Machine Learning Homework 6

Hao Liu

April 10, 2017

1 CART Implemenation

In this section, we will work on implementation of decision tree model as classifier and regressor (CART). We will use the SVM data from homework 4 for training and visualization.

1.1 Decision Tree Class

Complete the class **DTree**. Indeed you need to implment two functions in this class. **split_tree** function will be used for finding the best feature and its corresponding value for split. **split_node** function will apply **split_tree** to find its children.

1.2 Decision Tree Classifier

Complete the class **DTree_classifier**, which inherits from **DTree**. You need to:

1. choose either one of **compute_entropy** and **compute_gini** to complete
2. define the left and right subtree at initialization part
3. complete **compute_probability** function

Try either entropy or gini as split criterion. Evaluate the results and plot decision boundary for training set. You can also compare your model with decision tree from sklearn package for debugging. (Note that visualization for tree model requires **graphviz** installed.)

1.3 Decision Tree Regressor

Complete the class **DTree_regressor**, which inherits from **DTree**. You need to:

1. choose either one of **mean_square_error** and **mean_absolute_error** to complete
2. define the left and right subtree at initialization part

Evaluate the results and compare with decision tree regressor from sklearn package for debugging.

2 Gradient Boosting Implementation

In this section, we will implement gradient boosting method based on tree model. The data we are going to work with are svm data and kernelized regression data from homework 4.

2.1 Gradient Boosting Regressor

Complete the class **gradient_boosting**. Please use **Dtree_regressor** you defined in last question to train each decision stump in gradient boosting. We are going to use L2 loss. You can use **pseudo_residual_L2** function to compute residual. However, in general, this class should take a function for computing pseudo-residual.

1. Define function **fit** and **predict**.

2. Train your GBM model on svm training data, plot the decision boundary with different number of estimators. You don't need to plot the contour map though it is a regression model. Simply visualize the positive and negative region is fine.

3. Train your GBM model on kernelized regression training data, plot the function curve with different number of estimators to see how GBM is fitted to training data.