Data Classification: Advanced Concepts

Lijun Zhang
zlj@nju.edu.cn
http://cs.nju.edu.cn/zlj

Outline

- **Introduction**
- Multiclass Learning
- Rare Class Learning
- Scalable Classification
- Semisupervised Learning
- Active Learning
- Ensemble Methods
- Summary

Introduction

- Difficult Classification Scenarios
  - Multiclass learning
  - Rare class learning
  - Scalable learning
  - *Numeric class variables*

- Enhancing Classification
  - Semisupervised learning
  - Active learning
  - Ensemble learning

Outline

- Introduction
- **Multiclass Learning**
- Rare Class Learning
- Scalable Classification
- Semisupervised Learning
- Active Learning
- Ensemble Methods
- Summary

Multiclass Learning

- Many classifiers can be directly used for multiclass learning
  - Decision trees, Bayesian methods, Rule-based classifiers
- Many classifiers can be generalized to multiclass case
  - Support vector machines (SVMs), Neural networks, Logistic regression
- Generic meta-frameworks
  - Directly use the binary methods for multiclass classification

One-against-rest Approach

- k different binary classification problems are created
  - In the ith problem, the ith class is considered the set of positive examples, whereas all the rest are negative
- k models are applied during testing
  - If the positive class is predicted in the ith problem, then the ith class is rewarded with a vote
  - Otherwise, each of the remaining classes is rewarded with a vote
  - Weighted vote is also possible

One-against-one Approach

- A training data set is constructed for each of the $\binom{k}{2}$ pairs of classes
  - Results in $k(k-1)/2$ models
- $k(k-1)/2$ models are applied during testing
  - For each model, the prediction provides a vote to the winner
  - Weighted vote is also possible
- For each model, the size of training data is small ($2/k$ of the original one)

Outline

- Introduction
- Multiclass Learning
- **Rare Class Learning**
- Scalable Classification
- Semisupervised Learning
- Active Learning
- Ensemble Methods
- Summary

Rare Class Learning

- The class distribution is unbalanced
  - Credit card activity: 99% of data are normal and 1% of data are fraudulent
  - Given a test instance $\bar{X}$, whose nearest 100 neighbors contain 49 rare class instances and 51 normal class instances
    - ✓ k-nearest neighbor with $k = 100$ will output normal
    - ✓ However, it is surrounded by large fraction of rare instances relative to expectation
  - Outputting normal achieves 99% accuracy

The General Idea

- Achieving a high accuracy on the rare class is more important
  - The cost of misclassifying a rare class are much higher than those of misclassifying the normal class
- Cost-weighted Accuracy
  - A misclassification cost $C(i)$ is associated with class i
- Two Approaches
  - Example reweighting
  - Example resampling

Example Reweighting (1)

- All instances belonging to the ith class are weighted by $C(i)$
- Existing methods need to be modified
  - Decision trees
    - ✓ Gini index and entropy
  - Rule-based classifiers
    - ✓ Laplacian measure and information gain
  - Bayes classifiers
    - ✓ Class priors and conditional probabilities
  - Instance-based methods
    - ✓ Weighted votes

Example Reweighting (2)

- Support vector machines

$$\min_{\bar{W}\in\mathbb{R}^d,\xi_1,\ldots,\xi_n,b\in\mathbb{R}} \quad O = \frac{\|\bar{W}\|^2}{2} + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.} \quad \overline{W}\cdot\overline{X_i} + b \geq +1 - \xi_i \quad \forall i : y_i = +1$$
$$\overline{W}\cdot\overline{X_i} + b \leq -1 + \xi_i \quad \forall i : y_i = -1$$
$$\xi_i \geq 0 \quad \forall i.$$

$$\min_{\bar{W}\in\mathbb{R}^d,\xi_1,\ldots,\xi_n,b\in\mathbb{R}} \quad O = \frac{\|\bar{W}\|^2}{2} + C\sum_{i=1}^{n}C(y_i)\xi_i$$

$$\text{s.t.} \quad \overline{W}\cdot\overline{X_i} + b \geq +1 - \xi_i \quad \forall i : y_i = +1$$
$$\overline{W}\cdot\overline{X_i} + b \leq -1 + \xi_i \quad \forall i : y_i = -1$$
$$\xi_i \geq 0 \quad \forall i.$$

Sampling Methods

- Different classes are differentially sampled to enhance the rare class
  - The sampling probabilities are typically chosen in proportion to their misclassification costs
  - The rare class can be oversampled
  - The normal class can be undersampled
- One-sided selection
  - All instances of the rare class are used
  - A small sample of the normal class are used
  - Both efficient and effective

Synthetic Oversampling: SMOTE

- Oversampling the rare class
  - Repeated samples of the same data point
  - Repeated samples cause overfitting
- The SMOTE approach
  - For each rear instance, its k nearest neighbors belonging to the same class are determined
  - A fraction of them are chosen randomly
  - For each sampled example-neighbor pair, a synthetic data example is generated on the line segment connecting them

Outline

- Introduction
- Multiclass Learning
- Rare Class Learning
- **Scalable Classification**
- Semisupervised Learning
- Active Learning
- Ensemble Methods
- Summary

Scalable Classification

- Data cannot be loaded in memory
  - Traditional algorithms are not optimized to disk-resident data
- One solution—sampling the data
  - Lose knowledge in the discarded data
- Some classifiers can be made faster by using more efficient subroutines
  - Associative classifiers: frequent pattern mining
  - Nearest-neighbor methods: nearest-neighbor indexing

Scalable Decision Trees (1)

RainForest

- The evaluation of the split criteria in univariate decision trees do not need access to the data in its multidimensional form

- Only the count statistics of distinct attributes values need to be maintained over different classes
  - ✓ AVC-set at each node: counts of the distinct values of the attribute for different classes
  - ✓ Depends only on the number of features, number of distinct attribute values and the number of classes

Scalable Decision Trees (2)

- Bootstrapped Optimistic Algorithm for Tree construction (BOAT)
  - In bootstrapping, the data is sampled with replacement to create b different bootstrapped samples
  - These are used to create b different trees
  - BOAT uses them to create a new tree that is very close to the one constructed from all the data
  - It requires only two scans over the database

Scalable Support Vector Machines (1)

- Dual of Kernel SVM

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(\overline{X_i}, \overline{X_j})$$

  - n variables, and $O(n^2)$ memory
- The SVMLight approach
  - It is not necessary to solve the entire problem at one time.
  - The support vectors for the SVMs correspond to only a small number of training data points

Scalable Support Vector Machines (2)

Dual of Kernel SVM

$$L_D = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(\overline{X_i}, \overline{X_j})$$

- n variables, and $O(n^2)$ memory

The SVMLight approach

- Select q variables as the active working set S_q, and solve $L_D(S_q)$
  - ✓ Select a working set that leads to the maximum improvement in the objective
- Shrinking the training data during the optimization process

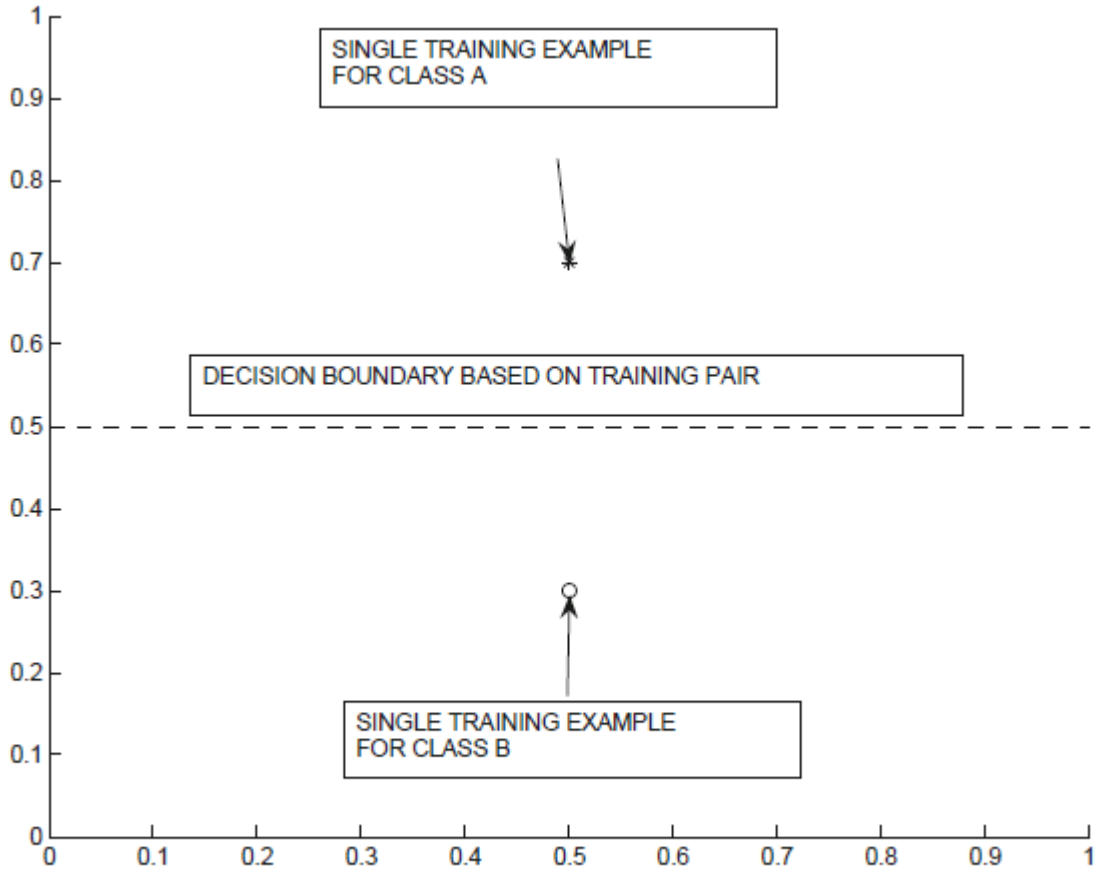
Outline

- Introduction
- Multiclass Learning
- Rare Class Learning
- Scalable Classification
- **Semisupervised Learning**
- Active Learning
- Ensemble Methods
- Summary

Semisupervised Learning

- Labeled data is expensive and hard to acquire
- Unlabeled data is often copiously available
- Unlabeled data is useful
  - Unlabeled data can be used to estimate the low-dimensional manifold structure of the data
  - Unlabeled data can be used to estimate the joint probability distribution of features

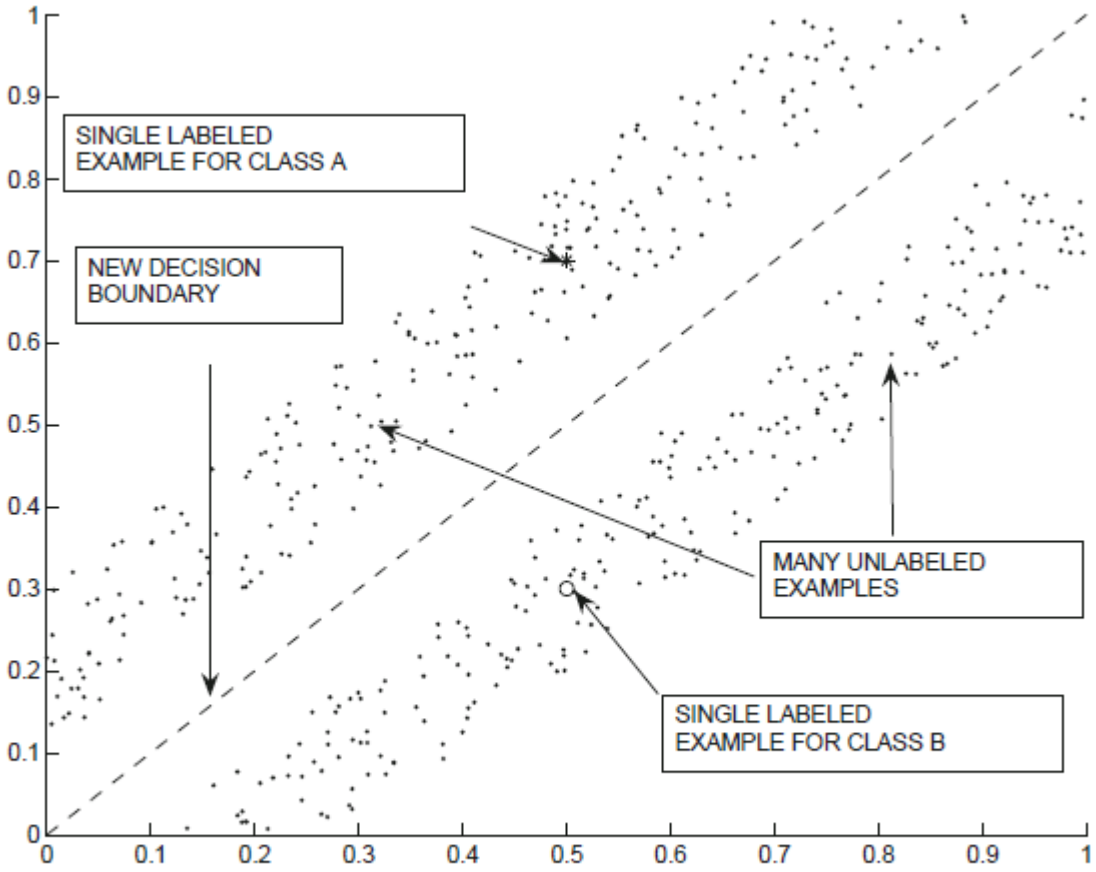
The 1st Example



(a) Labeled data

The 1st Example

Class variables are likely to vary smoothly over dense regions



(b) Labeled and unlabeled data

The 2nd Example

- The goal is to determine whether documents belong to the “Science” category.
- In labeled data, we found the word “Physics” is associated with the “Science” category
- In unlabeled data, we found the word “Einstein” often co-occur with “Physics”
- Thus, the unlabeled documents provide the insight that the word “Einstein” is also relevant to the “Science” category

Techniques for Semisupervised Learning

- **Meta-algorithms** that can use any existing classification algorithm as a subroutine
  - Self-Training
  - Co-training
- Specific Algorithms
  - Semisupervised Bayes classifiers
  - Transductive support vector machines
  - Graph-Based Semisupervised Learning

Self-training

The Procedure

- Initial labeled set L and unlabeled set U

1. Use algorithm $\mathcal{A}$ on the current labeled set L to identify the k instances in the unlabeled data U for which the classifier $\mathcal{A}$ is the most confident
2. Assign labels to the k most confidently predicted instances and add them to L. Remove these instances from U

Overfitting

- Addition of predicted labels may propagate errors

Co-training

The Procedure

- Two disjoint feature groups: F_1 and F_2
- Labeled sets L_1 and L_2

1. Train classifier $\mathcal{A}_1$ using labeled set L_1 and feature set F_1, and add k most confidently predicted instances from unlabeled set $U - L_2$ to training data set L_2 for classifier $\mathcal{A}_2$

2. Train classifier $\mathcal{A}_2$ using labeled set L_2 and feature set F_2, and add k most confidently predicted instances from unlabeled set $U - L_1$ to training data set L_1 for classifier $\mathcal{A}_1$

Techniques for Semisupervised Learning

- Meta-algorithms that can use any existing classification algorithm as a subroutine
  - Self-Training
  - Co-training

- **Specific Algorithms**
  - Semisupervised Bayes classifiers
  - Transductive support vector machines
  - Graph-Based Semisupervised Learning

Naive Bayes (1)

Model for Classification

- The goal is to predict

$$P(C = c|\bar{X} = (a_1 \dots a_d))$$

- Bayes theorem

$$P(C = c|x_1 = a_1, \dots x_d = a_d) = \frac{P(C = c)P(x_1 = a_1, \dots x_d = a_d|C = c)}{P(x_1 = a_1, \dots x_d = a_d)}$$
$$\propto P(C = c)P(x_1 = a_1, \dots x_d = a_d|C = c).$$

- Naive Bayes approximation

$$P(x_1 = a_1, \dots x_d = a_d|C = c) = \prod_{j=1}^{d} P(x_j = a_j|C = c)$$

- Bayes probability

$$P(C = c|x_1 = a_1, \dots x_d = a_d) \propto P(C = c) \prod_{j=1}^{d} P(x_j = a_j|C = c).$$

Naive Bayes (2)

Training

- $P(C = c)$

$$P(C = c) = \frac{r(c)}{n}$$

- $P(x_j = a_j | C = c)$: estimated as the fraction of training examples taking on value a_j, conditional on the fact, that they belong to class c

$$P(x_j = a_j | C = c) = \frac{q(a_j, c)}{r(c)}$$

Semisupervised Bayes Classification with EM

- The Key idea
  - Create semi-supervised clusters from the data, and learn from those clusters
- The Procedure
  - (E-step) Estimate posterior probability

$$P(C = c|\overline{X}) \propto P(C = c) \prod_{j=1}^{d} P(x_j = a_j|C = c)$$

  - (M-step) Estimate conditional probability

$$P(x_j = a_j|C = c) = \frac{\sum_{\overline{X} \in \mathcal{L} \cup \mathcal{U}} w(\overline{X}, c) I(x_j, a_j)}{\sum_{\overline{X} \in \mathcal{L} \cup \mathcal{U}} w(\overline{X}, c)}$$

Transductive support vector machines

Support Vector Machines

$$\min_{\bar{W}\in\mathbb{R}^d,\xi_1,\ldots,\xi_n,b\in\mathbb{R}} \quad O = \frac{\|\bar{W}\|^2}{2} + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.} \quad y_i(\overline{W}\cdot\overline{X_i}+b) \geq 1-\xi_i \quad \forall i.$$

$$\xi_i \geq 0 \quad \forall i.$$

Adding Unlabeled Data

$$z_i(\overline{W}\cdot\overline{X_i}+b) \geq 1-\xi_i \quad \forall i : \overline{X_i} \in \mathcal{U}.$$

$$z_i \in \{-1,+1\}$$

Integer Program

- Can only be solved approximately

Graph-Based Semisupervised Learning

Procedures

1. Construct a similarity graph on both the labeled and the unlabeled data records. Each data object O_i is associated with a node in the similarity graph. Each object is connected to its k-nearest neighbors.

2. The weight w_{ij} of the edge (i, j) is equal to a kernelized function of the distance $d(O_i, O_j)$ between the objects O_i and O_j, so that larger weights indicate greater similarity. A typical example of the weight is based on the *heat kernel* [90]:

$$w_{ij} = e^{-d(O_i,O_j)^2/t^2}. \tag{11.24}$$

Here, t is a user-defined parameter.

Semisupervised Learning over Graph

- Zhou et al. Learning with Local and Global Consistency. In NIPS, 2004.

Discussions

- Should we always use unlabeled data?
  - For semisupervised learning to be effective, the class structure of the data should approximately match its clustering structure
  - In practice, semisupervised learning is most effective when the number of labeled examples is extremely small

Outline

- Introduction
- Multiclass Learning
- Rare Class Learning
- Scalable Classification
- Semisupervised Learning
- **Active Learning**
- Ensemble Methods
- Summary

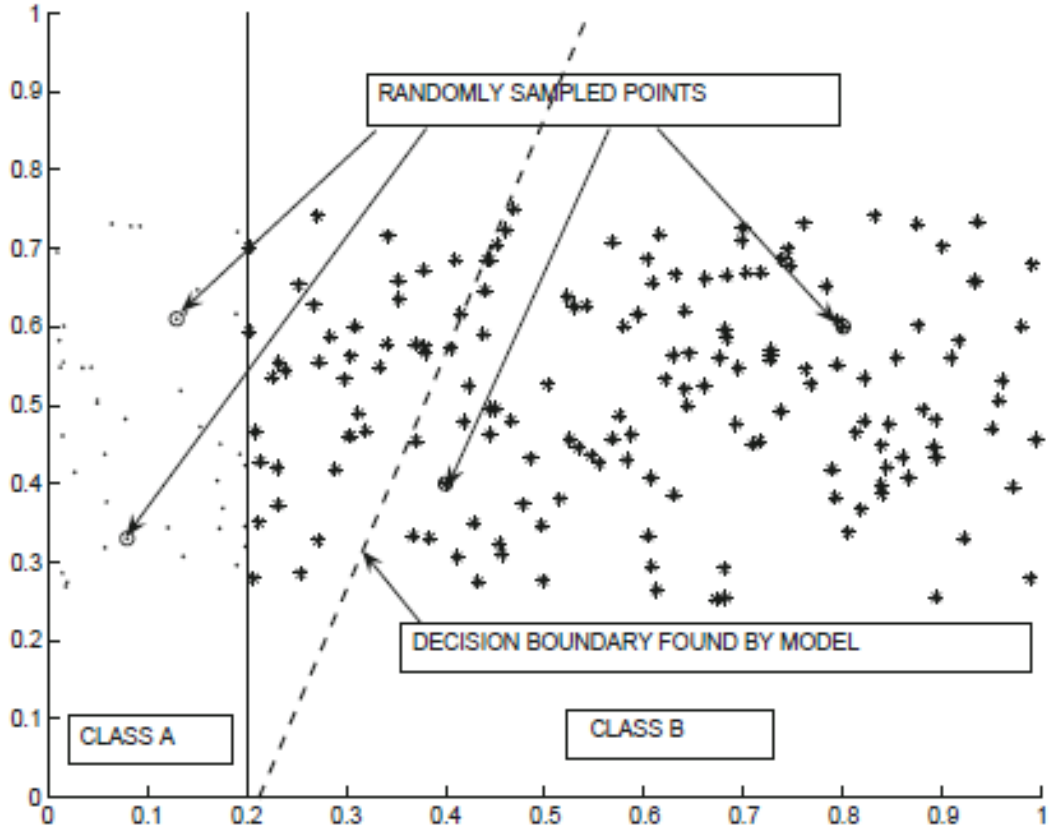
Active Learning

- Labels are Expensive
  - Document collections
  - Privacy-constrained data sets
  - Social networks

- Solutions
  - Utilize the unlabeled data—Semisupervised learning
  - Label the most informative data—Active learning

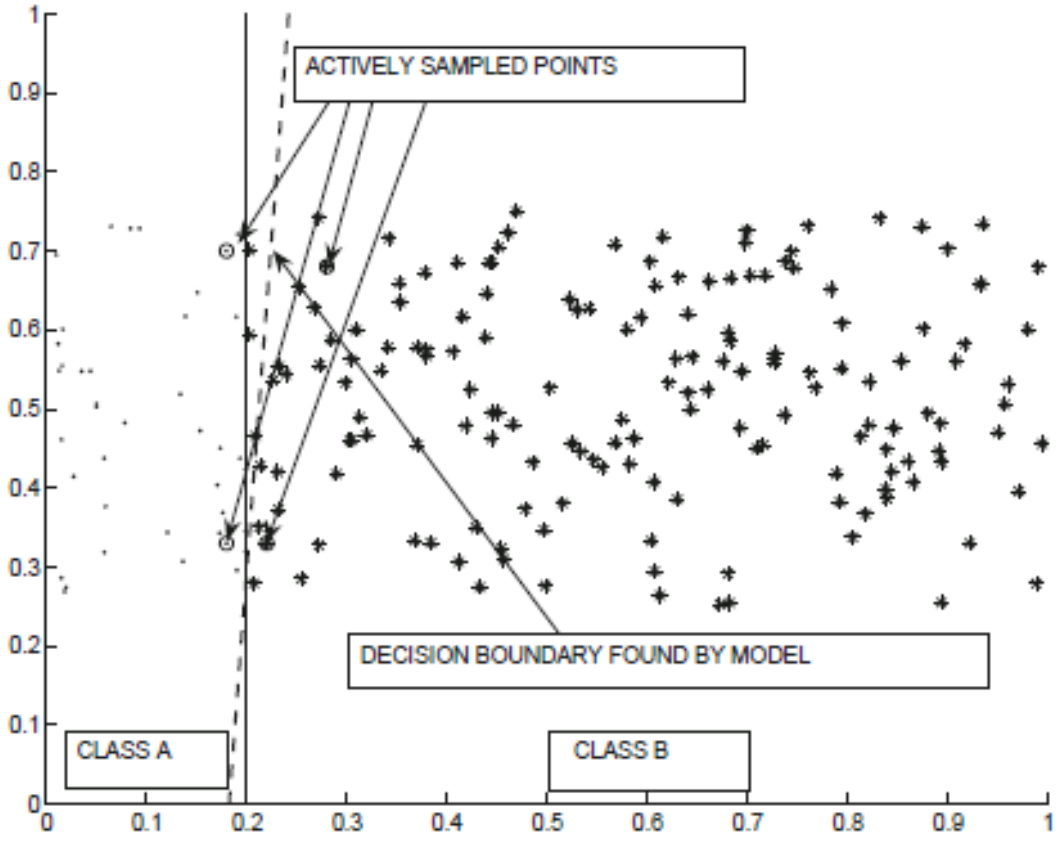
An Example (1)

- Random Sampling



An Example (2)

- Active Sampling



Modeling

- The Key Question
  - How do we select instances to label to create the most accurate model at a given cost?
- Two Primary Components
  - Oracle: The oracle provides labels for queries
  - Query system: The job of the query system is to pose queries to the oracle
- Two Types of Query Systems
  - Selective Sampling
  - Pool-based Sampling

Categories

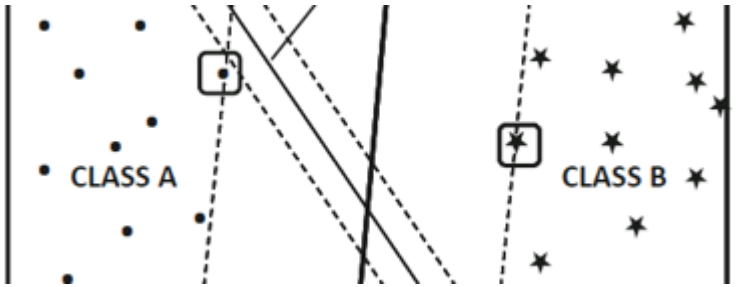
- Heterogeneity-based models
  - **Uncertainty Sampling**
  - Query-by-Committee
  - Expected Model Change
- Performance-based models
  - **Expected Error Reduction**
  - Expected Variance Reduction
- Representativeness-based models

Uncertainty Sampling

- Label those instances for which the value of the label is the least certain
- Bayes classifiers

$$\text{Certain}(\overline{X}) = \sum_{i=1}^{k} ||p_i - 0.5||$$

  - lower values are indicative of greater uncertainty
- SVM
  - Distance



Expected Error Reduction (1)

- Denote the unlabeled set as V
- Select samples from V to minimizes the prediction error of the remaining samples in V
- Select samples from V to minimizes the label uncertainty of the remaining samples in V
- Select samples from V to minimizes the expected label uncertainty of the remaining samples in V

Expected Error Reduction (2)

- Let $P_i(\bar{X})$ be the posterior probability of the label i for the query candidate instance $\bar{X} \in V$
- Let $P_j^{(\bar{X},i)}(\bar{Z})$ be the posterior probability of the label j for $\bar{Z} \in V$, after $(\bar{X}, i)$ is added to the training set
- The Error Objective of $\bar{X}$

$$E(\overline{X}, V) = \sum_{i=1}^{k} p_i(\overline{X}) \left(\sum_{j=1}^{k} \sum_{\overline{Z} \in V} ||P_j^{(\overline{X},i)}(\overline{Z}) - 0.5|| \right)$$

Representativeness-Based Models

- Heterogeneity-based models may select outliers
- Combine the heterogeneity behavior of the queried instance with a representativeness function

$$O(\overline{X}, V) = H(\overline{X}) R(\overline{X}, V)$$

  - $H(\bar{X})$ can be any heterogeneity criteria
  - $R(\bar{X}, V)$ is simply a measure of the density of $\bar{X}$ with respect to V
    - ✓ Average similarity of $\bar{X}$ to the instances in V

Outline

- Introduction
- Multiclass Learning
- Rare Class Learning
- Scalable Classification
- Semisupervised Learning
- Active Learning
- **Ensemble Methods**
- Summary

Ensemble Methods

- 三个臭皮匠顶个诸葛亮
  - Is it always possible?
- Ensemble Method
  - Different classifiers may make different predictions on test instances
  - Increase the prediction accuracy by combining the results from multiple classifiers

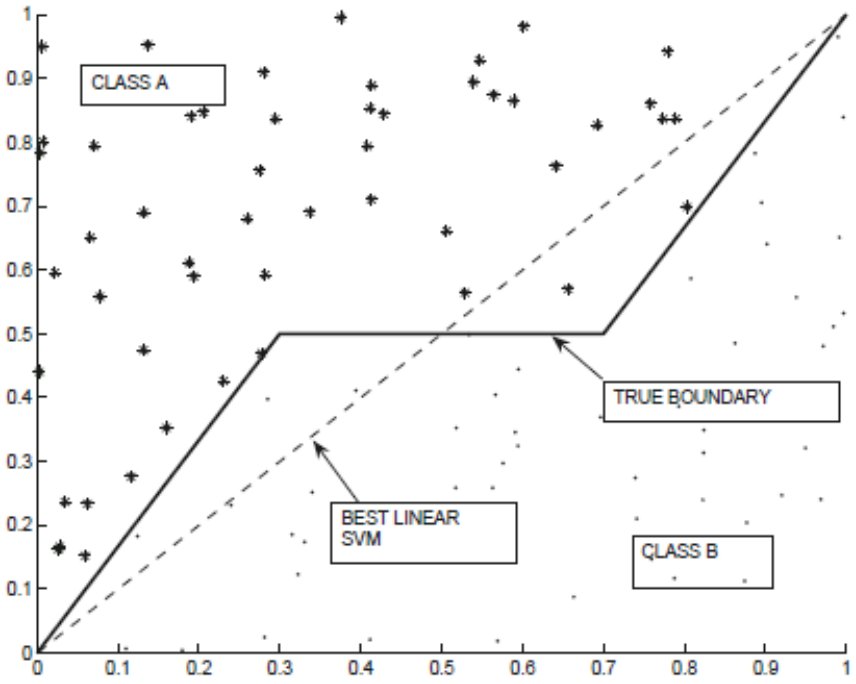
The generic ensemble framework

- Three freedoms
  - Learners, training data, combination

Algorithm *EnsembleClassify*(Training Data Set: $\mathcal{D}$
Base Algorithms: $\mathcal{A}_1 \ldots \mathcal{A}_r$, Test Instances: $\mathcal{T}$)
begin
$j = 1$;
repeat
Select an algorithm $\mathcal{Q}_j$ from $\mathcal{A}_1 \ldots \mathcal{A}_r$;
Create a new training data set $f_j(\mathcal{D})$ from $\mathcal{D}$;
Apply $\mathcal{Q}_j$ to $f_j(\mathcal{D})$ to learn model $\mathcal{M}_j$;
$j = j + 1$;
until(termination);
report labels of each $T \in \mathcal{T}$ based on combination of
predictions from all learned models $\mathcal{M}_j$;
end

Why Does Ensemble Analysis Work?

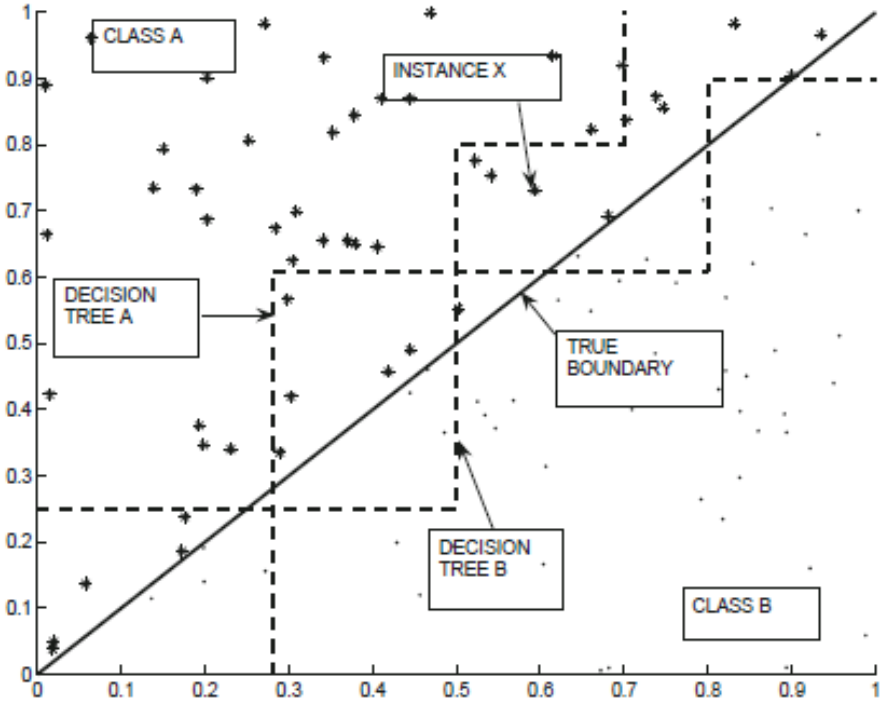
- There are three types of error
  - Bias: Every classifier makes its own modeling assumptions about the decision boundary



(a) bias

Why Does Ensemble Analysis Work?

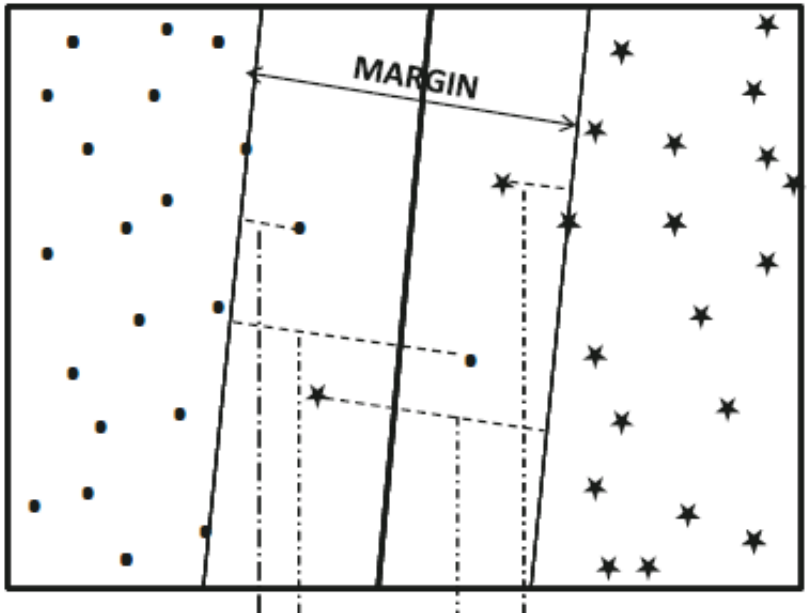
- There are three types of error
  - Variance: Random variations in the training data will lead to different models



(b) variance

Why Does Ensemble Analysis Work?

- There are three types of error
  - Noise: The noise refers to the intrinsic errors in the target class labeling

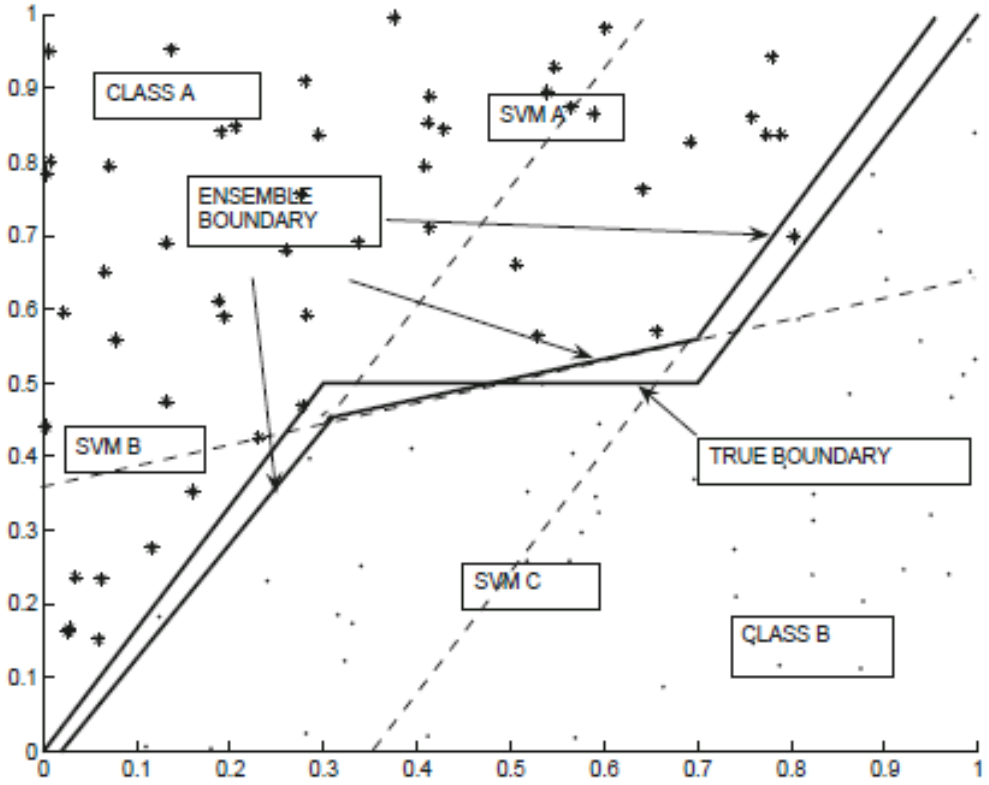


Bias-Variance Trade-off

Technique	Source/level of bias	Source/level of variance
Simple models	Oversimplification increases bias in decision boundary	Low variance. Simple models do not overfit
Complex models	Generally lower than simple models. Complex boundary can be modeled	High variance. Complex assumptions will be overly sensitive to data variation
Shallow decision trees	High bias. Shallow tree will ignore many relevant split predicates	Low variance. The top split levels do not depend on minor data variations
Deep decision trees	Lower bias than shallow decision tree. Deep levels model complex boundary	High variance because of overfitting at lower levels
Rules	Bias increases with fewer antecedents per rule	Variance increases with more antecedents per rule
Naive Bayes	High bias from simplified model (e.g., Bernoulli) and naive assumption	Variance in estimation of model parameters. More parameters increase variance
Linear models	High bias. Correct boundary may not be linear	Low variance. Linear separator can be modeled robustly
Kernel SVM	Bias lower than linear SVM. Choice of kernel function	Variance higher than linear SVM
k-NN model	Simplified distance function such as Euclidean causes bias. Increases with k	Complex distance function such as local discriminant causes variance. Decreases with k
Regularization	Increases bias	Reduces variance

Why Does Ensemble Analysis Work?

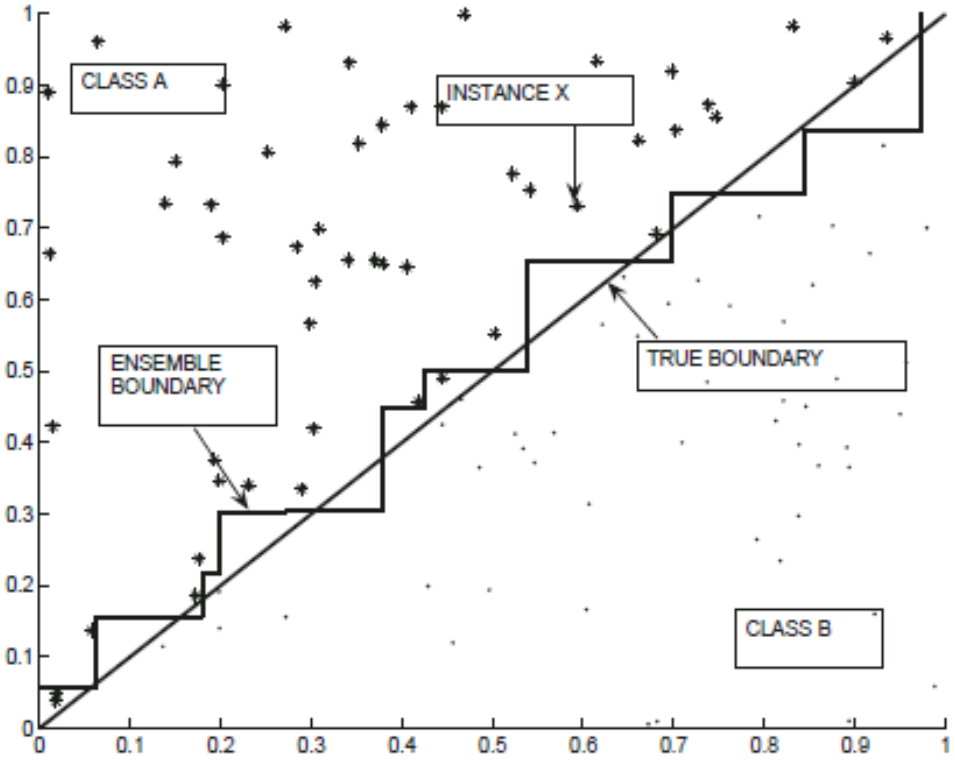
- Reduce Bias



(a) bias

Why Does Ensemble Analysis Work?

- Reduce Variance



(b) variance

Formal Statement of Bias-Variance Trade-off

- The Classification Problem

$$y = f(\overline{X}) + \epsilon.$$

  - ϵ is the noise
- Given training data $\mathcal{D}$

$$g(\overline{X}, \mathcal{D}) = \text{sign}\{\overline{W} \cdot \overline{X} + b\}$$

- The Expected Mean Squared Error over $\{(\overline{X_1}, y_1), \ldots, (\overline{X_n}, y_n)\}$

$$E_{\mathcal{D}}[MSE] = \frac{1}{n}\sum_{i=1}^{n}\left(\underbrace{(f(\overline{X_i}) - E_{\mathcal{D}}[g(\overline{X_i}, \mathcal{D})])^2}_{\text{Bias}^2} + \underbrace{E_{\mathcal{D}}[(g(\overline{X_i}, \mathcal{D}) - E_{\mathcal{D}}[g(\overline{X_i}, \mathcal{D})])^2]}_{\text{Variance}}\right) + \underbrace{\epsilon_a^2}_{\text{Error}}$$

Bagging (Bootstrapped Aggregating)

- The Basic Idea
  - If the variance of a prediction is σ^2, then the variance of the average of k i.i.d. predictions is σ^2/k
- The Procedure
  - A total of k different bootstrapped samples are drawn independently
    - ✓ Data points are sampled uniformly from the original data with replacement
  - A classifier is trained on each of them
- Prediction
  - The dominant vote of the different classifiers

Random Forests

- Bagging based on Decision Trees
  - The split choices at the top levels are statistically likely to remain approximately invariant to bootstrapped sampling
- A generalization of the basic bagging method, as applied to decision trees
  - Reduce the correlation explicitly
- The Key Idea
  - Use a randomized decision tree model
  - Random-split Selection

Random-split Selection

Random Input Selection

- At each node, select of a subset S of attributes of size q randonly
- The splits are executed using only this subset S
- $q = 1 + \log_2 d$

Random Linear Combinations

- At each node, L features are randomly selected and combined linearly with coefficients generated uniformly from $[-1,1]$
- A total of q such combinations are generated in order to create a new subset S

Boosting

The Basic Idea

- A weight is associated with each training instance
- The different classifiers are trained with the use of these weights

- The weights are modified iteratively based on classifier performance
- Focus on the incorrectly classified instances in future iterations by increasing the relative weight of these instances

AdaBoost

☐ Aim to Reduce Bias

Algorithm *AdaBoost*(Data Set: $\mathcal{D}$, Base Classifier: $\mathcal{A}$, Maximum Rounds: T)
begin
  $t = 0$;
  for each i initialize $W_1(i) = 1/n$;
  repeat
    $t = t + 1$;
    Determine weighted error rate ϵ_t on $\mathcal{D}$ when base algorithm $\mathcal{A}$
      is applied to weighted data set with weights $W_t(\cdot)$;
    $\alpha_t = \frac{1}{2}\log_e((1-\epsilon_t)/\epsilon_t)$;
    for each misclassified $\overline{X_i} \in \mathcal{D}$ **do** $W_{t+1}(i) = W_t(i)e^{\alpha_t}$;
      else (correctly classified instance) **do** $W_{t+1}(i) = W_t(i)e^{-\alpha_t}$;
    for each instance $\overline{X_i}$ **do** normalize $W_{t+1}(i) = W_{t+1}(i)/[\sum_{j=1}^{n} W_{t+1}(j)]$;
  until $((t \geq T)$ OR $(\epsilon_t = 0)$ OR $(\epsilon_t \geq 0.5))$;
  Use ensemble components with weights α_t for test instance classification;
end

Outline

- Introduction
- Multiclass Learning
- Rare Class Learning
- Scalable Classification
- Semisupervised Learning
- Active Learning
- Ensemble Methods
- **Summary**

Summary

- Multiclass Learning
  - One-against-rest, One-against-one
- Rare Class Learning
  - Example Reweighting, Sampling
- Scalable Classification
  - Scalable Decision Trees, Scalable SVM
- Semisupervised Learning
  - Self-Training, Co-training, Semisupervised Bayes Classification, Transductive SVM, Graph-Based Semisupervised Learning
- Active Learning
  - Heterogeneity-Based, Performance-Based, Representativeness-Based
- Ensemble Methods
  - Bais-Variance, Bagging, Random Forests, Boosting