

Active Learning. Adaptive Design of Experiments

Evgeny Burnaev

Skoltech, Moscow, Russia

1 Active Learning

2 Active Learning: Adaptive Design of Experiments for Regression

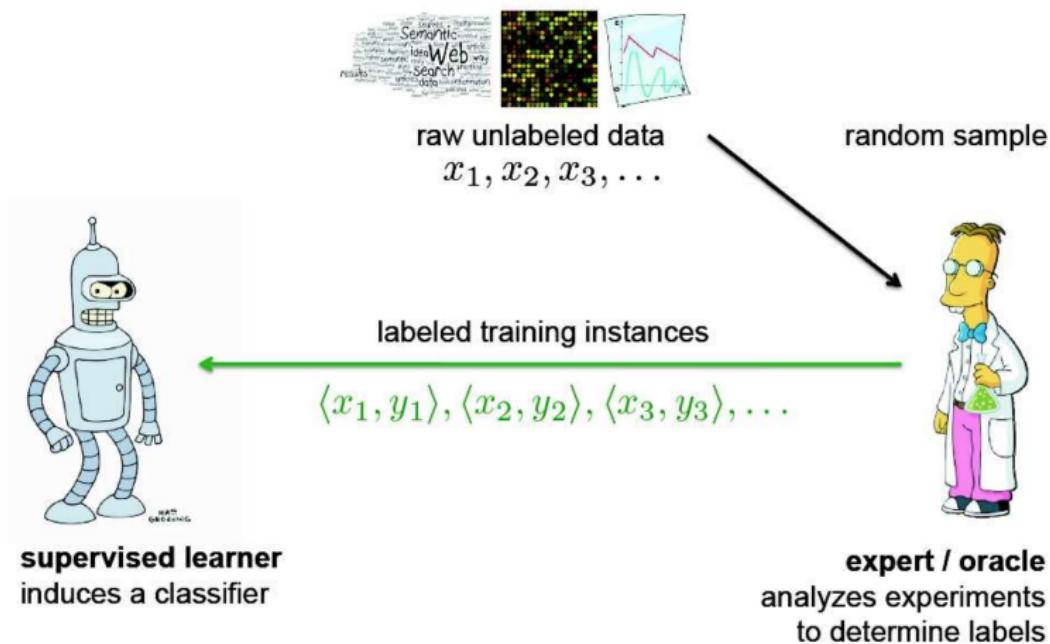
3 Active Learning: Classification

1 Active Learning

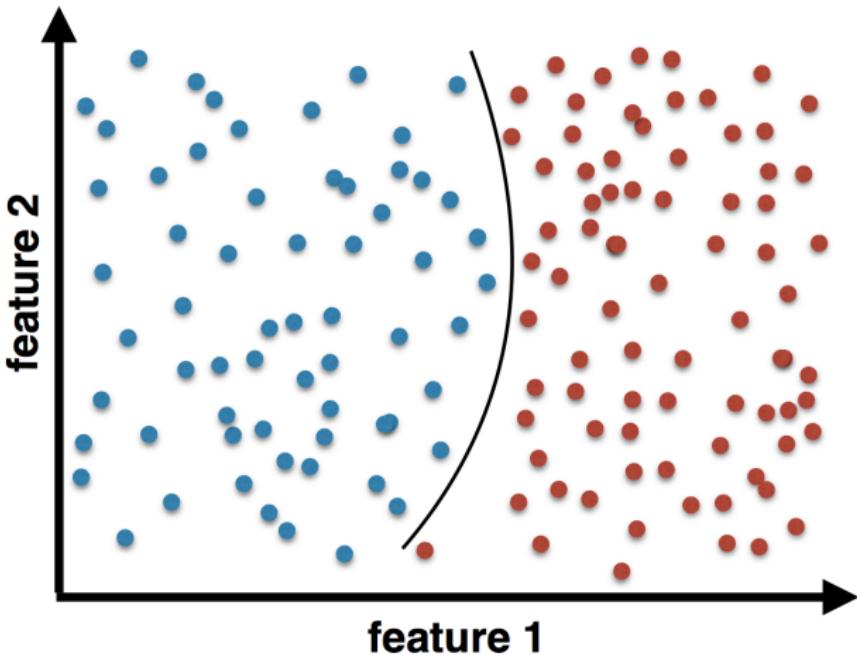
2 Active Learning: Adaptive Design of Experiments for Regression

3 Active Learning: Classification

(Passive) Supervised Learning

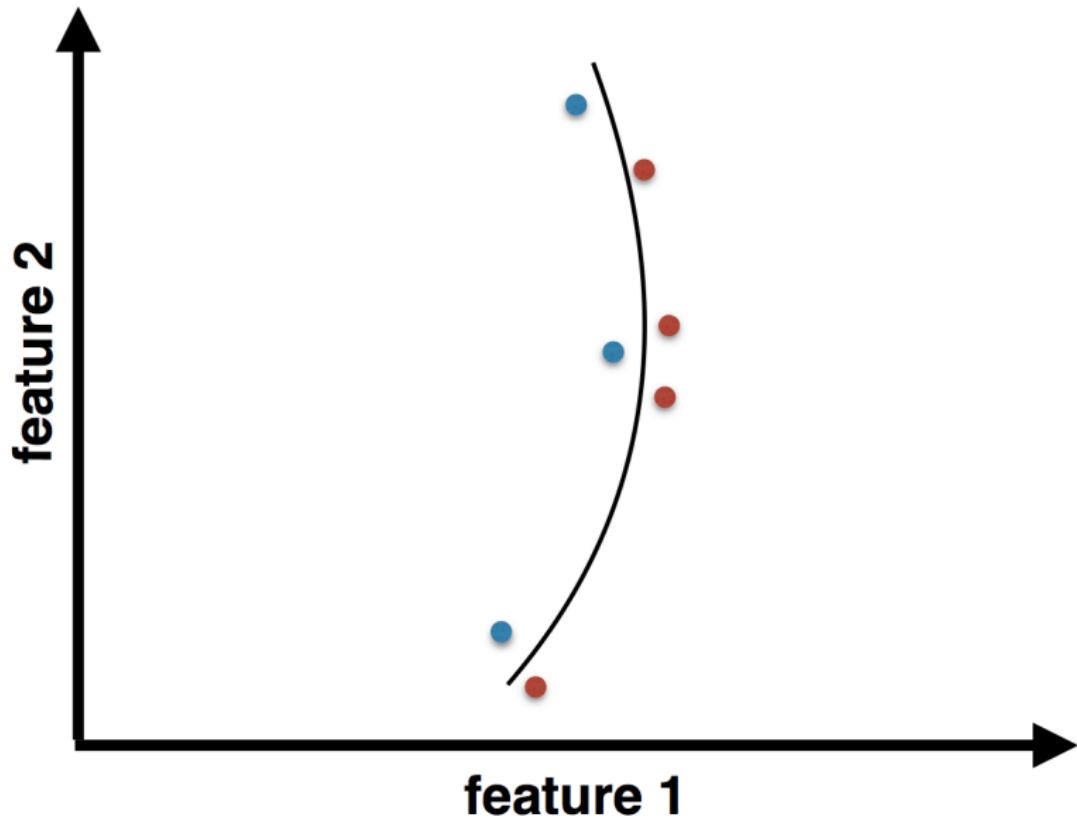


So what's wrong with Supervised Learning

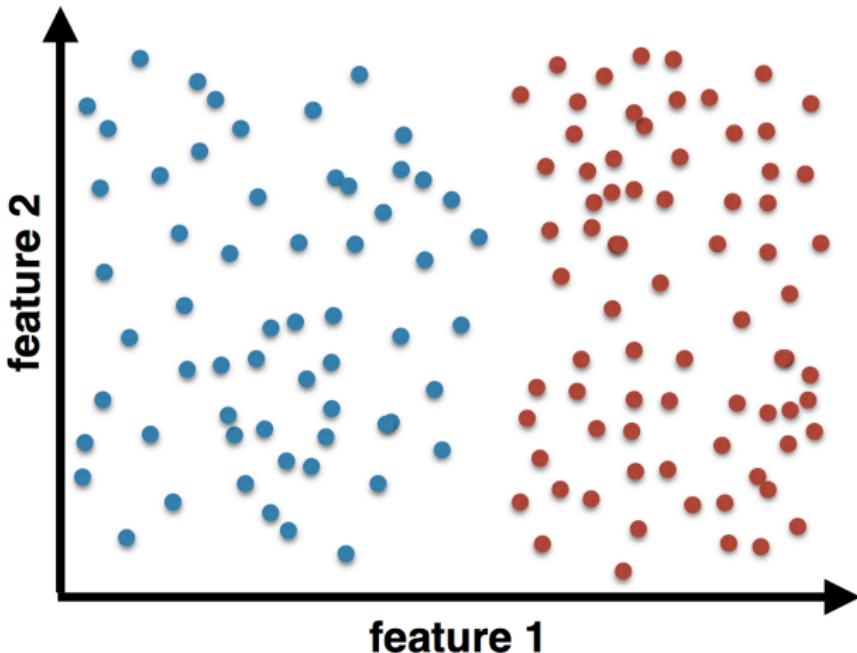


- Traditional approach almost universally adopted

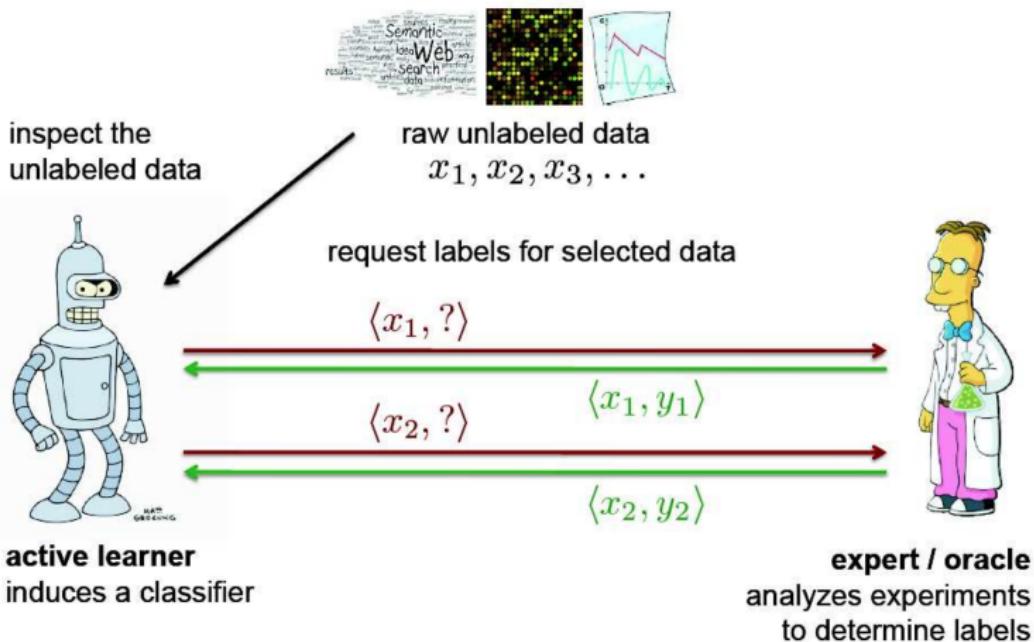
Well, we actually only needed this!



So this was a complete waste of time!

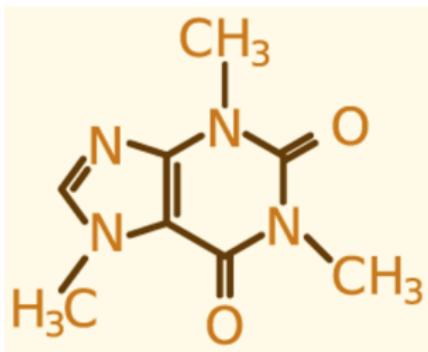


- Random sampling inevitably leads to redundancy



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

- 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

1 Active Learning

2 Active Learning: Adaptive Design of Experiments for Regression

3 Active Learning: Classification

Definition

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

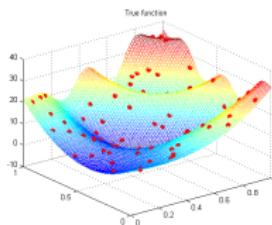


Figure – Unknown function

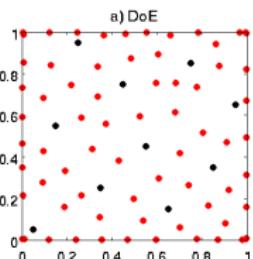


Figure – Initial and added points

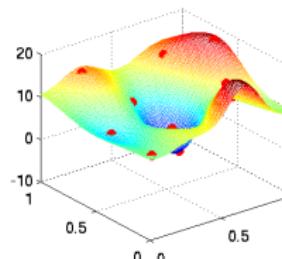


Figure – Initial model

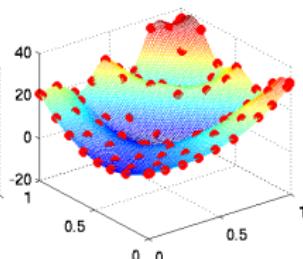


Figure – Final model

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

Adaptive DoE:

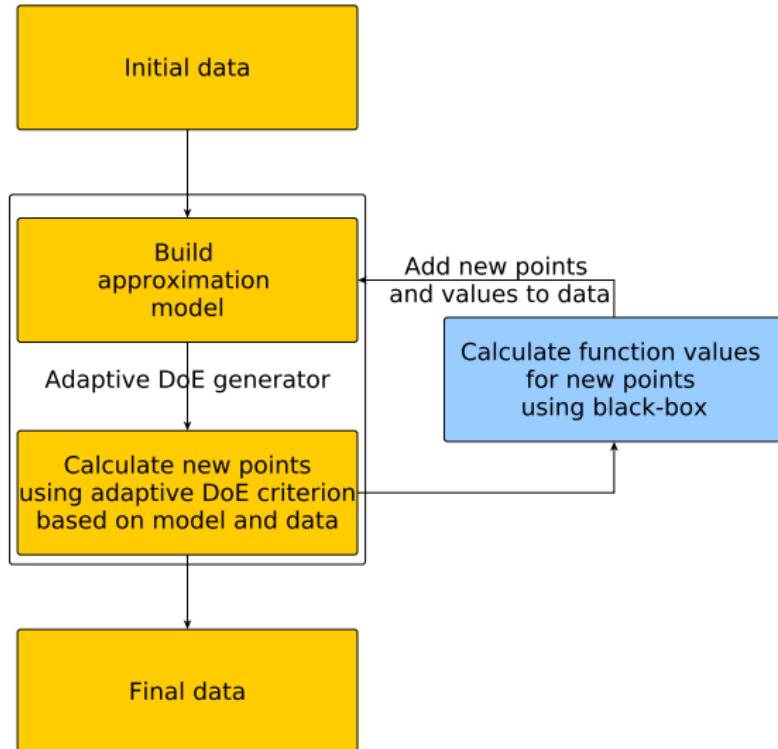
- Generate initial sample using space-filling technique
- Build a model using the obtained training set
- Update the training set by iteratively adding points which improves the model the most

Adaptive DoE 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:

- ➊ Approximation contains information about the dependency



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

- ➋ Adaptive update of training set gives flexibility in sample size



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

Reasons for such approach:

- ① Approximation contains information about the dependency



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



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

Reasons for such approach:

- ① Approximation contains information about the dependency



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



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

Reasons for such approach:

- ① Approximation contains information about the dependency



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



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

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

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

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

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

- Maximum variance criterion:

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

where $\hat{\sigma}^2(\mathbf{x}|S)$ is an error at point \mathbf{x} of the model trained on $S = (\mathbf{X}, \mathbf{y})$

- As $\hat{\sigma}^2(\mathbf{x}|S)$ 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}

- Maximum variance criterion:

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

where $\hat{\sigma}^2(\mathbf{x}|S)$ is an error at point \mathbf{x} of the model trained on $S = (\mathbf{X}, \mathbf{y})$

- As $\hat{\sigma}^2(\mathbf{x}|S)$ 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 \mathbf{X}

- Maximum variance criterion:

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

where $\hat{\sigma}^2(\mathbf{x}|S)$ is an error at point \mathbf{x} of the model trained on $S = (\mathbf{X}, \mathbf{y})$

- As $\hat{\sigma}^2(\mathbf{x}|S)$ 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}

- Maximum variance criterion:

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

where $\hat{\sigma}^2(\mathbf{x}|S)$ is an error at point \mathbf{x} of the model trained on $S = (\mathbf{X}, \mathbf{y})$

- As $\hat{\sigma}^2(\mathbf{x}|S)$ 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}

- Maximum variance criterion:

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

where $\hat{\sigma}^2(\mathbf{x}|S)$ is an error at point \mathbf{x} of the model trained on $S = (\mathbf{X}, \mathbf{y})$

- As $\hat{\sigma}^2(\mathbf{x}|S)$ 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}



- GP-based criterion

- Minimum mean squared error on next iteration:

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

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
- Minimum mean squared error on next iteration:

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

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
- Minimum mean squared error on next iteration:

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

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
- Minimum mean squared error on next iteration:

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

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

- 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

- 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:

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

where

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

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



It leads to noisy values and multiple non-robust minima

The criterion can be rewritten in the following form:

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

where

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

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



It leads to noisy values and multiple non-robust minima

The criterion can be rewritten in the following form:

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

where

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

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



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 $\mathcal{I}_{MV}(\mathbf{x})$

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

Advantages

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

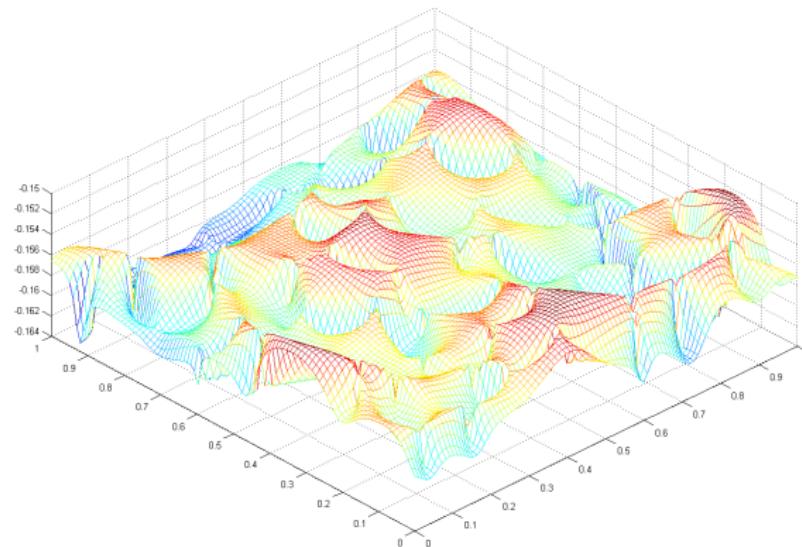
We can solve the problem by combining two criteria: $\mathcal{I}_{\rho_2}(\mathbf{x})$ and $\mathcal{I}_{MV}(\mathbf{x})$

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

Advantages

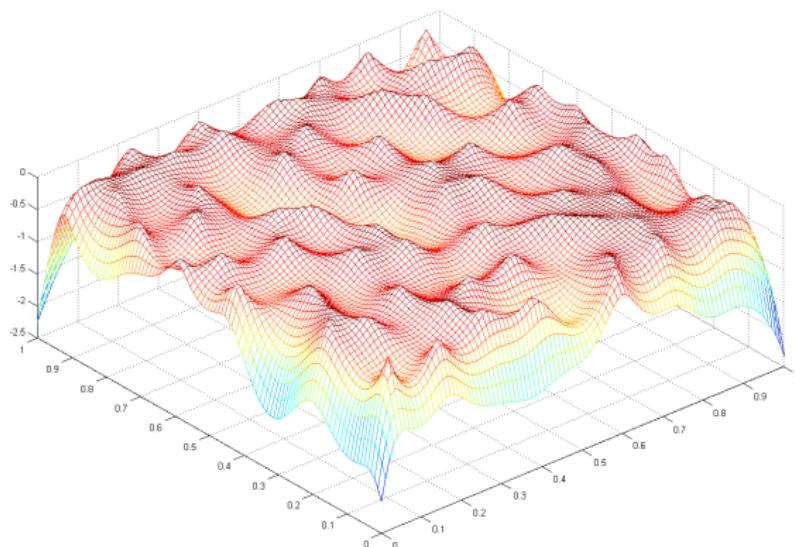
- No issues with denominator
- Still takes into account all the design space

- High modality
- Narrow and non-robust local minima



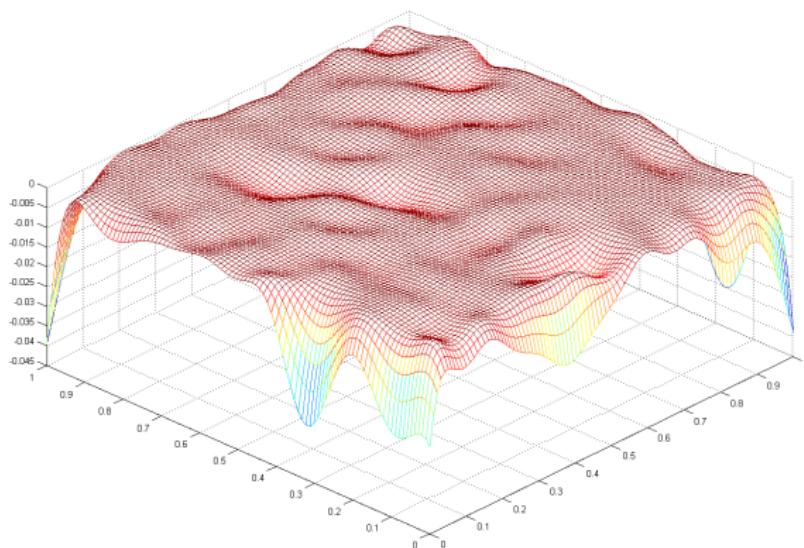
Surface of MaxVar criterion

- 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

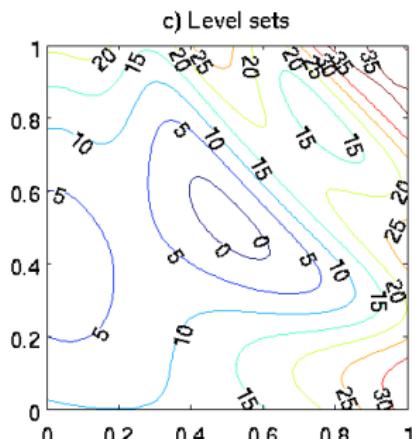
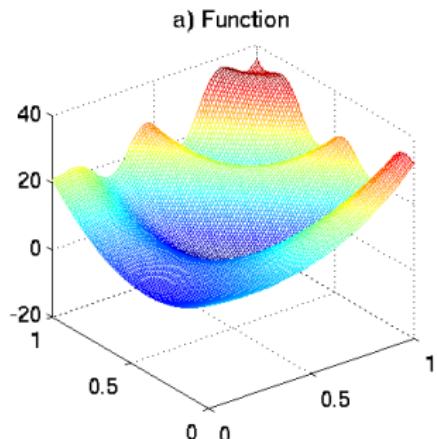
- Very fast computation
- Doesn't depend on the model type
- Samples point uniformly in the design space

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

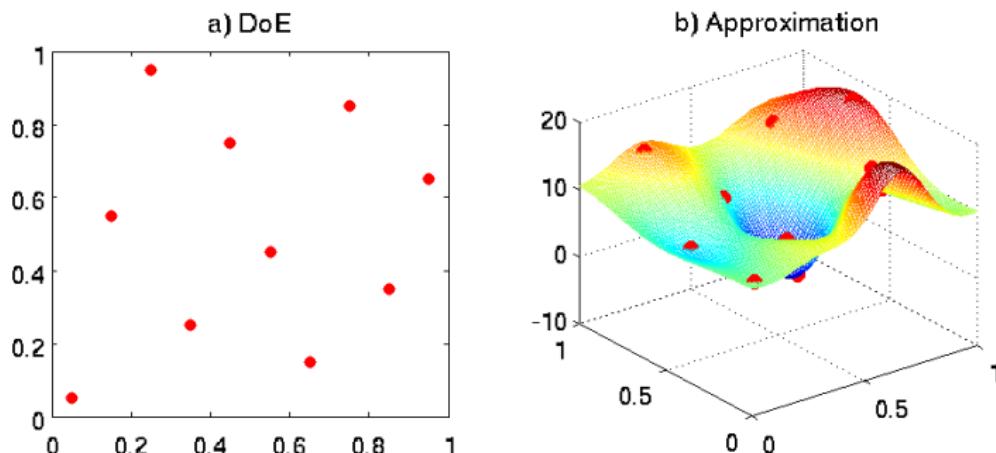


Figure – Initial training set and approximation

70 points were added using MaxVar criterion

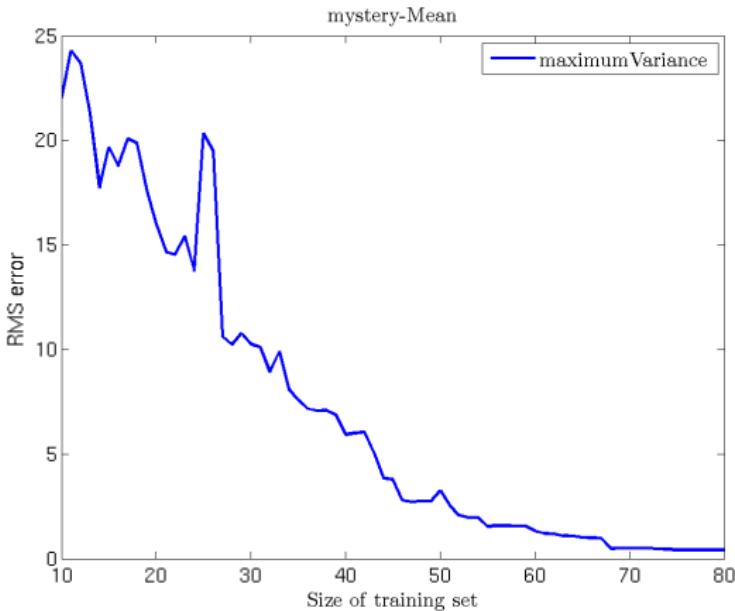


Figure – Approximation error vs size of the training set

Example: final approximation

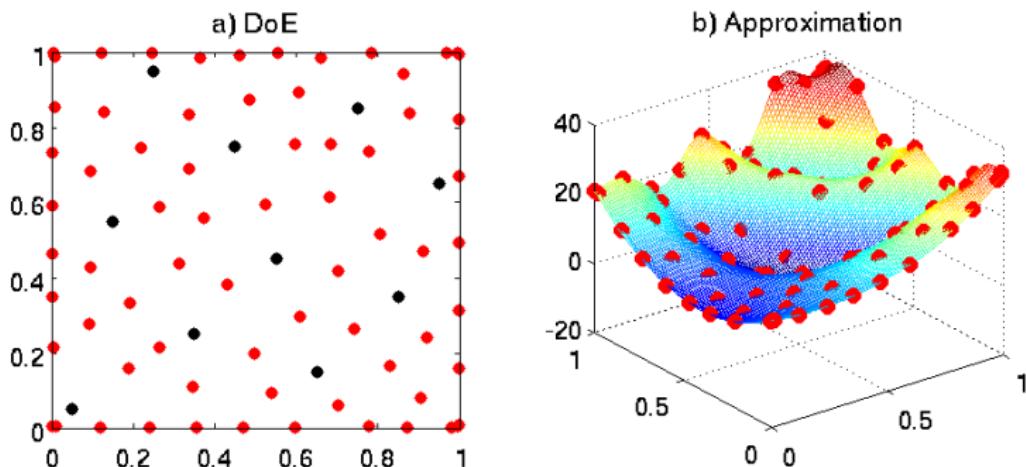


Figure – Final training set and approximation

Example: results

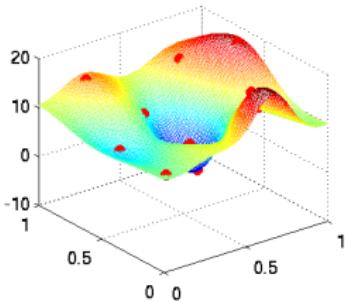


Figure – Initial model

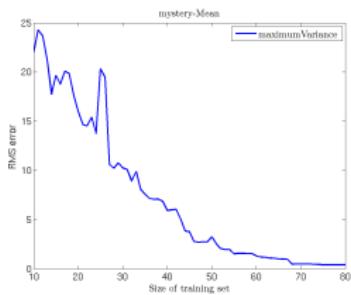


Figure – Error vs training set size

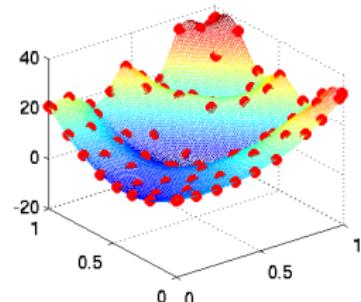


Figure – Final model

1 Active Learning

2 Active Learning: Adaptive Design of Experiments for Regression

3 Active Learning: Classification

- 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!!!

- 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!!!

- 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!!!

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

- 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?)

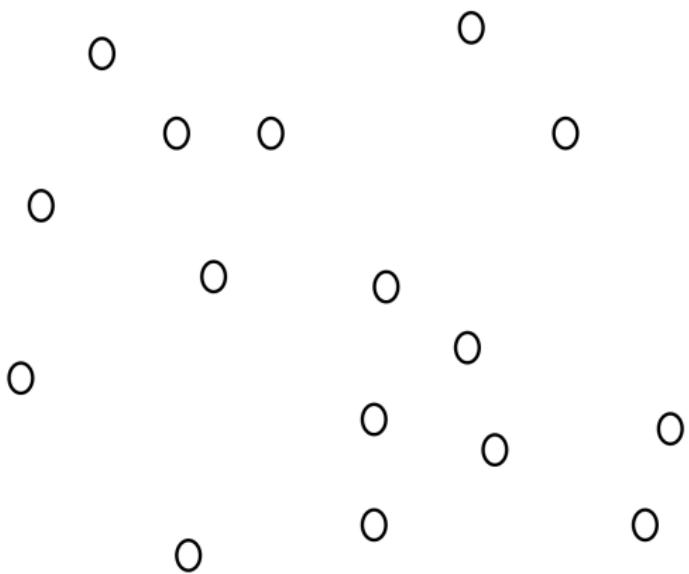
- Start with a pool of unlabeled data
- Pick a few points at random and get their labels
- Repeat the following until we have budget left for getting labels
 - 1. Fit a classifier to the labels seen so far
 - 2. Pick the BEST unlabeled point to get a label for
 - (closest to the boundary?)
 - (most uncertain?)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

- Start 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?)

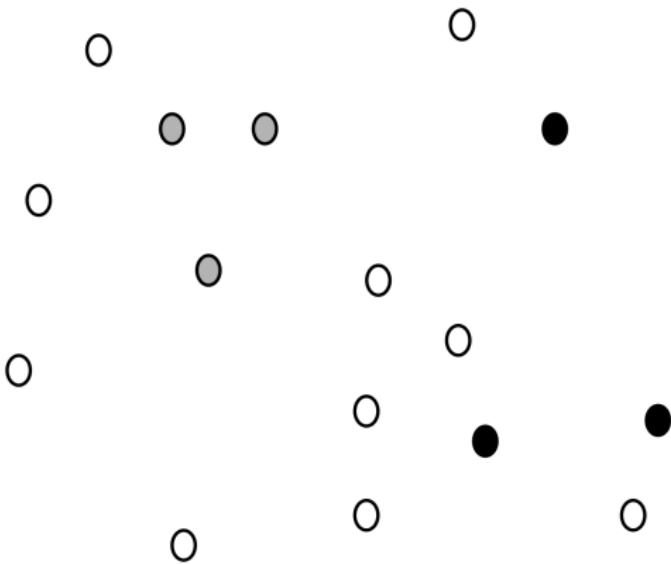
- Start with a pool of unlabeled data
- Pick a few points at random and get their labels
- Repeat the following until we have budget left for getting labels
 1. Fit a classifier to the labels seen so far
 2. Pick the BEST unlabeled point to get a label for
 - (closest to the boundary?)
 - (most uncertain?)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

- Start 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?)

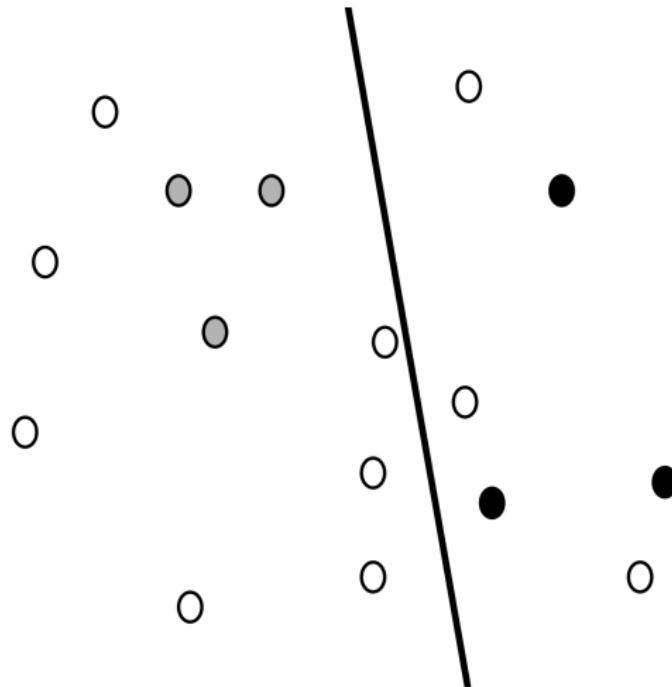
Start: Unlabeled Data



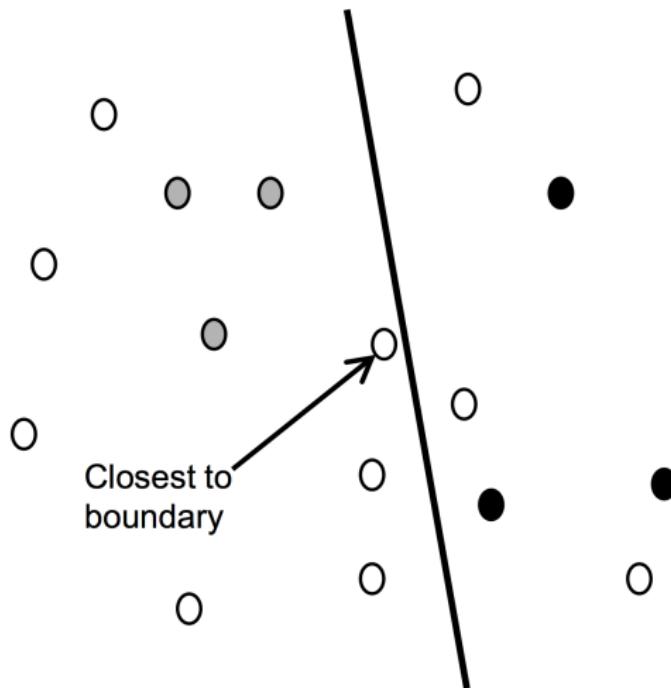
Label a Random Subset



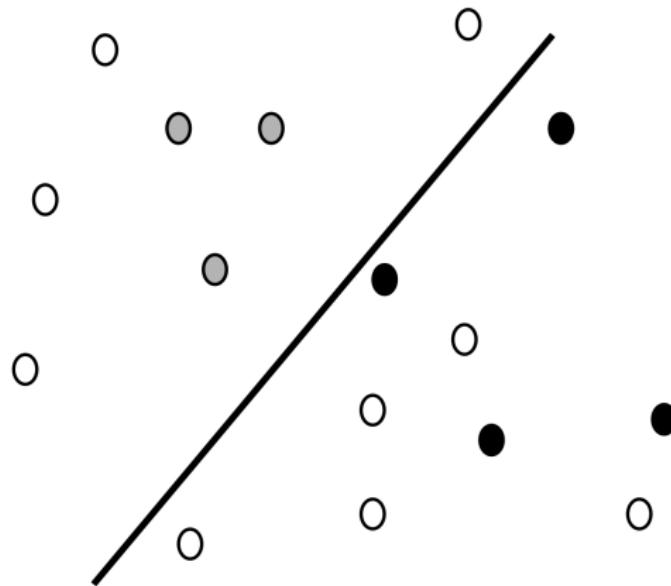
Fit a Classifier to Labeled Data



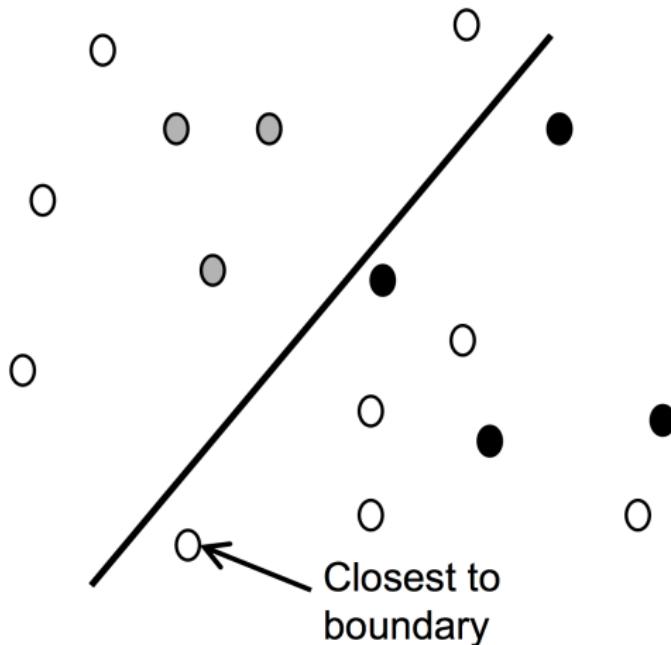
Pick the Best Next Point To Label



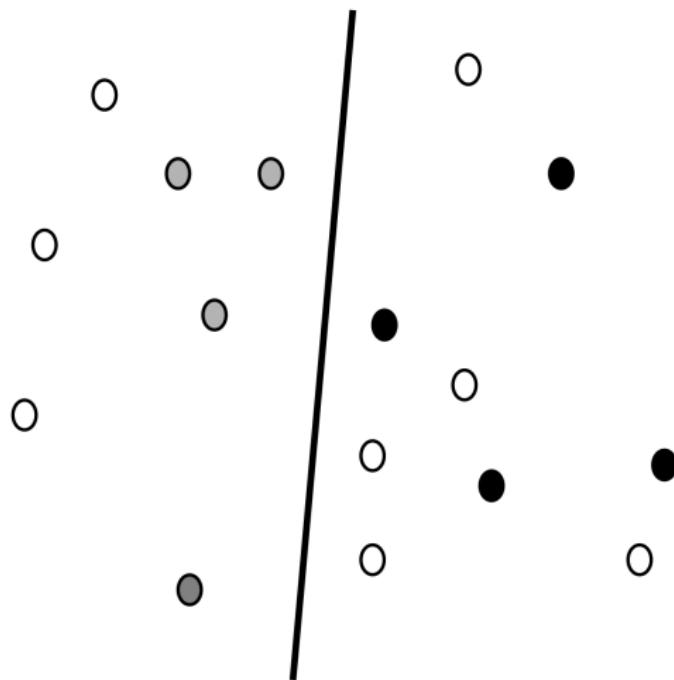
Fit a Classifier to Labeled Data



Pick the Best Next Point To Label

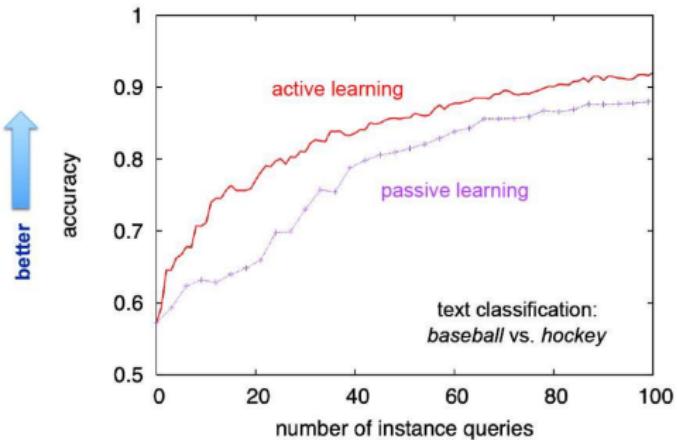


Fit a Classifier to Labeled Data



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

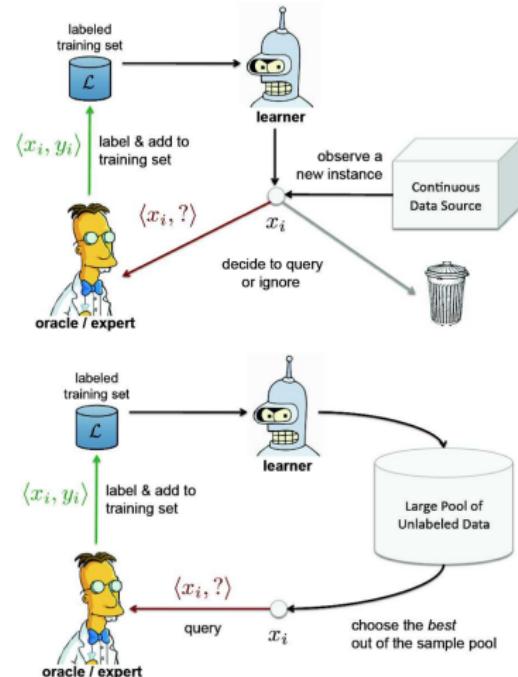
Learning Curves



Types of Active Learning

Largely falls into one of these two types:

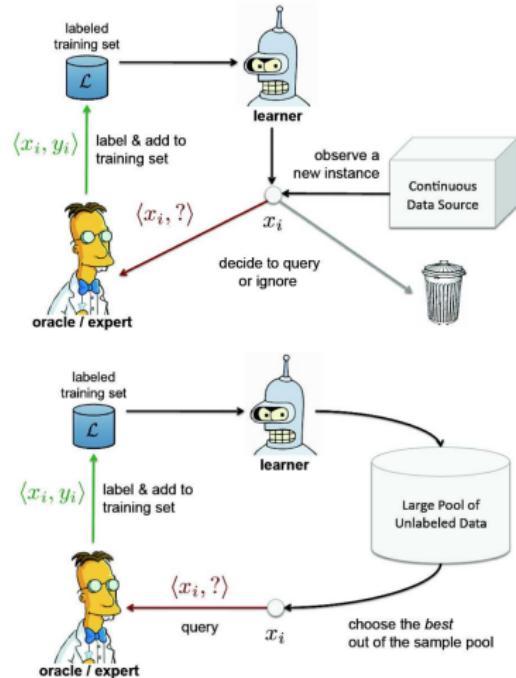
- Stream-Based Active Learning
 - Consider one unlabeled example at a time
 - Decide whether to query its label or ignore it
- Pool-Based Active Learning
 - Given: a large unlabeled pool of examples
 - Rank examples in order of informativeness
 - Query the labels for the most informative example(s)



Types of Active Learning

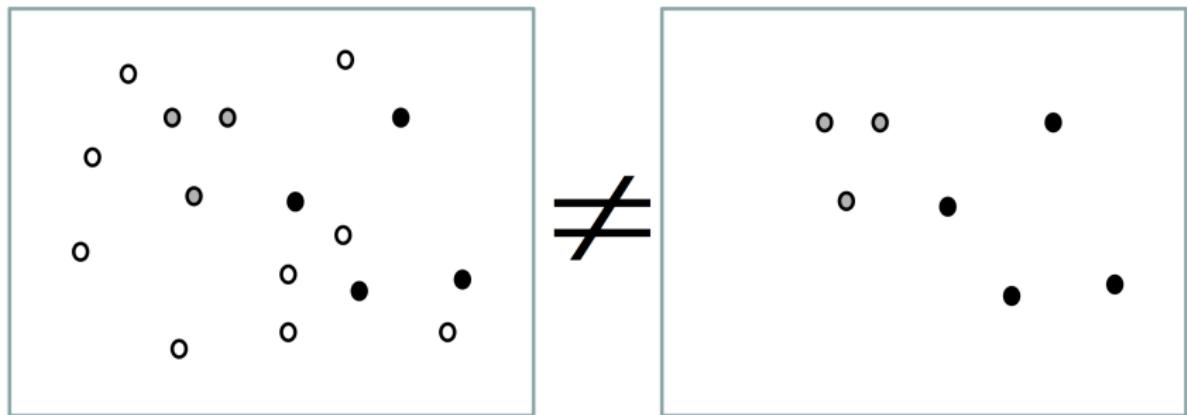
Largely falls into one of these two types:

- 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)



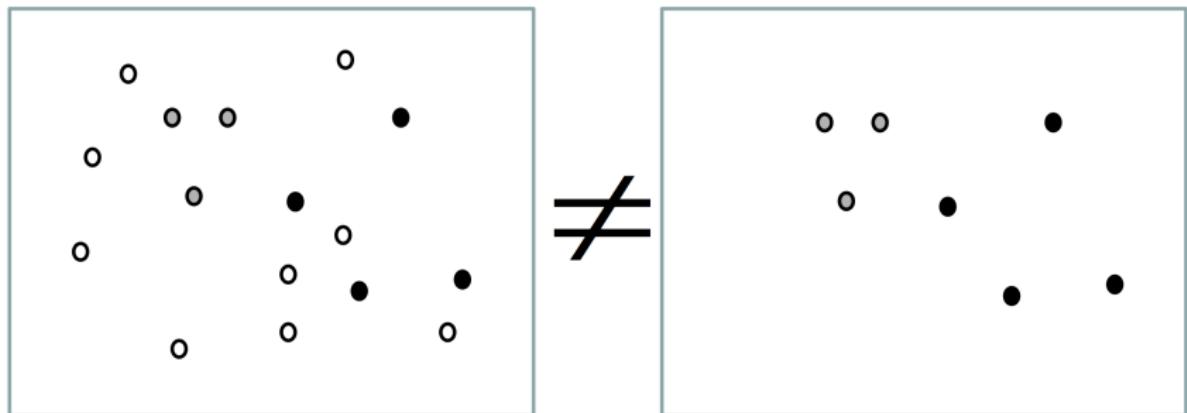
Biased Sampling

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



Biased Sampling

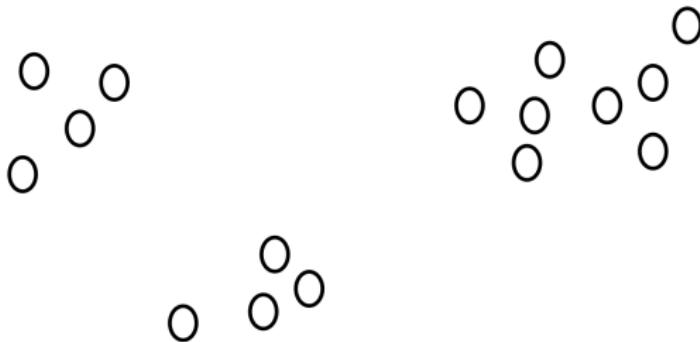
- The labeled points may not be representative of the underlying distribution
- This can increase error, even with infinitely many labeled samples



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

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

If the data looked like this ...

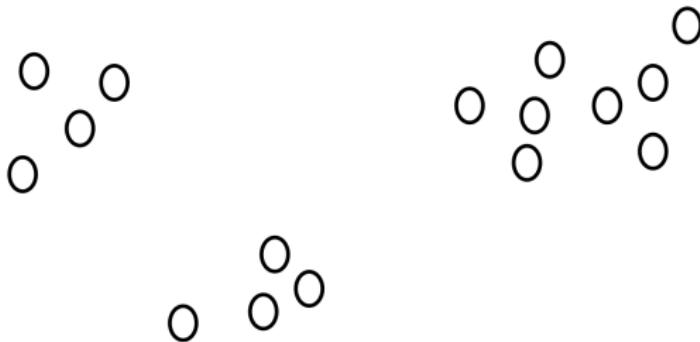


... then we might just need 3 labeled points

- Issues

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

If the data looked like this ...

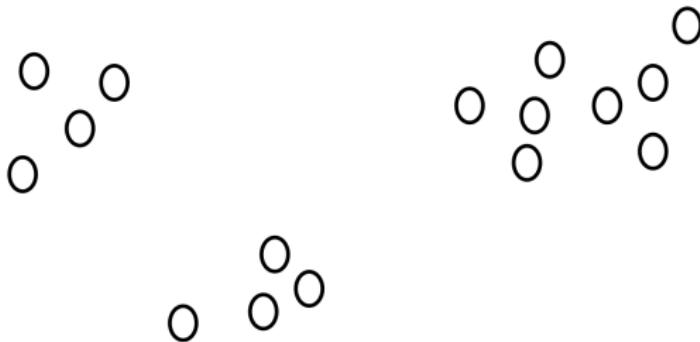


... then we might just need 3 labeled points

- Issues

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

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

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m+1, \dots, m+k$
 - a. Predict y_i using h
 - b. Get true label y_i
 - c. Add (\mathbf{x}_i, y_i) to S_m to get S_{m+1}

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m+1, \dots, m+k$
 - a. Predict y_i using h
 - b. Get true label y_i
 - c. Add (\mathbf{x}_i, y_i) to the sample

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m+1, \dots, m+k$
 - a. Predict y_i using h
 - b. Get true label y_i
 - c. Add (\mathbf{x}_i, y_i) to S_m to get S_{m+1}

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m + 1, \dots, m + k$
 - select an object \mathbf{x}_i
 - get a label y_i
 - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m + 1, \dots, m + k$
 - select an object \mathbf{x}_i
 - get a label y_i
 - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m + 1, \dots, m + k$
 - select an object \mathbf{x}_i
 - get a label y_i
 - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m + 1, \dots, m + k$
 - select an object \mathbf{x}_i
 - get a label y_i
 - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m + 1, \dots, m + k$
 - select an object \mathbf{x}_i
 - get a label y_i
 - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$

- Problem: train hypothesis $h : \mathbb{X} \rightarrow Y$ using a sample $S = \{\mathbf{x}_i, y_i\}_{i=1,2,\dots}$ 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 $i = m + 1, \dots, m + k$
 - select an object \mathbf{x}_i
 - get a label y_i
 - re-train model h using the sample $S_{i-1} \cup (\mathbf{x}_i, y_i)$

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

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

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

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

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

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

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

- 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

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

- 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

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

- Least confidence principle

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

- Margin sampling principle

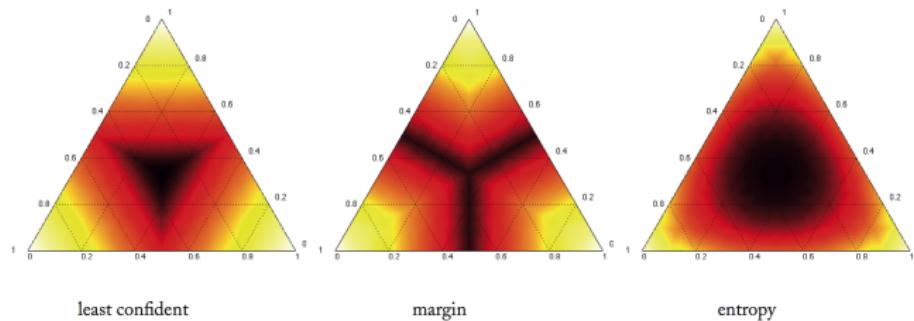
$$\mathbf{x}_i = \arg \min_{\mathbf{x} \in \mathbb{X}} (p_1(\mathbf{x}) - p_2(\mathbf{x}))$$

- Maximum entropy principle

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

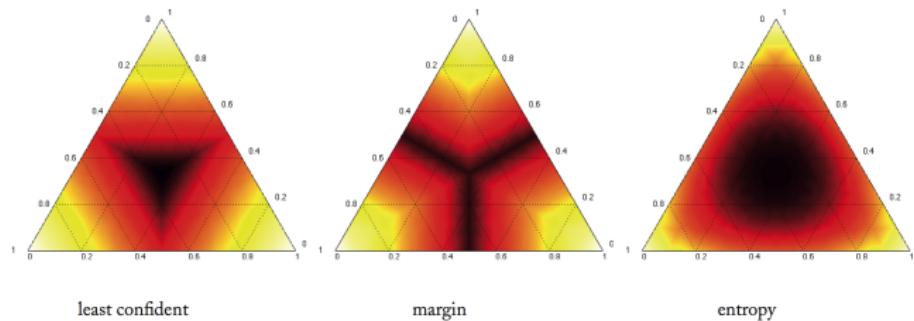
Uncertainty sampling

- 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



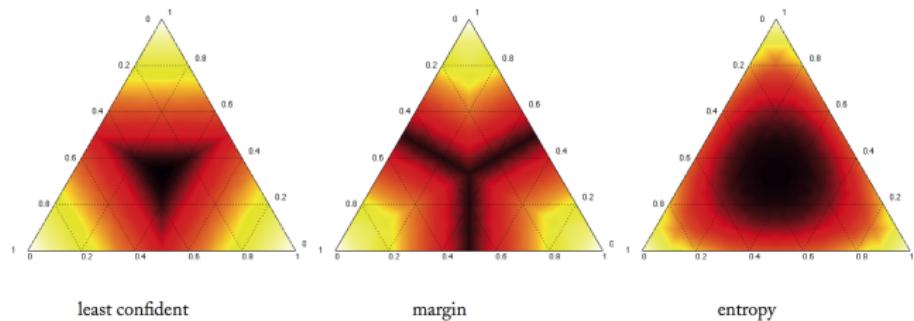
Uncertainty sampling

- 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



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



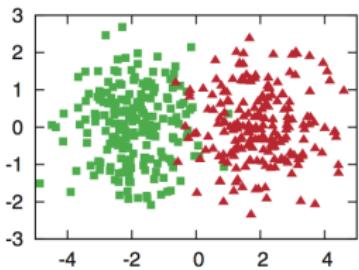
- Example. Synthetic dataset: $m = 30$, $m + k = 400$
 - two Gaussian density
 - logistic regression is constructed using 30 randomly selected objects
 - logistic regression is constructed using 30, adaptively selected using active learning

- 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

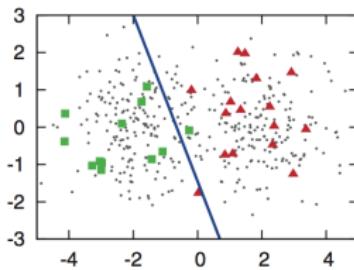
- Example. Synthetic dataset: $m = 30$, $m + k = 400$
 - two Gaussian density
 - logistic regression is constructed using 30 randomly selected objects
 - logistic regression is constructed using 30, adaptively selected using active learning

Active vs. Passive learning

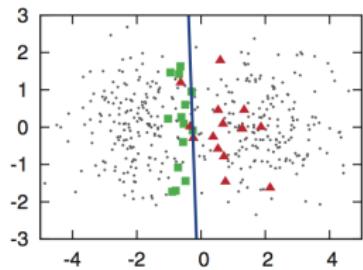
- Example. Synthetic dataset: $m = 30$, $m + k = 400$
 - two Gaussian density
 - logistic regression is constructed using 30 randomly selected objects
 - logistic regression is constructed using 30, adaptively selected using active learning



(a) a 2D toy data set

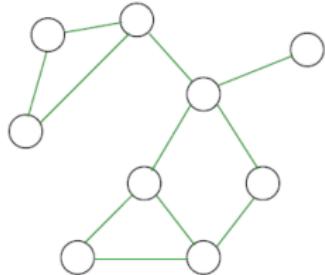


(b) random sampling

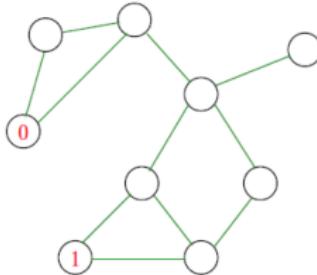


(c) uncertainty sampling

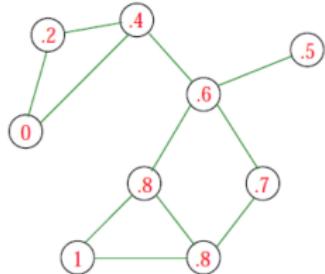
(1) Build neighborhood graph



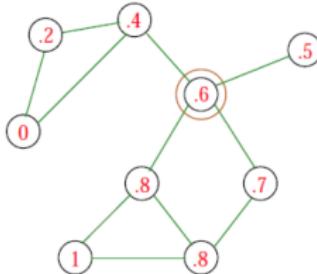
(2) Query some random points



(3) Propagate labels



(4) Make query and go to (3)



- Select \mathbf{x}_i having the highest disagreement between decisions of a committee of models $h_t(\mathbf{x}_i) = \arg \max_{y \in Y} P_t(y|\mathbf{x})$, $t = 1, \dots, T$
- 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 $\widehat{p}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{h_t(\mathbf{z})=y}$

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

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

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

- Select \mathbf{x}_i having the highest disagreement between decisions of a committee of models $h_t(\mathbf{x}_i) = \arg \max_{y \in Y} P_t(y|\mathbf{x})$, $t = 1, \dots, T$
- 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 $\hat{p}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{h_t(\mathbf{z})=y}$

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

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

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

- Select \mathbf{x}_i having the highest disagreement between decisions of a committee of models $h_t(\mathbf{x}_i) = \arg \max_{y \in Y} P_t(y|\mathbf{x})$, $t = 1, \dots, T$
- 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 $\widehat{p}(y|\mathbf{z}) = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{h_t(\mathbf{z})=y}$

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

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

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

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

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

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

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

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

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

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

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

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

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

- 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)

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

- Maximum Reliability of non-labeled data principle

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

- Minimum Entropy of non-labeled data principle

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

- 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)

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

- Maximum Reliability of non-labeled data principle

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

- Minimum Entropy of non-labeled data principle

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

- 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)

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

- Maximum Reliability of non-labeled data principle

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

- Minimum Entropy of non-labeled data principle

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

- 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

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

- 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

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

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

- 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

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

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

- 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

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

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

- 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

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

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

Example of Density Weighting

