CS 4375.003
October 23^{rd,} 2022
Fernando Colman and Linus Fackler

Kernel and Ensemble Narrative

SVM

The most important aspect of how Support Vector Machines work is the concept that the goal of the algorithm is to find an optimal separating line or hyperplane that will divide the data into two different sections. It does so by taking in the data it's training on as input and then outputting the hyperplane. The definition of the hyperplane line is typically that it's linear and straight, while the definition of an 'optimal' line means that the line has the greatest distance from both sides of the data which are closest to it, meaning that the line has the greatest 'margin' from both of the partitioned data sets. These distances from the hyperplane to the data points are typically calculated as the Euclidean distances from the data points to the line itself, which we call vectors. However, in higher dimensions, the Euclidean distances might not suffice to measure errors from the line. This is why, besides being able to make straight lines separating the data into two sections, SVM can also use kernel functions to change how the datasets are represented and therefore allow for the creation of non-linear hyperplanes. For example, SVM can use the dot product between two dots (or the line) instead of the Euclidean distances between them to create a line that isn't linear or straight. The fact that you can change the kernel of SVM is both a strength and a weakness. It's a big strength when you can clearly see which kernel function to use for a particular dataset, but a big weakness when choosing the kernel function isn't so straightforward, and therefore choosing a bad one can either overfit or even underfit the data.

Random Forest, XGBoost, AdaBoost

The algorithms of Random Forest, XGBoost, and AdaBoost are all very similar in the sense that they are all ensemble methods, meaning that they combine the results of multiple models to give an output. However, they all have their differences as to how that ensemble technique is actually carried out. For Random Forest, its methodology is actually relatively simple since what it does is produce a very large amount of trees and then averages (or takes the majority) of their output to arrive at a conclusion. This is why the name "Random Forest", with the random side coming from the fact that the trees need to not be related to one another. This methodology takes advantage of the fact that it's very possible for an individual tree to give an erroneous output based on some small factor, but trusting the entire forest of trees is much more reasonable to eliminate that chance of random error. AdaBoost is very similar to Random Forest except that its decision trees only have a depth of one, but its XGBoost that is different since it actually calculates residuals for its tree and uses every tree as a sequential improver for the next one, making the last tree the final result.