Kernel and Ensemble Methods

By Aditi Chaudhari and Abigail Solomon

Kernel Methods

The Kernel Method that we chose to use for both regression and classification was SVM, or support-vector machines. SVM separates the data points in each class using a hyperplane. The hyperplane exists in multidimensional space since many data sets have multiple predictors. On either side of the hyperplane, there are margins and any instances that exist on the margin are referred to as support vectors. An instance is classified through determining what side of the margin the instance falls on. A few observations may fall on the wrong side if the data cannot be entirely separated by the margins, and these observations are known as slack variables. Slack variables can ensure that the model does not overfit the data. The term C dictates how much of an impact the slack variables can have on finding an optimal decision boundary. Cost values that are larger lead to larger margins, while cost values that are small lead to smaller margins. A validation set can be used to find the optimal value of C.

A linear classifier may not be able to classify the data accurately, so data can be mapped to a higher dimensional space and a polynomial kernel or radial kernel can be used to separate the data. For the radial kernel, an additional hyperparameter known as the gamma parameter controls the bias-variance tradeoff. A larger gamma value could have low bias and high variance and consequently overfit the data. On the other hand, a smaller gamma value could have high bias and lower variance. These kernels can be used for classification or regression.

Using SVM has advantages and disadvantages. An advantage that we found is that different values of C and gamma can be used to provide higher or lower variance and bias. A disadvantage of using SVM is that it is computationally expensive to train and can take a long time to run the algorithm.

Ensemble Methods

Ensemble methods are machine learning algorithms that combine several weak learners either sequentially or in parallel. The sequential approach involves combining the weak learners one after the other, while the parallel approach involves using weak learners at the same time and then aggregating the results. The ensemble methods that were explored in this assignment were Random Forest, XGBoost, and AdaBoost. The results from each ensemble method were compared to each other and the results obtained by a decision tree.

First, lets discuss how Random Forest works. The Random Forest algorithm makes use of bagging, or repeatedly sampling data from the data set in order to minimize variance. At every split within the tree, a subset of random predictors is chosen from all of the predictors and then one is chosen. Usually, the size of these subsets is the square root of the number of predictors. This happens over and over until the tree is built. Random Forests are advantageous over decision trees because they prevent trees from choosing similar predictors in the same order, which allows trees from the strongest predictor to be discovered first. The disadvantage of using the Random Forest algorithm over decision trees is that Random Forest uses a greedy algorithm, but this is mitigated by limiting the choice of predictors.

Next, lets take a look at the strengths and weaknesses of each ensemble method. The Random Forest algorithm was more accurate and had a higher Matthews Correlation Coefficient in comparison to the decision tree that was created. However, a major weakness of the Random Forest algorithm is that it took a long time to run. Initially, we tried running the Random Forest algorithm with the entire dataset, which contained 284,807 observations. Our computers could not handle running the Random Forest algorithm for that many observations, so we changed our data set to only have 10,000 observations, and our Random Forest algorithm finally worked correctly. In comparison to the other ensemble methods, the Random Forest algorithm took the most time to classify the credit cards as fraudulent or non-fraudulent. XGBoost is advantageous in that it is able to run up to 10 times faster than earlier algorithms since it was written in C++ and makes use of multithreading. In comparison to other ensemble methods, XGBoost took the least amount of time to classify the credit cards as fraudulent or non-fraudulent. XGBoost also was tied with the Random Forest algorithm for the most accurate ensemble method that we used. A potential weakness of XGBoost, though, is that the train and test data must be processed in the right way prior to using the algorithm. If someone is unfamiliar with how to process the data, the results will be skewed. Finally, AdaBoost did not perform as well as the Random Forest algorithm or the XGBoost algorithm when comparing accuracy or the Matthews Correlation Coefficient, and it took a while to run the algorithm. Both of these points highlight a weakness with the AdaBoost algorithm. While each ensemble method has its own strengths and weaknesses, all three of them did outperform the decision tree with regards to accuracy and the Matthews Correlation Coefficient, which makes them more accurate than decision trees.