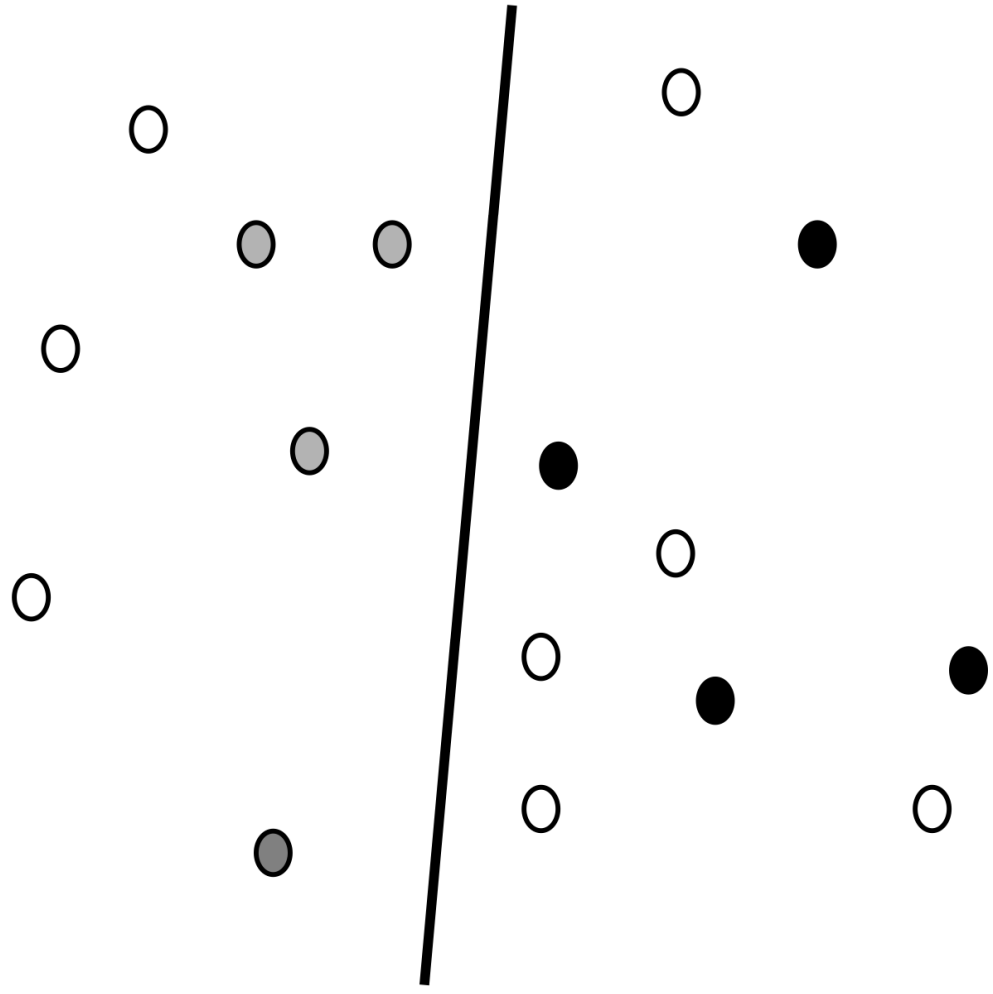
Fit a Classifier to Labeled Data



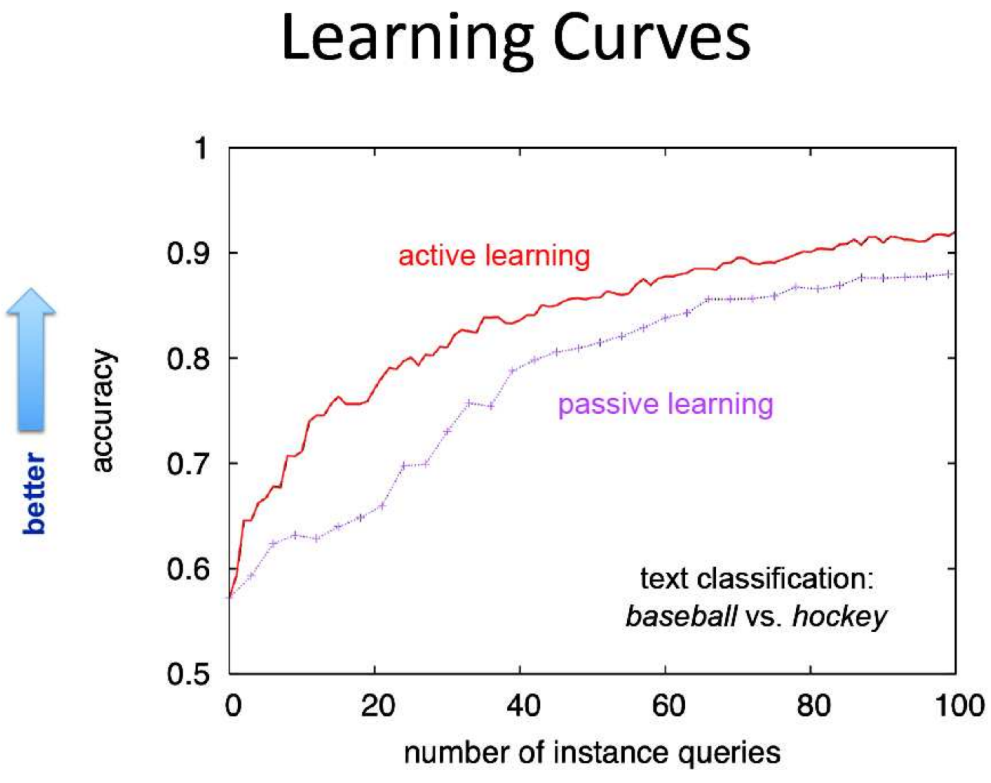
Pick the Best Next Point To Label



Fit a Classifier to Labeled Data



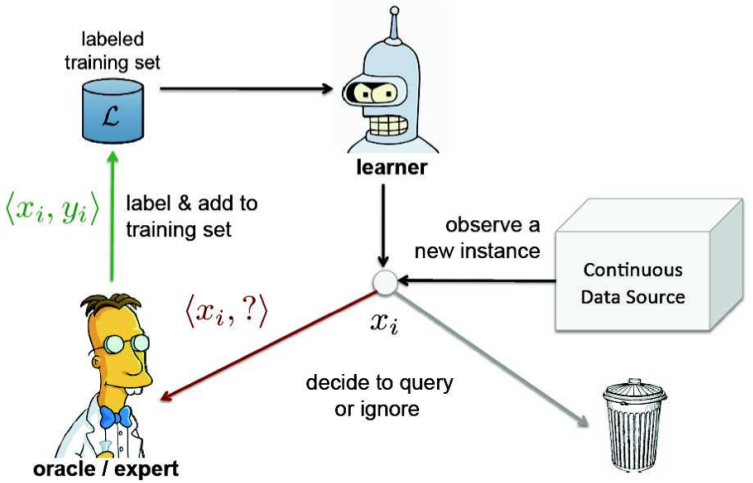
- Passive Learning curve: Randomly selects examples to get labels for
- Acctive learning curve: Active learning examples to get labels for

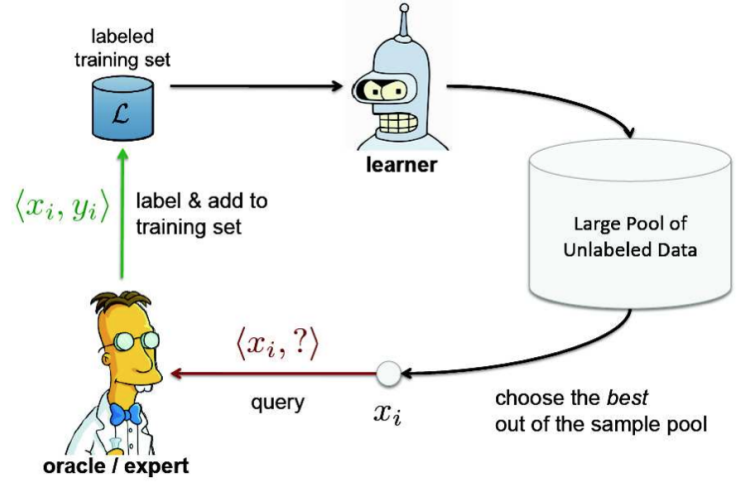


Types of Active Learning

Largely falls into one of these two types:

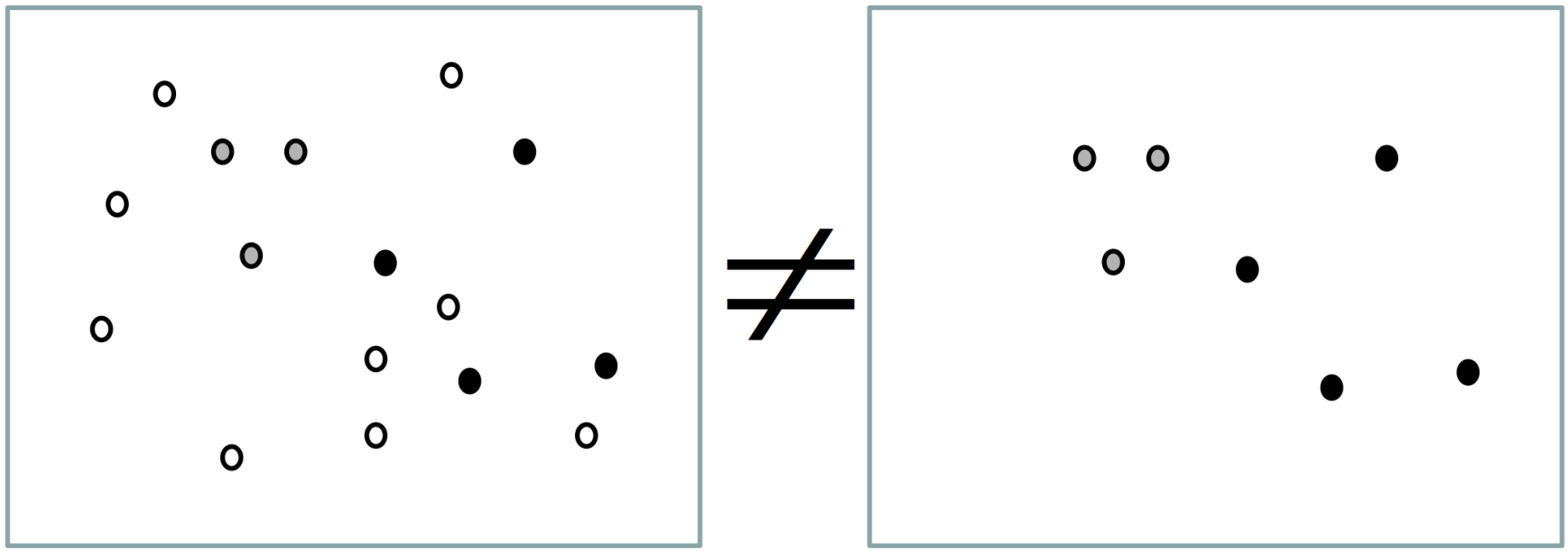
- Stream-Based Active Learning
  - Consider one unlabeled example at a time
  - Decide whether to query its label or ignore it
- Pool-Based Active Learning
  - Given: a large unlabeled pool of examples
  - Rank examples in order of informativeness
  - Query the labels for the most informative example(s)





Biased Sampling

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

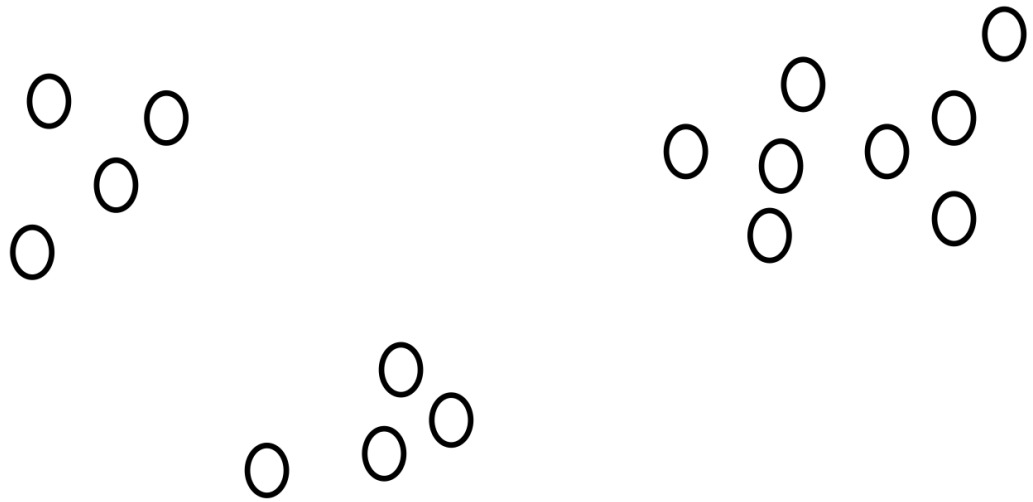


Two Rationales for Active Learning

- 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

Notations

- Problem: train hypothesis $h : \mathbb{X} \to Y$ 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 $S_{i-1} \cup (\mathbf{x}_i, y_i)$

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

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

- Margin sampling principle

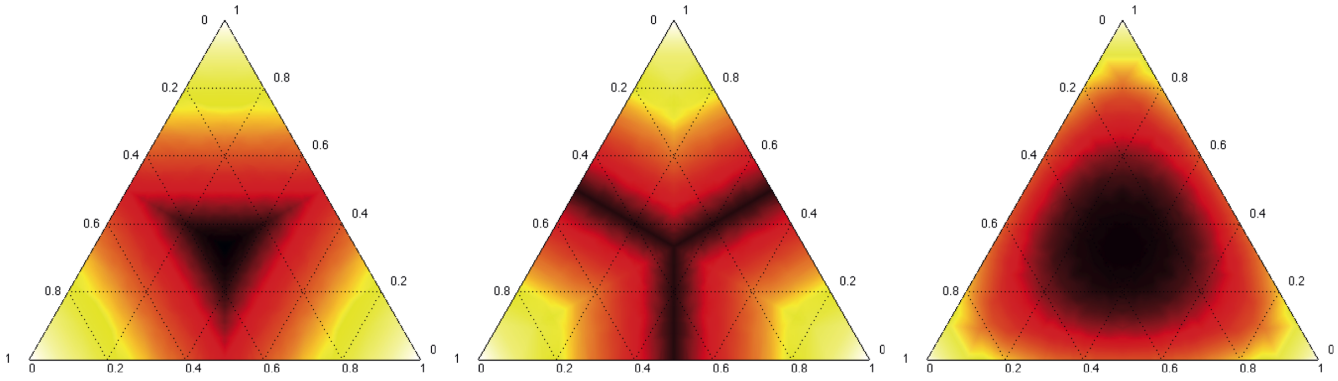
$$\mathbf{x}_i = \arg\min_{\mathbf{x}\in\mathbb{X}} (p_1(\mathbf{x}) - p_2(\mathbf{x}))$$

- Maximum entropy principle

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

Uncertainty sampling

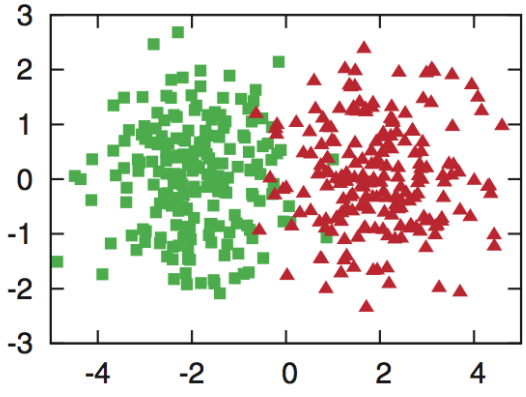
- In case of two classes these three principles are equivalent
- In a multi-class setting there are differences
- Below we show contour lines the corresponding criteria



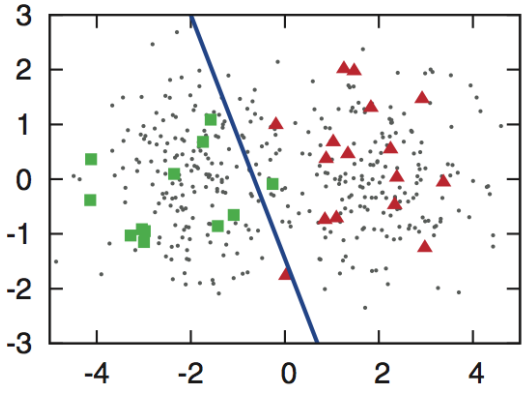
least confident margin entropy

Active vs. Passive learning

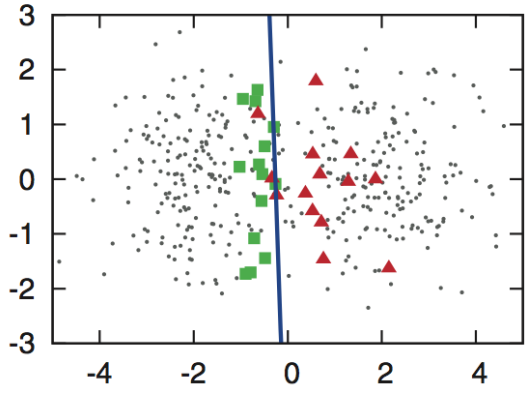
- Example. Synthetic dataset: $m = 30$, $m + k = 400$
  - two Gaussian density
  - logistic regression is constructed using 30 randomly selected objects
  - logistic regression is constructed using 30, adaptively selected using active learning



(a) a 2D toy data set



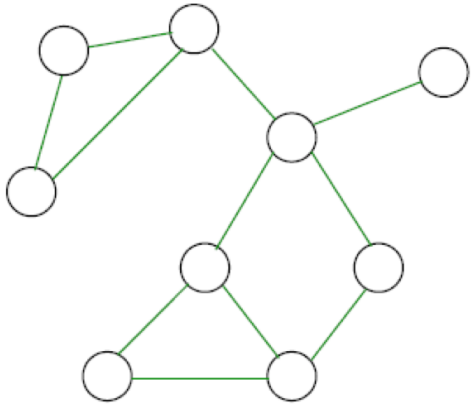
(b) random sampling



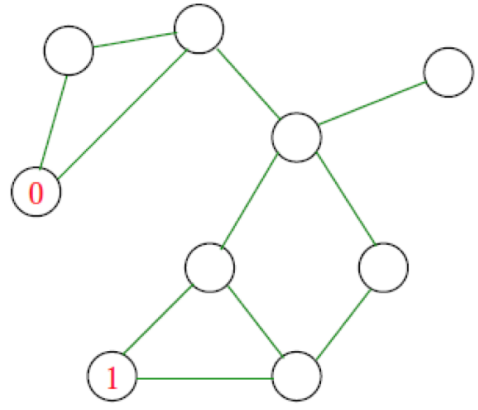
(c) uncertainty sampling

Uncertainty Sampling based on Label-Propagation

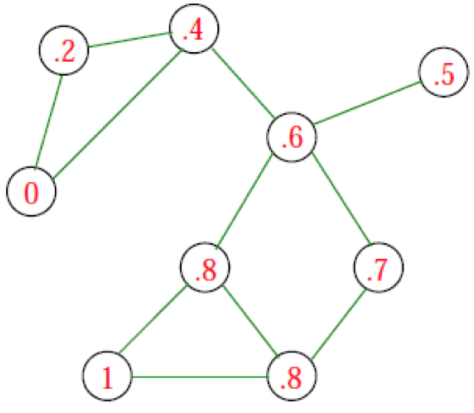
(1) Build neighborhood graph



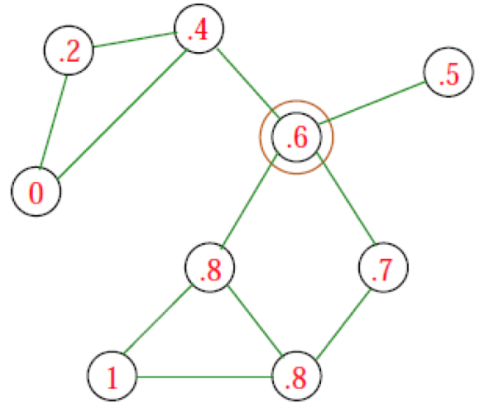
(2) Query some random points



(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

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

  where $\widehat{p}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^{T} 1_{h_t(\mathbf{z})=y}$
- Maximum of an average KL-divergence principle: select those $\mathbf{x}_i$ on which $P_t(y|\mathbf{x}_i)$ are the most different

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

  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 $\mathbf{z} \in \mathbb{X}$ 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 $\nabla_\theta \widehat{R}(\theta; \mathbf{z}, y)$ is a gradient vector the loss function
- We calculate

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

Expected Error Reduction

- Select those $\mathbf{x}_i$, which after re-learning provides the most reliable classification of the non-labeled feature vectors from $\mathbb{X}$
- For $\mathbf{z} \in \mathbb{X}$ 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

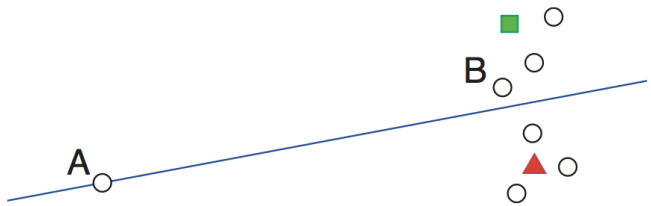
$$\mathbf{x}_i = \arg\max_{\mathbf{z} \in \mathbb{X}} \sum_{y \in Y} P(y|\mathbf{z}) \sum_{j=m+1}^{m+k} P_{\mathbf{z}y}(h_{\mathbf{z}y}(\mathbf{x}_j)|\mathbf{x}_j)$$

- Minimum Entropy of non-labeled data principle

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

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

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

- It 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