Aloksai Choudari and Alekhya Pinnamaneni
Dr. Mazidi
CS 4375.003
Kernel and Ensemble Methods

Narrative

The support vector machine (SVM) is a supervised learning algorithm that is used for machine learning on regression and classification datasets. The SVM separates classes of data by drawing a separating hyperplane in between both classes, which has margins on either side that separate the observations in both the classes. The observations that fall on the margins are called support vectors. Designating an instance to a class is done by determining which side of the margin an observation falls on. Standard SVM uses a linear decision boundary and assigns data to a class based on which side of the decision boundary it falls on. The margin is the perpendicular distance from this decision boundary to the support vectors on both sides of the boundary. The width of the margin is determined by calculating the difference of two support vectors, one from each side. The classification rule for a new vector will either be +1 or -1. For support vectors, the classification rule is 0. An important hyper-parameter for SVM is C. The C hyper-parameter controls how slack variables are allowed in the SVM, or how many observations are allowed to fall on the wrong side of the margin. Smaller C values can cause overfitting and result in lower bias and higher variance. Increasing the C value to allow some slack variables can reduce the risk of overfitting the dataset. SVM has three kernel methods: linear, polynomial, and radial. The linear kernel uses a linear decision boundary and works best for datasets where the classes are linearly separable, which is uncommon because data is not usually that simple. When the data is not linearly separable, polynomial kernels and radial kernels work better because they map the data to a higher dimension. The polynomial kernel and radial kernels use a non-linear decision boundary, which often generates more accurate results. Radial kernels are a step up from polynomial kernels and have an additional hyper-parameter called gamma. Gamma controls the bias-variance tradeoff. A larger gamma value can lead to overfitting with low bias and high variance. A smaller gamma value can lead to underfitting with higher bias, but lower variance. Tuning should be used to find the optimal values for the C and gamma hyper-parameters that will generate the most accurate and representative SVM model. There are many pros to using SVM. The SVM is easy to implement in code, and it is easy to interpret the model. SVM can also be effective even if the size of the training dataset is small and is even

more effective in high dimensional spaces. However, SVM is not as effective for larger datasets and doesn't work well when there are too many attributes in the dataset.

Ensemble methods work by combining multiple weak learners, either sequentially or in parallel, to create a strong learner. In the Ensemble notebook in this portfolio, we used three ensemble techniques to perform machine learning on a multi-class classification dataset: Random Forests, XGBoost, and Adabag. The Random Forest algorithm works by generating multiple uncorrelated decision trees in parallel, and combines them to get the best model. An advantage of random forest is that it generally has excellent performance in terms of accuracy and runtime, due to the parallel and decorrelated nature of the algorithm. A downside of the random forest algorithm, however, is that it is not easy to interpret and cannot be plotted. XGBoost also builds multiple decision trees in parallel, but it improves on the random forest's algorithm by taking advantage of multi-threading get much faster runtimes. An advantage of XGBoost is that it runs much faster than previous algorithms and the results can be plotted, unlike other algorithms including random forest. A downside of the XGBoost algorithm is that the training data has to be converted into a numerical matrix, and the training labels must be encoded into 0s and 1s. The Adabag algorithm sequentially builds decision trees one at a time, increasing the weights of observations with larger errors each time to improve the accuracy of the next learner (decision tree). An advantage of the Adabag algorithm is that it learns e and increases accuracy over each iteration, which can result in more accurate predictions. A downside of the Adaboost algorithm is that it runs much slower than the Random Forest and XGBoost algorithms because the algorithm works sequentially instead of in parallel.