Portfolio Component 5: Kernel and Ensemble Methods

Andrew Gerungan – awg190000 – 10/23/2022

SVM

Support Vector Machines, known as SVMs, are algorithms that essentially divide a dataset into different regions and utilize the way that it separates the data into different regions to predict which side of the margin an instance falls on as to predict test values. Not all data is easily linearly separable, so SVM kernels exist to map the existing linear function of the dataset to another form that is linearly separable. One such example of an SVM kernel is the polynomial kernel, which maps the data to a higher dimension so that it is linearly separable. The other SVM kernel that we worked with in this pfc is radial kernel, which instead adds an additional hyperparameter that controls the shape of the hyperplane boundary. My take on the advantages of SVM is that it works well when there is a clear separation that can be found in the dataset and when the dataset works well in high dimensionality. In terms of disadvantages, the biggest I found was that on larger datasets, the SVM's speed dropped dramatically. It also doesn't work very well when the classes overlap a lot.

Random Forest

Random Forest works very similarly to simple decision trees except that it uses different data samples and different subsets of features for each tree, attempting to reduce variance. Some of the advantages of this is that it reduces overfitting in the decision trees and is flexible. One of the disadvantages of this is that because it has to produce multiple trees for its combination, it requires more computational power.

Adabag Boost

Boosting is similar to Random Forest but it uses the full set of predictors as well as the entire training data and then produces the various trees for combination. One advantage of this is that it reduces overfitting; one disadvantage of this is that because it tries hard to fix outliers, it almost is reliant on outliers.

XGBoost

The XGBoost algorithm works similarly to the general boosting algorithm; firstly, it requires that the training data is converted to a numeric matrix and the training labels are converted into 0/1 integers. It also uses a setting of nrounds = 100. The biggest advantage of this is that it runs quickly and efficiently. It also works well with large data sets. One of the disadvantages of XGBoost is that it can over-fit data, especially with noisy data with many outliers.