Narrative Document of Kernel and Ensemble Methods

Group Member: Bishal Neupane, Saugat Gyawali

What is SVM?

SVM or support vector machines is an algorithm that can be performed in binary classification, multiclass classification, and regression. It is a supervised learning model. The main objective of an SVM is to find a hyperplane in multiple-dimensional shapes which can classify the data points distinctly.

How SVM works?

Support vector machines work based on mapping data into high-dimensional features to categorize the data. The data are categorized by the separator, also known as a hyperplane. For SVM, we need to select the best hyperplane. We need to select in such a way that the distance to the nearest element of each tag to the separator is the largest. It was for linearly separated data. Imagine there is a case when it is not linearly separated, we can increase it by adding a new dimension. For SVM regression, it is essential to decide on a decision boundary. The decision boundary needs to be kept in such a way that the data points which are closest to the hyperplane need to be in the boundary line. And we need to take care of data points that are in between the boundary line. The best-fit hyperplane would be one that has the maximum number of points in between them.

How do kernels work?

As we mentioned earlier, SVM works by mapping data into a high dimension. But we can't always figure out how we map into a higher dimension. So, we need a set of mathematical functions which are defined with the help of a kernel. What the kernel does is that it takes input data and transforms it into the given form. In a feature space, it returns the inner product. There are different types of kernels. For example, the linear kernel is used when there is a dataset that is linearly separable, or it can be separated by using a single line. Another one is a polynomial kernel; we need it when we need to transform it into a higher dimension so that the hyperplane can divide or categorize the data. The next one is the radial kernel which uses an additional hyperparameter called gamma which controls the shape of the boundary. It depends on the value of gamma whether it considers the data points in a decision boundary or not.

Strengths of SVM:

They are very good whenever we do not have any idea of data how is formed or what it looks like. Moreover, In higher dimensional space, they are more effective. Also, the kernel is the real strength of SVM because we can use such mathematical models to update. Implementation of SVM is also easy.

Weakness of SVM:

They are inefficient for large datasets. Moreover, it will not perform well when the number of features is higher than the number of training data. Also, there is no probabilistic explanation of the categorization.

Second Part:

How does random forest work?

The main idea behind random forest is that it trains the multiple trees on a subset of those data. This algorithm uses a different data sample and subsets of features. There is a variance reduction because of it. It consists of a group of decision trees. As we know, when we use a lot of uncorrelated decision trees then there is a more stable result. Whenever we are training our model using a random forest, we are training a decision tree group. And we chose the one which has the most votes that produce an accurate result.

Comparison of Random Forest with XGBoost:

One of the main differences is that random forest trains the multiple decision trees on a subset of those data whereas XGBoost gives a score before entering into the modeling process. The run time of XGBoost is fast because XGBoost creates the decision tree and pass the result from one tree to another and use that value to calculate the result. But random forest creates many decision trees and later combine the result which makes it slower.

Comparison of Random Forest and AdaBoosting:

Random Forest creates many Decisions tree and later combine the result of all trees to get the final result. But AdaBoosting uses something called Decision Lumps which is basically a node and its two children. So, AdaBoosting is faster than Random Forest but the accuracy might be higher in the random forest because some of the information is lost in AdaBoosting because only a node and two children are passed.

Strengths of ensemble technique:

It has higher predictive accuracy.

They can efficiently work for both linear and non-linear data.

It is more stable.

Weakness of ensemble technique:

It is hard to predict and interpret.

It highly depends on selection, if we select the wrong one there is a high chance of reducing predictive accuracy.