Lecture 13: Ensemble Methods

- What are ensemble methods?
- Bagging
- Bias-variance decomposition: how ensembles work

Part of the slides are based on talks by Dietterich and Schapire.

Horse race prediction

- Ask a professional for advice
- Presented with a set of races, the professional can give *rules of thumb* that are better than random
- But cannot specify a single rule that is very accurate
- *How can you make money?*

Main idea

- Derive simple "rules of thumb" (i.e. classifiers) based on data
- Combine their prediction (e.g. by some voting scheme)

This works well if:

- Individual classifiers are *accurate*

  I.e. their true error is < 0.5 (better than random)
- Individual classifiers are *diverse*

  I.e. they make *independent errors*

Ensemble methods

Several popular approaches:

1. Manipulating the training examples
   - Bagging (Breiman, 1996)
   - Boosting (Freund & Schapire, 1995)
2. Injecting randomness
   - Randomized splits in decision trees
   - Random initial weights in neural networks
3. Using feature subsets
4. Changing the class label encoding

   E.g., Error-correcting codes (Dietterich, 1996)

Why is this a good idea?

Three main reasons (Dietterich, 2000):

- Statistical: reducing the risk of finding the wrong classifier
- Computational: avoiding local minima
- Representational: being able to represent hypothesis outside the language space we're considering

Bagging (Boostrap Aggregating)

Given a training set S of size m:

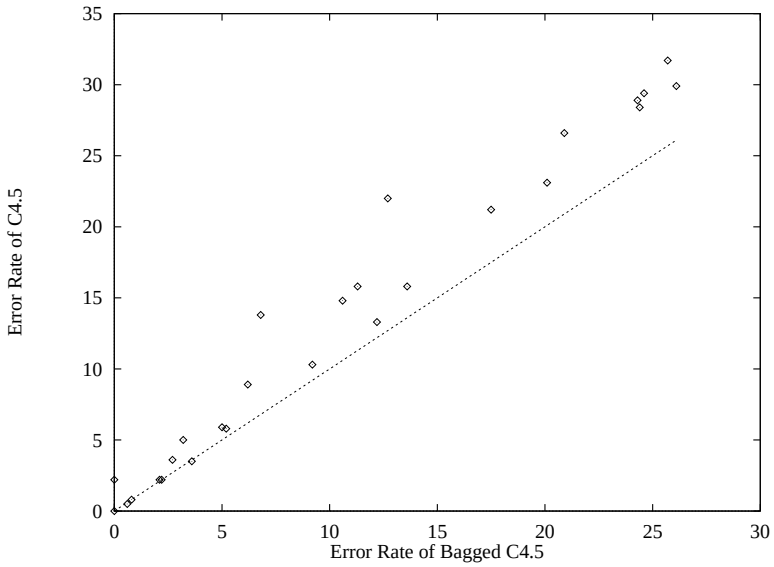
1. Construct several *bootstrap replicates* $S_1, \ldots, S_L$ (typically $30 \leq L \leq 200$)

   I.e. drawn m samples *with replacement* from S

   That means the same example can be drawn *multiple times*
2. Construct a classifier based on every set $S_1, \ldots, S_L$.

Classifying new instances is done by a majority vote among the classifiers

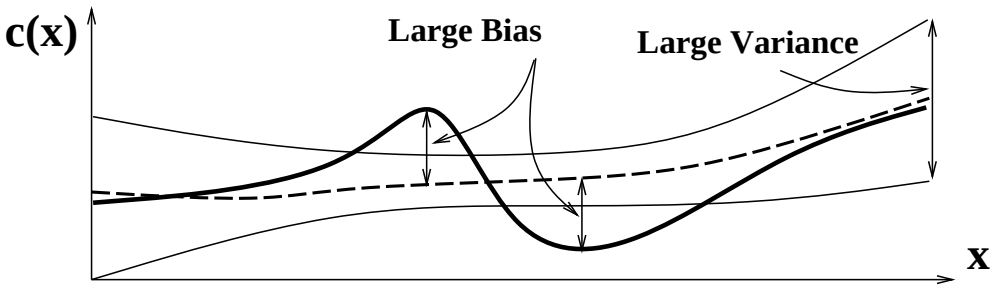
Experiment: Bagging decision trees



Why does bagging work?

Bias-variance theory

The expected error of a classifier can be decomposed into *bias* and *variance* components. (Note that we are talking here about statistical bias!)



- **Bias** comes from not having good hypotheses in the considered class
- **Variance** results from the hypothesis class containing too many

0.001in=0.401920.001in0.1in=0.401920.1in

hypotheses

Bias-variance decomposition

Several definitions of bias and variance for classification tasks (e.g. Friedman, Kong & Dietterich, Tibshirani, Kohavi & Wolpert). We will use the (Kong & Dietterich, 1995) decomposition.

- Imagine we have infinitely many training sets $S_1, S_2, \ldots$, all of size m, drawn according to the same probability distribution D.
- We train a learning algorithm A on each set and obtain a series of hypotheses $h_1, h_2 \ldots$.
- Consider a particular instance $\mathbf{x}$ and use each classifier to predict its class: $\hat{y_1}, \hat{y_2}, \ldots$. Let p be the proportion of predictions that are incorrect

Bias

The **bias** of the learning algorithm A on instance $\mathbf{x}$ for training sets of size m is:

$$Bias(A, m, \mathbf{x}) = \begin{cases} 1 & \text{if } p > 0.5 \\ 0 & \text{if } p \leq 0.5 \end{cases}$$

The bias captures *systematic errors* of a learning algorithm

An algorithm is said to be *biased* on $\mathbf{x}$ if it misclassifies $\mathbf{x}$ more often than it correctly classifies it.

Sources of bias

- Inability to represent certain decision boundaries

  E.g. linear threshold units, naive Bayes, decision trees
- Incorrect assumptions

  E.g. failure of independence assumption in naive Bayes
- Classifiers that are *too global (too smooth)*

  E.g. a single linear separator, a small decision tree

Variance

Unbiased variance:

$$Var_U(A, m, \mathbf{x}) = \begin{cases} p & \text{if } A \text{ is unbiased at } \mathbf{x} \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, this measures the error rate on examples where the algorithm is unbiased

Biased variance:

$$Var_B(A, m, \mathbf{x}) = \begin{cases} 1 - p & \text{if } A \text{ is biased at } \mathbf{x} \\ 0 & \text{otherwise} \end{cases}$$

This measures the error overestimate on examples on which the algorithm is biased.

Source of variance

- Statistical sources
  - Classifiers that are *too local* and can easily fit the data
    E.g. nearest neighbor, large decision trees, RBF
  - Classifiers with large VC dimension
- Computational sources
  - Making decisions based on small subsets of the data
    E.g. decision tree splits near the leaves
  - Randomization in the learning algorithm
    E.g. neural nets with random initial weights
  - Learning algorithms that make sharp decisions
    Can be unstable (e.g. the decision boundary can change if one training example changes)

Bias-variance error decomposition

The **expected error** of A at $\mathbf{x}$ is:

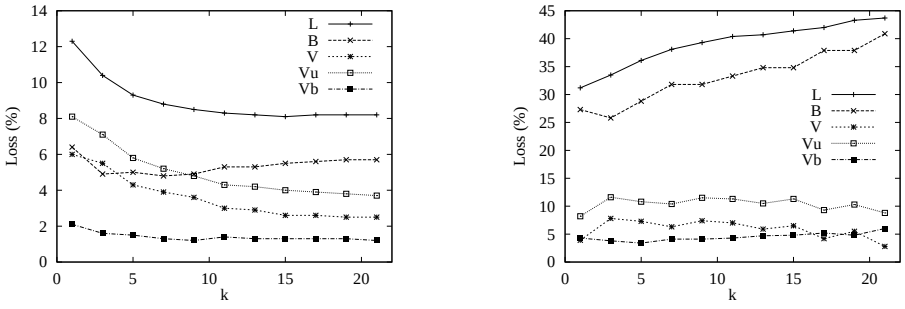
$$error(A, m, \mathbf{x}) = Bias(A, m, \mathbf{x}) + Var_U(A, m, \mathbf{x}) - Var_B(A, m, \mathbf{x})$$

If you do the computation, $error(A, m, \mathbf{x}) = p$, but this way of writing emphasizes the *source* of the errors

Two important lessons:

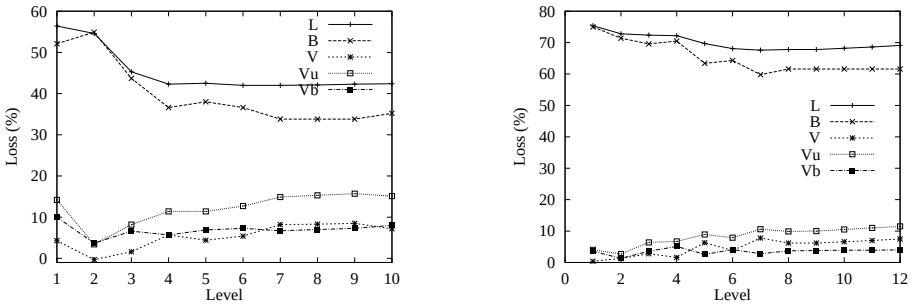
- There is usually a trade-off between bias and variance
- Just increasing the expressive power of the hypotheses does **not** necessarily improve the accuracy of the classifiers!

Effect of k in k-nearest neighbor on bias and variance



Increasing k reduces variance and increases bias

Effect of Decision Tree depth on bias and variance



Deeper decision trees reduce bias and increase variance

Why does bagging work?

- Takes several classifiers and averages the predictions
- Averaging decreases variance
- Bagging is a **variance reduction technique**