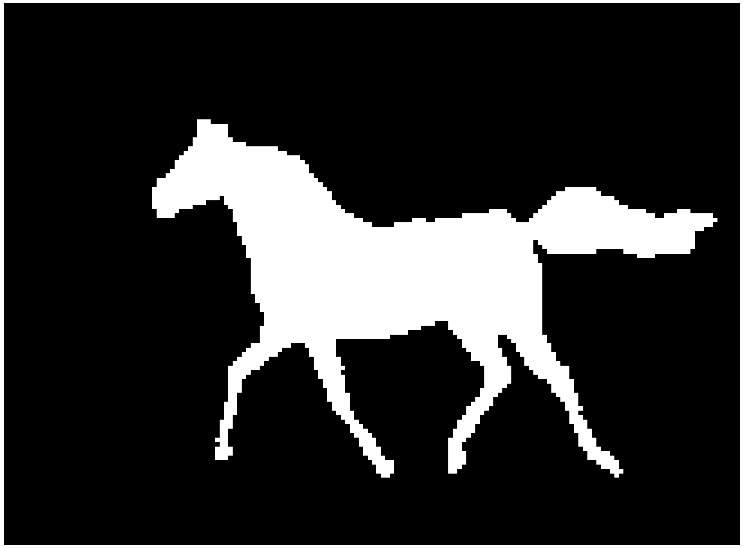
Active Learning

Nicholas Ruozzi
University of Texas at Dallas

Supervised Learning

- We're given lots and lots of labelled examples
  - Goal is to predict the label of unseen examples
  - Observations:
    - We don't necessarily need that many data points to construct a good classifier (think SVMs)
    - In certain applications, labels are ***expensive***
      - They can cost time, money, or other resources

Image Segmentation



Someone had to produce these labels by hand!

Expensive Data

- In general, data is easy to come by but labels are expensive
  - Labelled speech
  - Labelled images and video
  - Large corpora of texts
- These tasks are mind numbing and boring
  - Can pay people to do them! (Amazon Mechanical Turk)
  - Can get expensive fast and we need some way to ensure that they are accurately solving the problem or else we are wasting money!

Semi-supervised Learning

- Given a collection of labeled and unlabeled data, use it to build a model to predict the labels of unseen data points
  - We never get to see the labels of the unlabeled data
  - However, if we assume something about the data generating process, the unlabeled data can still be useful...
    - Could find the model that maximizes the probability of both the labeled and unlabeled data (another application of EM!)

Active Learning

- Given lots of unlabeled examples
  - Learn to predict the label of unseen data points
  - The added feature: we have the ability to ask for the label of any one of the unlabeled inputs (e.g., a labeling oracle/expert)
    - Treat asking the oracle for a label as an expensive operation
    - The performance of the algorithm will be judged by how few queries it can make to learn a good classifier

Related to Experimental Design

- Suppose that we want to determine what disease a patient has
  - We can run a series of (possibly expensive) tests in order to determine the correct diagnosis
  - How should we choose the tests so as to minimize cost (dollars and life) while still guaranteeing that we come up with the correct diagnosis?

A First Attempt

- Could just randomly pick an unlabeled data point
  - Request its label
  - Add it to the training data
  - Retrain the model
  - Repeat
- If labels are expensive, can be a terrible idea
  - Many unlabeled data points may have very little impact on the predicted labels
  - This is effectively the supervised setting

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

A Motivating Example

- Binary classification via linear separators
- Suppose we are given a collection of unlabeled data points in one dimension
- Assuming that the data is separable (and noise free), how many queries to the labeling oracle do we need to find a separator?

Ideal case: number of hypotheses consistent with the labeling is approximately halved at each step

Types of Active Learning

- Pool based
  - We're given all of the unlabeled data upfront
- Streaming
  - Unlabeled examples come in one at a time and we have to decide whether or not we want to label them as they arrive
  - Also applies to situations in which storing the all data is not possible

Basic Strategy

- Iteratively build a model
- Use the current model to find “informative” unlabeled examples
- Select the most informative example(s)
  - Label them and add them to the training data
- Retrain the model using the new training data
- Repeat

Basic Strategy

- Iteratively build a model
- Use the current model to find "informative" unlabeled examples
- Select the most informative example(s)
    - Label them and add them to the training data
- Retrain the model using the new training data
- Repeat

Note: this procedure will result in a biased sampling of the underlying distribution in general (the actively labeled dataset is not reflective of the underlying data generating process)

Informative Examples

- For learning algorithms that model the data generating process...
  - A data point is informative if the current model is not confident in its prediction for this example
  - Least confident labeling (binary label case):

$$\arg \max_{x \text{ unlabeled}} 1 - \max_y p(y|x, \theta)$$

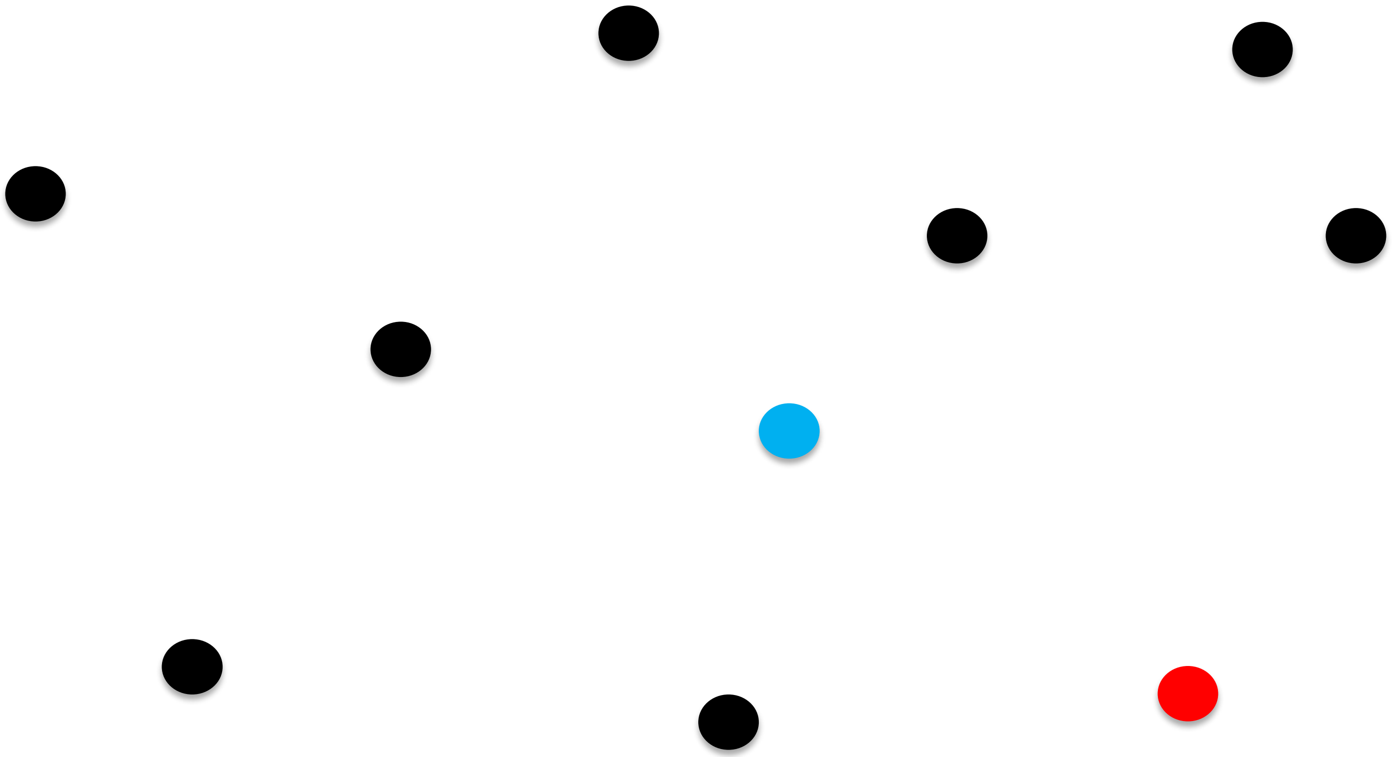
- For learning algorithms, like SVMs, that are simply selecting among a collection of hypotheses...
  - Unlabeled data points that are far from the current decision boundary are unlikely to provide useful information

Query-by-Committee

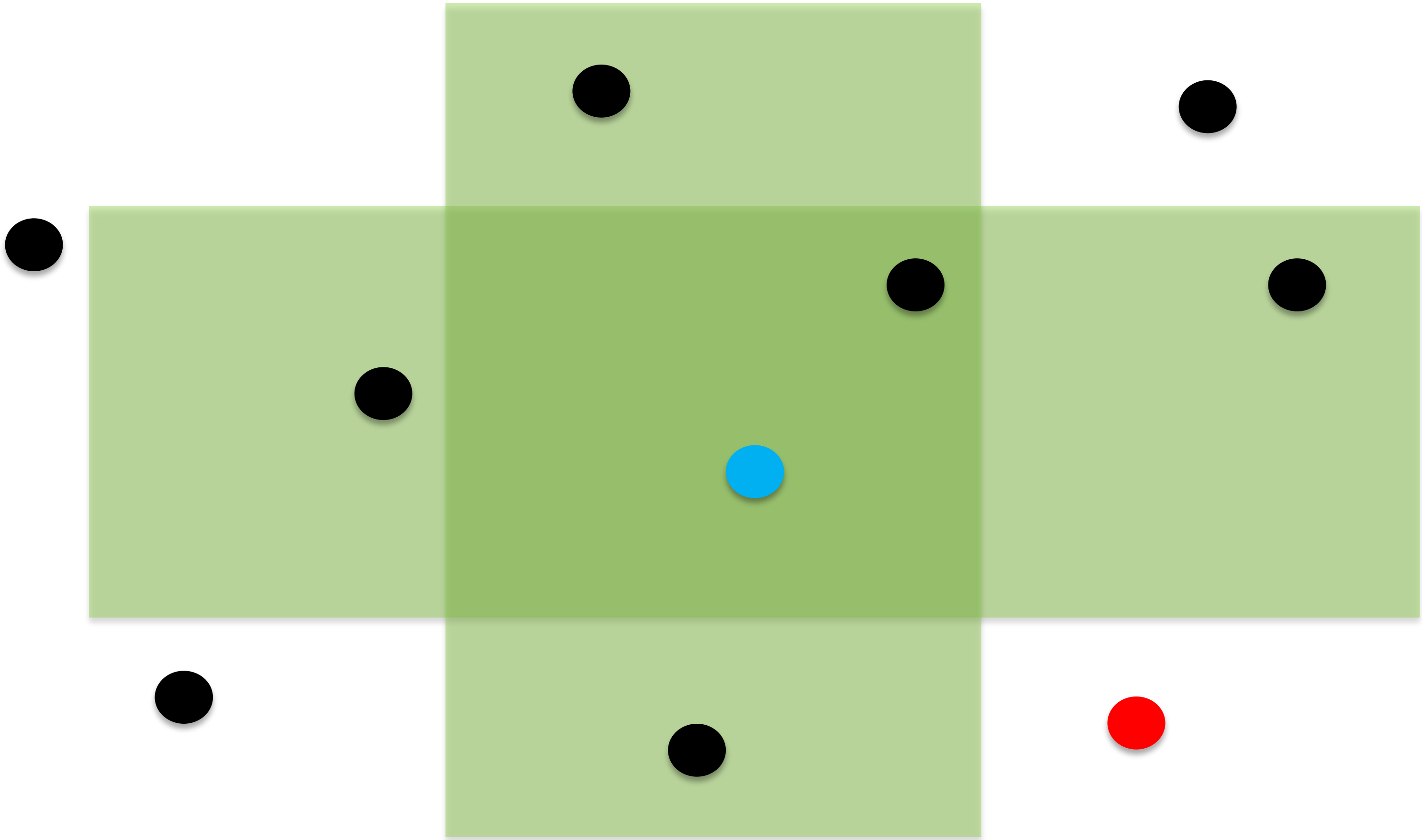
- Select a committee of T consistent classifiers using the labeled data
- Find examples for which the committee has the largest disagreement
  - For example, in a binary labeling problem, find the examples for which the committee's votes are split as close to 50/50 as possible between +1 and -1
- Request the label for these examples

Goal: reduce the version space as much as possible by selecting points whose label will eliminate the most hypotheses

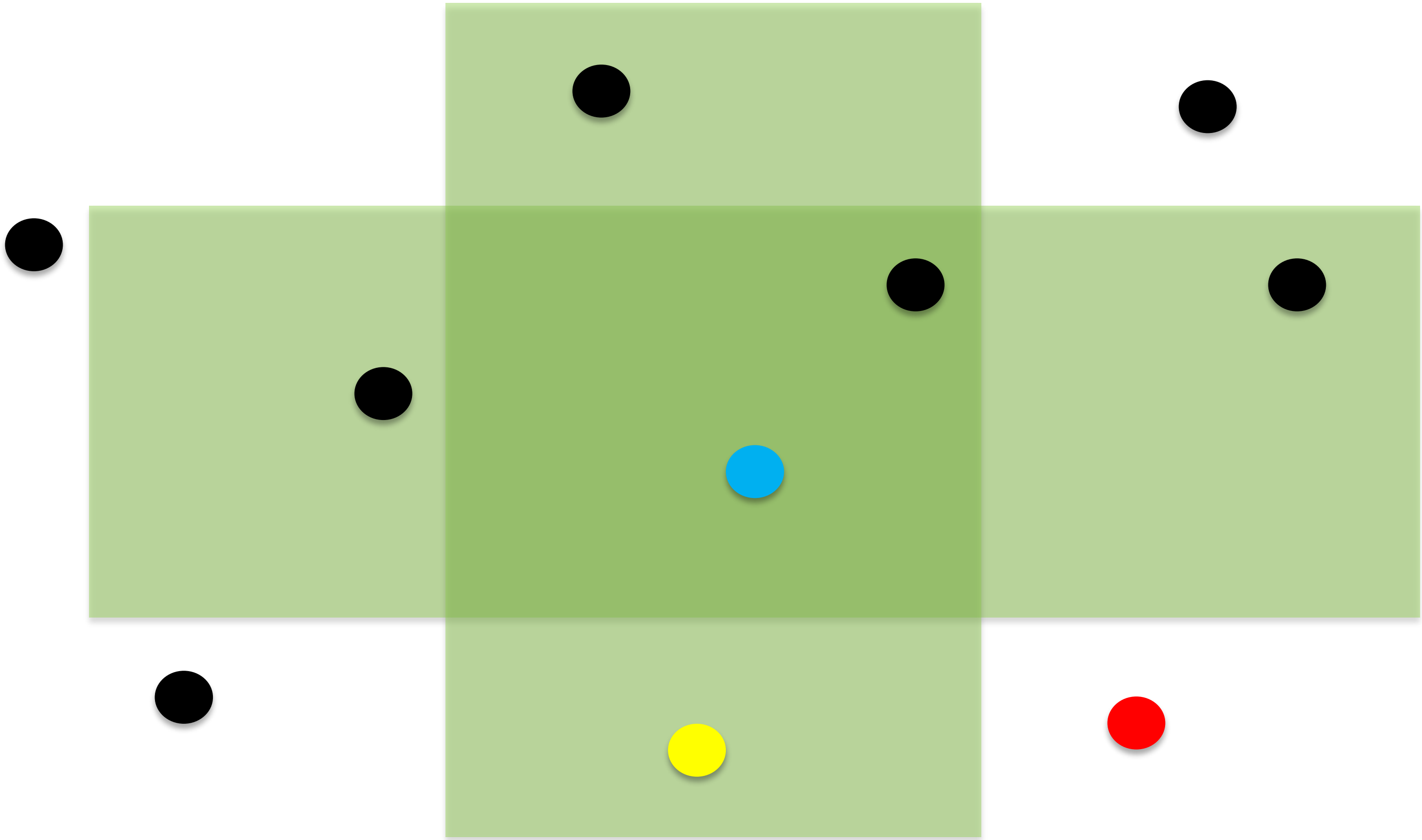
Query-by-Committee



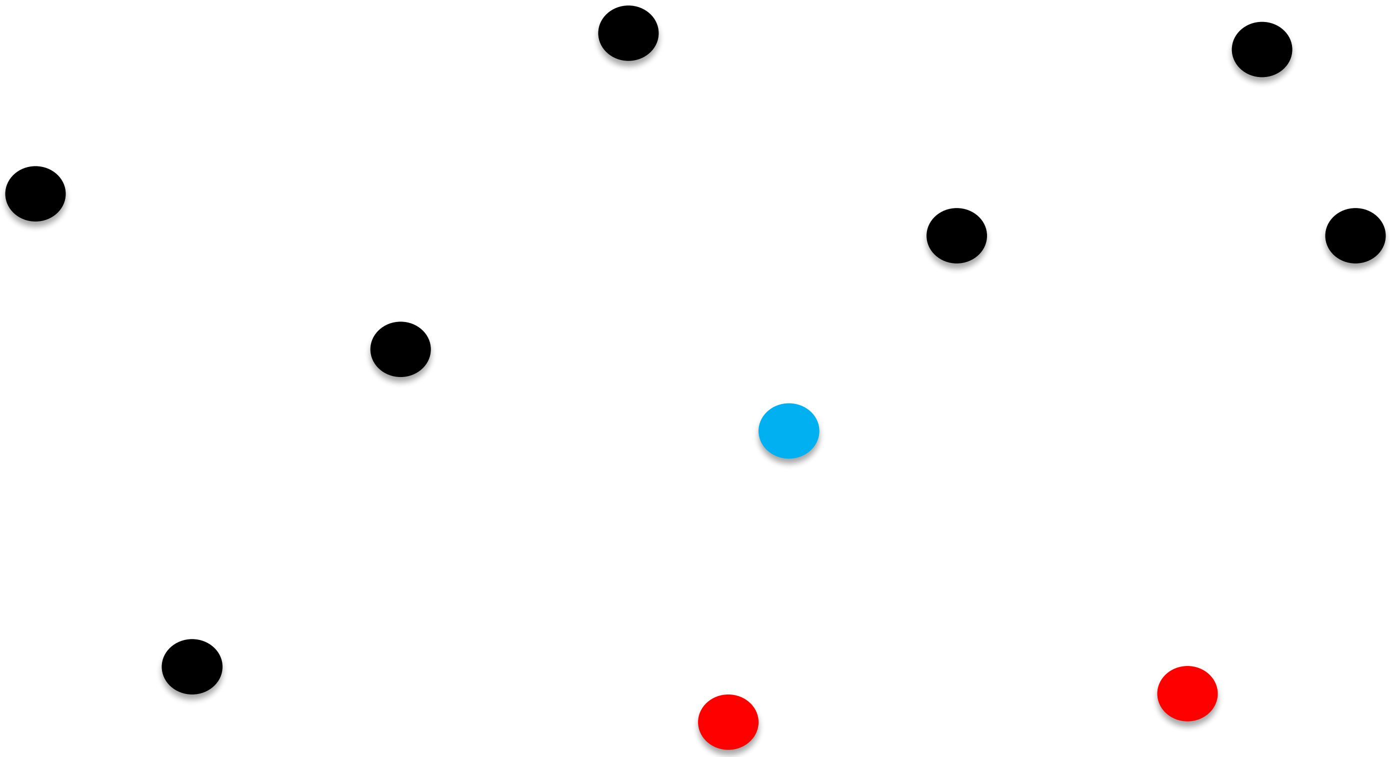
Query-by-Committee



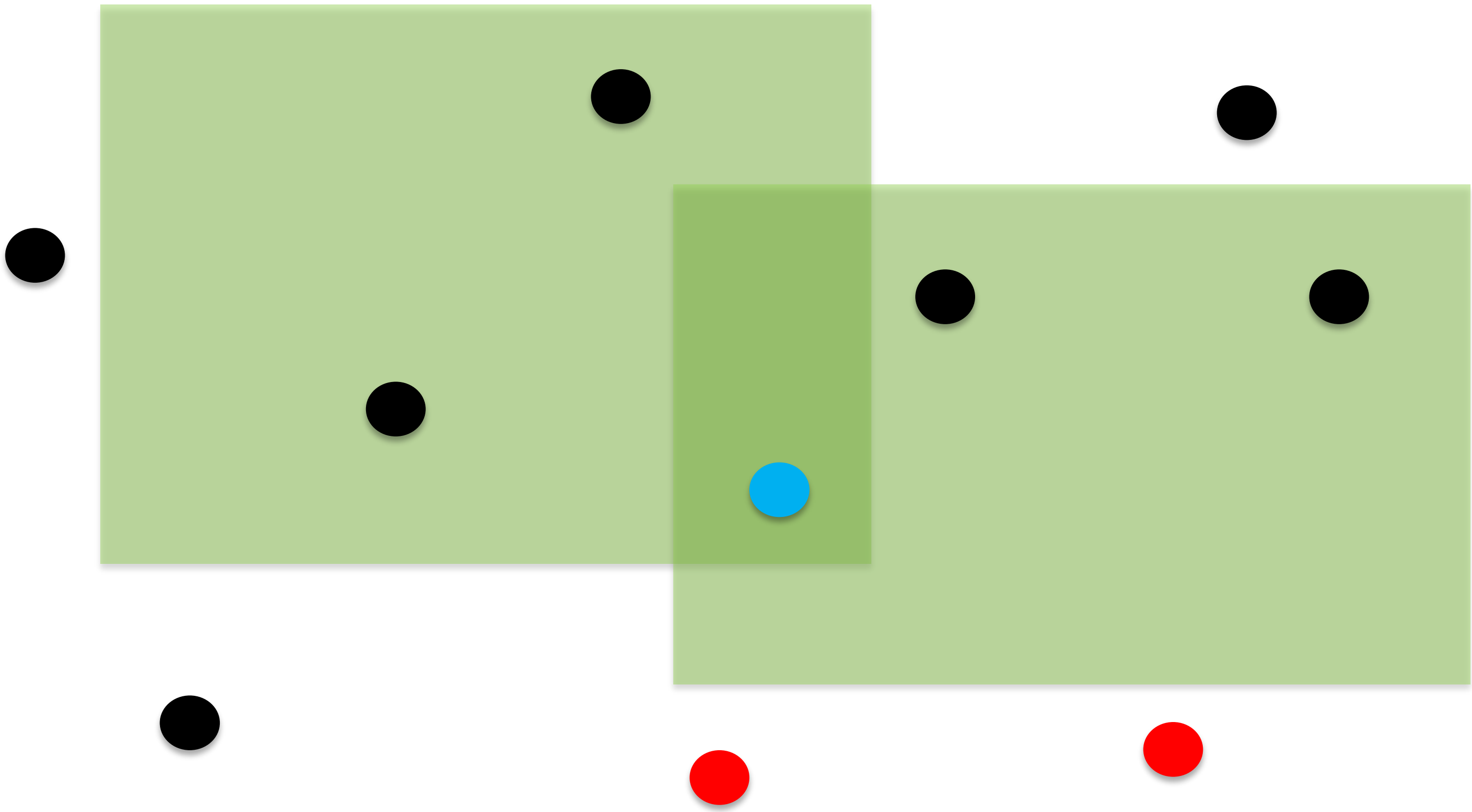
Query-by-Committee



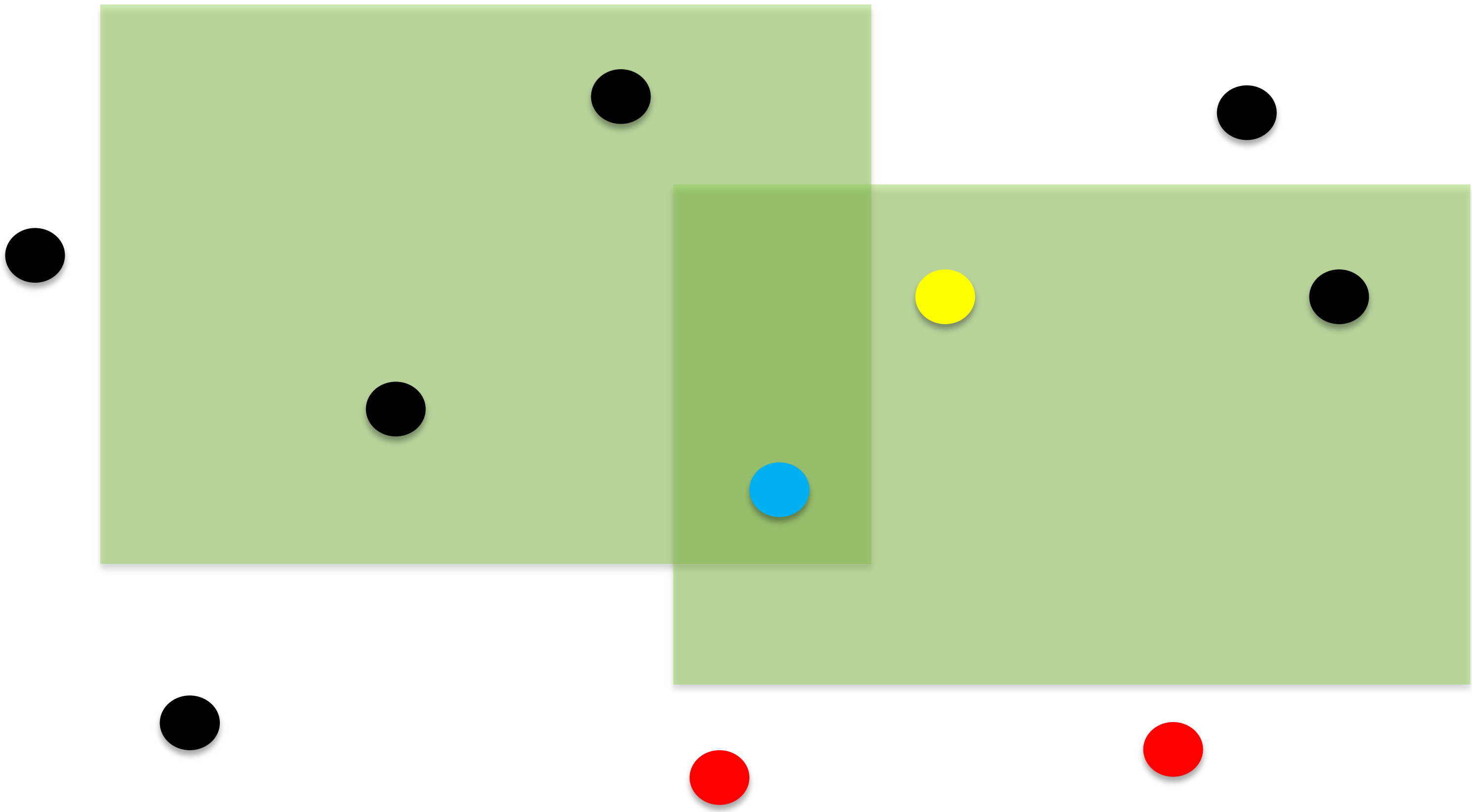
Query-by-Committee



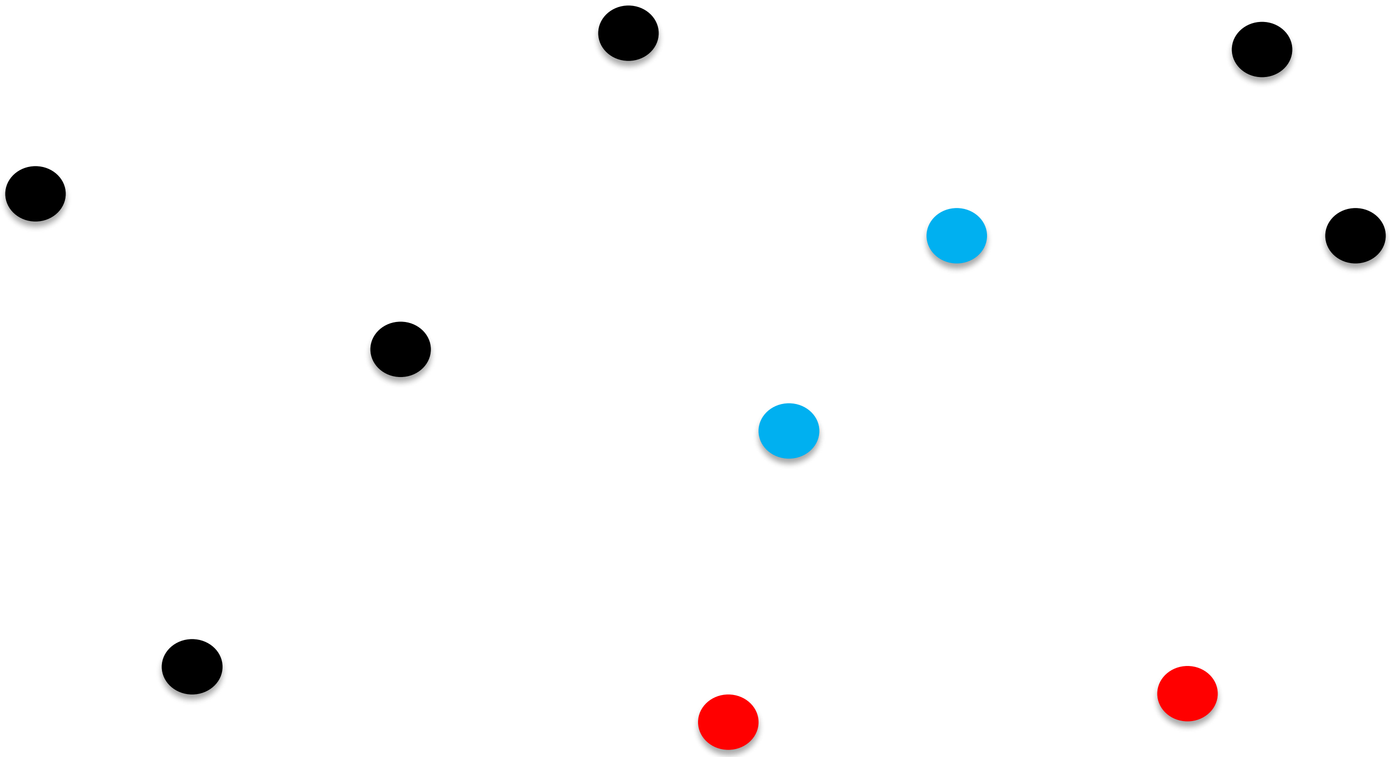
Query-by-Committee



Query-by-Committee



Query-by-Committee



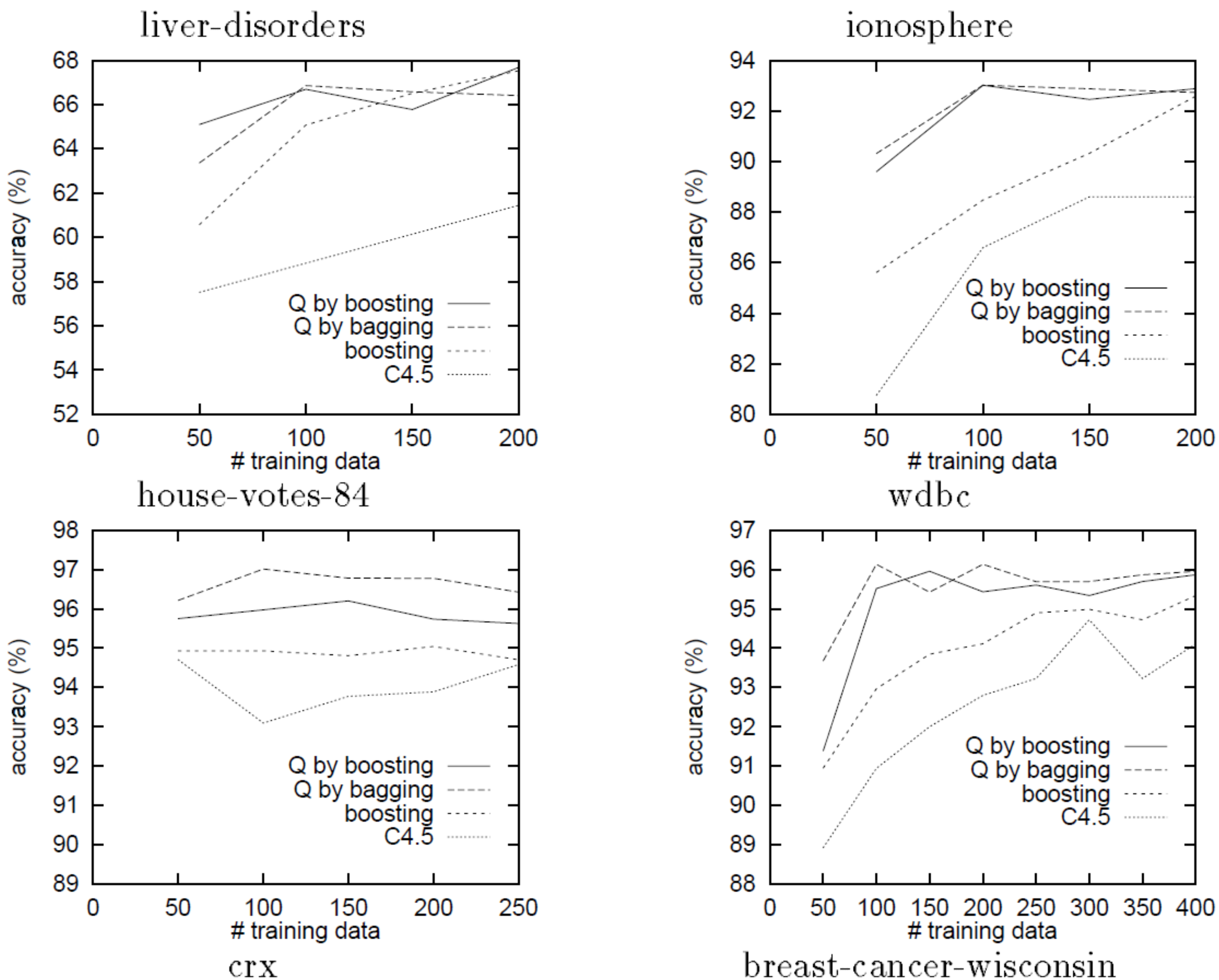
Query-by-Committee

- How to form a committee?
  - Need to pick consistent hypotheses (ideally, we'd consider all possible consistent hypotheses, but that may not be computationally feasible)
  - We could sample hypotheses from the version space with respect to the underlying distribution over hypotheses $p(\theta|labeled\ data)$
    - Difficult/expensive to compute this distribution in practice
  - Other ideas?

Query-by-Bagging

- At each step, generate T samples from the labeled data by resampling as in bagging
  - Train a perfect classifier on each sample
  - The committee is chosen to be these T classifiers
- Perform one iteration of the query-by-committee scheme using the above selected committee
- Can also do query-by-boosting! (same basic idea)
  - Run AdaBoost for T iterations to build a classifier
  - The AdaBoost classifier already contains the weighted vote of the committee

Experimental Comparison



Abe & Mamitsuka, ICML '98

Outliers

- A data point may have an uncertain/controversial label simply because it is an outlier
  - Such data points are unlikely to help the learner and could even hurt performance
  - Some methods to help correct for this (density weighting, etc.)

Other Query Selection Heuristics

- Many other heuristics to select informative data points
  - Select examples whose inclusion results in the most significant change in the model
  - Select examples that reduce the expected generalization error the most over unlabeled examples (labeled using the model)
  - Select examples that reduces the model variance the most

Mellow Learners

- Consider the streaming setting
- Let H_1 be the hypothesis class
- At step t,
  - Receive unlabeled point $x^{(t)}$
  - If there is any disagreement within H_t about x_t's label, query label $y^{(t)}$ and set $H_{t+1} = \{h \in H_t : h(x^{(t)}) = y^{(t)}\}$ else $H_{t+1} = H_t$

Mellow Learners

- Consider the streaming setting
- Let H_1 be the hypothesis class
- At step t,
  - Receive unlabeled point $x^{(t)}$
  - If there is any disagreement within H_t about x_t's label, query label $y^{(t)}$ and set $H_{t+1} = \{h \in H_t : h(x^{(t)}) = y^{(t)}\}$ else $H_{t+1} = H_t$

Can be intractable to compute and store H_t's

Mellow Learners

- Consider the streaming setting
- Let H_1 be the hypothesis class
- At step t,
  - Receive unlabeled point $x^{(t)}$
  - If there is any disagreement within H_t about x_t's label, query label $y^{(t)}$ and set $H_{t+1} = \{h \in H_t : h(x^{(t)}) = y^{(t)}\}$ else $H_{t+1} = H_t$

Results, roughly, in an exponential decrease in size of hypothesis space for data points with strong disagreement

Challenges

- Is it always possible to find queries that will effectively cut the size of the set of consistent hypotheses (a.k.a. the version space) in half?
  - If so, how can we find them?
  - Can we construct approaches that come with rigorous guarantees (e.g., the PAC learning for the active learning setting)?
  - How to handle noisy labels?

Supervised Learning

- Regression & classification
- Discriminative methods
  - k-NN
  - Decision trees
  - Perceptron
  - SVMs & kernel methods
  - Logistic regression
- Parameter learning
  - Maximum likelihood estimation
  - Expectation maximization
- Active learning

Bayesian Approaches

- MAP estimation
- Prior/posterior probabilities
- Bayesian networks
  - Naive Bayes
  - Hidden Markov models
  - Structure learning via Chow-Liu Trees

Unsupervised Learning

- Clustering
  - k-means
  - Hierarchical clustering
- Expectation maximization
  - Soft clustering
  - Mixtures of Gaussians

Learning Theory

- PAC learning
- VC dimension
- Bias/variance tradeoff
- Chernoff bounds
- Sample complexity

Optimization Methods

- Gradient descent
  - Stochastic gradient descent
  - Subgradient methods
- Coordinate descent
- Lagrange multipliers and duality

Matrix Based Methods

- Dimensionality Reduction
  - PCA
  - Matrix Factorizations
- Collaborative Filtering
  - Semisupervised learning

Ensemble Methods

- Bootstrap sampling
- Bagging
- Boosting

Other Learning Topics

- Active learning
- Reinforcement learning
- Neural networks
  - Perceptron and sigmoid neurons
  - Backpropagation

Questions about the course content?

(Reminder: I do not have office hours this week)

For the final...

- You should understand the basic concepts and theory of all of the algorithms and techniques that we have discussed in the course
- There is no need to memorize complicated formulas, etc.
  - For example, if I ask for the sample complexity of a scheme, I will give you the generic formula
- However, you should be able to derive the algorithms and updates
  - e.g., Lagrange multipliers and SVMs, the EM algorithm, etc.

For the final...

- No calculators, books, notes, etc. will be permitted
  - As before, if you need a calculator, you have done something terribly wrong
- The exam will be in roughly the same format
  - Expect true/false questions, short answers, and two-three long answer questions
- Exam will emphasize the new material, but ALL material will be tested
- Take a look at the practice exam!

Final Exam

Wednesday, 12/13/2017

11:00AM - 1:45PM

ECSS 2.306

Related Courses at UTD

- Natural Language Processing (CS 6320)
- Statistical Methods in Artificial Intelligence and Machine Learning (CS 6347)
- Artificial Intelligence (CS 6364)
- Information Retrieval (CS 6322)
- Intelligent Systems Analysis (ACN 6347)
- Intelligent Systems Design (ACN 6349)

ML Related People

- Vincent Ng (NLP)
- Vibhav Gogate (MLNs, Sampling, Graphical Models)
- Sanda Harabagiu (NLP & Health)
- Dan Moldovan (NLP)
- Sriraam Natarajan (MLNs, Graphical Models)
- Nicholas Ruozzi (Graphical Models & Approx. Inference)