Class 15

October 06, 2022

Contents

Ensemble Methods

Have many samples and combine them to get a result or something.

Bagging: Bootstrap Aggregation

Points belonging to red/blue class in training set

I want to create multiple classifiers (that are different otherwise we are merely copying). How to create variety in classifier? We change the training data.

Bootstrap

Sampling the training data with replacement.

Basically choose a sample (***sample same size as the original training set***), use it, then put it back. Then pick sample again. In such a scenario, some samples might be picked multiple times and some not at all.

Combine the decision from these classifiers and we get the final decision. We'll discuss later how to combine the bootstrap data classifer result. This is known as **bootstrap aggregation**.

- Reduces the variance of an estimated prediction function
- For classification, a committee of classifiers each cast a vote for the predicted class.
- Another advantage of bagging is that the computation can be parallelized.

Boosting

Another variant of ensemble members. Here each classifer is tuned to be *complementary* to previous ones.

So we see where does a classifier perform well and where does it not perform well. Then we try to generate a classifier that is complementary to this performance.

We combine them using a weighted average (weights proportional to their performance on validation set).

AdaBoost is a popular boosting method.

Example

Choose a weak classifer with boundary equation $h_1(x) = 0$ and $H(x) : sign(h_1(x))$ is the classifier.

We combine them as

$$H(x) : sign(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$$

Strong vs Weak Classifier

Strong classifier is able to model very complex boundaries (potential to become very good).

Good classifer based on how well it classifies data.

Random Forest

- Ensemble Classifier
- Consist of many decision trees
- CART Method (recursive binary splitting) recap
- Outputs the class that is the mode of the class's output by individual trees
  - Method combines "bagging" and the random selection of features
  - Uses *decorrelated* trees

Just changing the question at the top (root) node of the tree creates very different decision trees.

1. Training data: N examples and M features
2. Create bootstrap samples from the training data (We keep all the examples). But we select a subset of features. So, first set has let's say some 20 features. Then the next set will have some randomly picked 20 features (might be different from the first one).
3. Now, we create decision trees for each of these sets. We know the decision trees for each set will be different (since the features are different).