ECE 562 Machine Learning, Fall 2019
Instructor: Zafer Aydın
Computer Homework 2

Introduction

Download the template codes for Udacity's Intro to Machine Learning course by running the following command

git clone https://github.com/udacity/ud120-projects.git

Submit your scripts as q1.py and a text document that includes your answers to the questions below through Canvas.

Assignment

1. Using the starter code in decision_tree/dt_author_id.py, get a decision tree up and running as a classifier, setting min_samples_split=40. It will probably take a while to train. What's the accuracy?

2. You found in the SVM mini-project that the parameter tune can significantly speed up the training time of a machine learning algorithm. A general rule is that the parameters can tune the complexity of the algorithm, with more complex algorithms generally running more slowly.

Another way to control the complexity of an algorithm is via the number of features that you use in training/testing. The more features the algorithm has available, the more potential there is for a complex fit. We will explore this in detail in the "Feature Selection" lesson, but you'll get a sneak preview now.

What's the number of features in your data? (Hint: the data is organized into a numpy array where the number of rows is the number of data points and the number of columns is the number of features; so to extract this number, use a line of code like len(features_train[0]).)

3. go into `../tools/email_preprocess.py`, and find the line of code that looks like this:

selector = SelectPercentile(f_classif, percentile=10)

Change percentile from 10 to 1, and rerun dt_author_id.py. What's the number of features now? What do you think SelectPercentile is doing? Would a large value for percentile lead to a more complex or less complex decision tree, all other things being equal? Note the difference in training time depending on the number of features.

4. What's the accuracy of your decision tree when you use only 1% of your available features (i.e. percentile=1)?

5. Implement the k-nearest neighbor classifier and try choosing the k parameter from 1 to 10 with increments of 1. What is the best accuracy? Which k value gave the best accuracy?

6. Implement the AdaBoost classifier and try choosing the number of estimators parameter from 5 to 600 with increments of 5. What is the best accuracy? Which number of estimators value gave the best accuracy?

7. Implement the random forest classifier and try choosing the number of trees parameter from 5 to 500 with increments of 5. What is the best accuracy? Which number of trees gave the best accuracy?