CS 4641 Project 1 Report

HU Heng

February 7, 2019

1 Overview

This report is for CS 4641 Machine Learning project 1 supervised learning. In the following pages, you will see analysis of five different learning algorithms. I will show the results by some figures which is about the performance of each algorithm on different hyperparameters. I will first introduce the datasets I used in my experiences. Then the five algorithms and the corresponding results will be described and analyzed. Finally, I will show how the size of training dataset influences the classification result.

2 Datasets

The datasets I choose include wine dataset and adult dataset. I will introduce these two datasets and explain the reason why I choose these two datasets in detail.

2.1 Adult Dataset

Adult dataset is extracted from the 1994 Census database. There are 14 attributes and the result is whether the income is greater than $50k or not. It is a binary classification problem and there are 48842 instances in total. I believe there should be enough data to train a good model using different algorithms. In addition, binary classification problem is not complicated and it would be suitable for training. The training result should be reasonably good.

2.2 Wine Dataset

Wine data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. There are 13 di-

mensions and three possible classes. There are 178 instances in total. Previously, working on adult dataset is a binary classification task and there are more than 10k training data. I would like to move from binary classification problem to multi-class classfication problem. In addition, I am not sure whether 178 instances are large enough for classification. So I would like to do classification on this dataset and see the preformance of different algorithms.

3 Adult Dataset Classification and Result

3.1 K Nearest Neighbor

For k nearest neighbor algorithm, the hyperparameter is the value k. Fig.1 shows the accuracy with different k. We can see clearly from the figure that the training accuracy goes down rapidly at beginning and the validation accuracy increases as k becomes larger. When k is at around 20, the accuracy of both training set and validation set become stable. We may conclude that

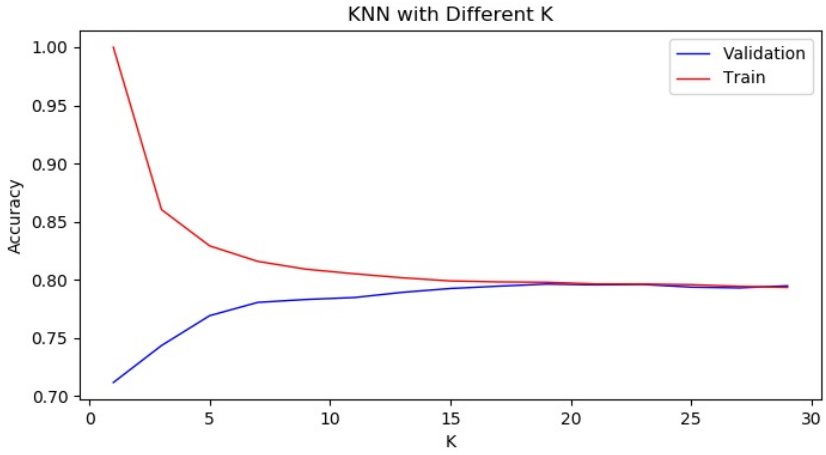

Figure 1: Accuracy with Different K

when k is small, overfitting may happen since there is a large gap between training accuracy and validation accuracy. This is reasonable because if we only look at a small number of neighbors, there would be a higher error probability.

For the selection of hyperparameter k, we may conclude that k=19 is good

and the model achieves 79.63% accuracy in testing dataset.

3.2 Decision Tree

For decision tree algorithm, the hyperparameters include max number of leaves, max depth, etc. Here I test on different choose of max number of leaves and I use information gain for the criterion. From Fig.2, we can see clearly that the accuracy raise up rapidly first when leaf number is very small. Then the accuracy raise up slowly. After the number of leaves is greater than 20, the accuracy almost does not change. This makes sense because when a decision tree can be fully produced, there is no use to increase the maximal number of leaves. We may conclude that when the maximal

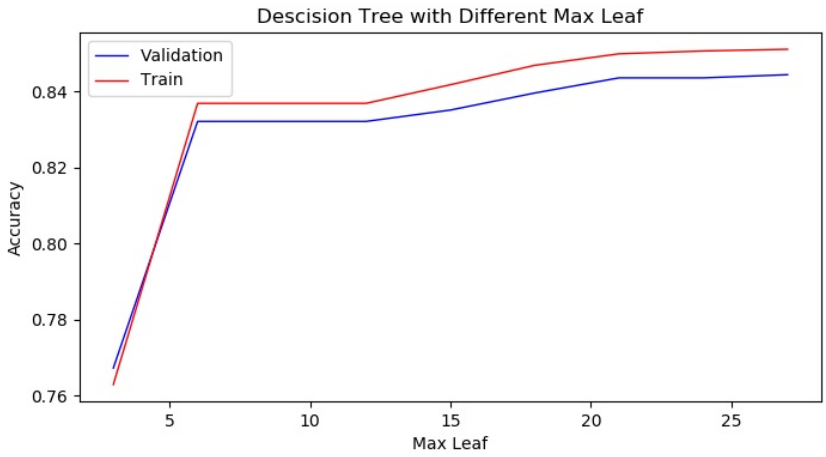

Figure 2: Accuracy with Different Number of Max Leaves

number of leaves is so small, increasing the maximal number can make the performance of the algorithm much better while when k becomes larger, the increase of the accuracy is not that obvious. I believe there is a trade-off between the complexity of the decision tree and the accuracy. If one want his decision tree to reach as higher accuracy as possible, he may choose to have no limitation on maximal number of leaves. On the contrary, if one would like to limit the complexity of his decision tree, he may just choose a reasonably small number as the limit.

For the selection of this parameter, I choose max number equals 21 and the test accuracy is 84.40%. The testing accuracy is actually very close to

training and validation accuracy, which indicates that overfitting does not happen here.

3.3 Boosting

Ada boosting is used as boosting algorithm here. The hyperparameter I choose here is the number of estimators. From Fig.3, we can see that the training accuracy and validation accuracy both increase as the number of estimators increase. However, if we look at the data, the accuracy only increase in a small number. We may conclude that the number of estimator

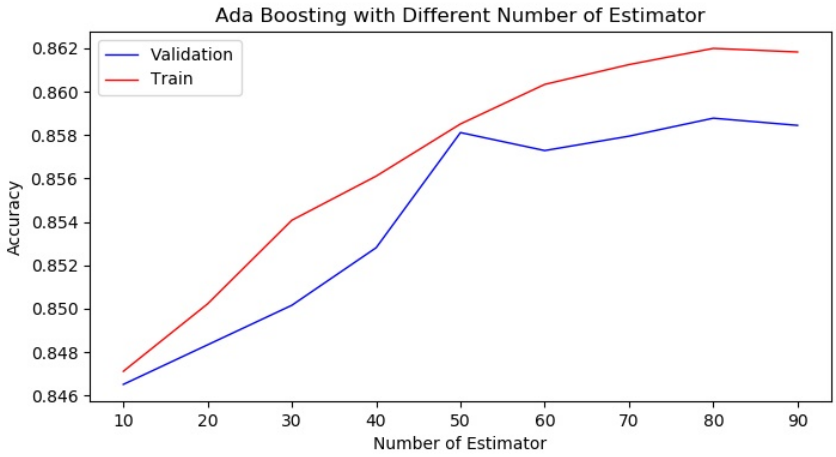

Figure 3: Accuracy with Different Number of Estimators

does not have great influence on the algorithm but it is a good way to increase the accuracy after it reaches a high value.

3.4 Support Vector Machine

For support vector machine, I use four different kernel functions: poly, linear, rbf and sigmoid. The performace of each kernel function is shown in Fig.4. It is clear that rbf function is the best one among those four and sigmoid is the worst one. An interesting thing is that the gap between training accuracy and validation accuracy is very samll for all of these four algorithms. I would say that support vector machine is a good algorithm because it seems overfiting may not happen.

Using rbf as kernel function and the test accuracy is 83.22%.

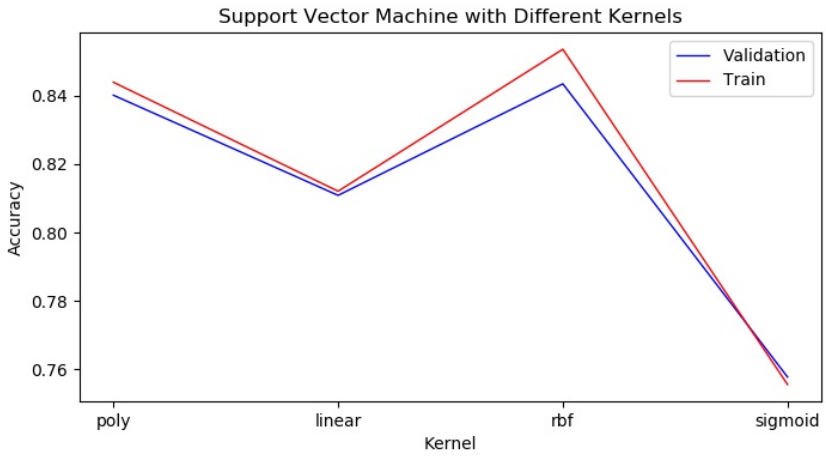

Figure 4: Accuracy Using Different Kernel Function

3.5 Neural Network

Neural network has many hyperparameters. I test on the number of hidden layers and fix other hyperparameters. From Fig5, it is clear that the accuracy of neural network is not stable. It first decreases as number of layers goes up and then increases. The best accuracy is attained when there are 19 hidden layers of neural network. It looks strange for me as the trend

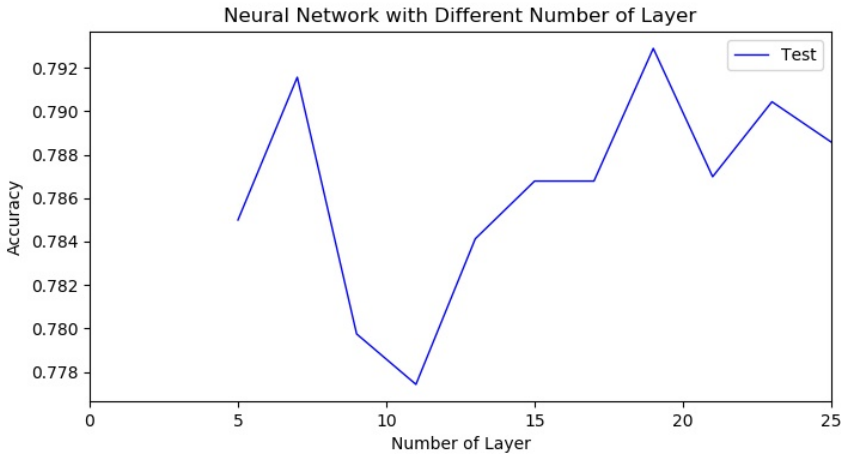

Figure 5: Accuracy with Different Number of Layers

of accuracy with regard to the number of hidden layers is not consistent. Neural network seems to be a black box for me as I really don't know what happens inside the hidden neurons.

The only thing I can tell is that the test accuracy for neural network with 19 hidden layers is 78.86%. It is not bad compared with other algorithms.

3.6 Summary

Table.1 shows the performance of different algorithms on test data. Decision tree seems to be the best algorithm for adult data and neural network does not work as well as other algorithms. At my first sight, it is contradictory to my experience that neural network is used broadly and has get excellent result. I think the point is that this dataset is a trivial one. The dimensions and size of the dataset is not complicated and traditional machine learning algorithms can handle this problem well. As far as I know, neural network

Table 1: Performance of Algorithms

Algorithm	**Testing Accuracy**
KNN	79.47%
Decision Tree	84.40%
Boosting	82.34%
SVM	83.23%
Neural Network	78.86%

needs large scale data to get trained and the hyperparameters need to be set up correctly in order to have faster training speed and better accuracy. For this training set, neural network can get a good testing accuracy, but not the best. Other traditional machine learning algorithms work even better.

4 Wine Dataset Classification and Result

4.1 K Nearest Neighbor

Based on Fig.6

4.2 Decision Tree

Based on Fig.7

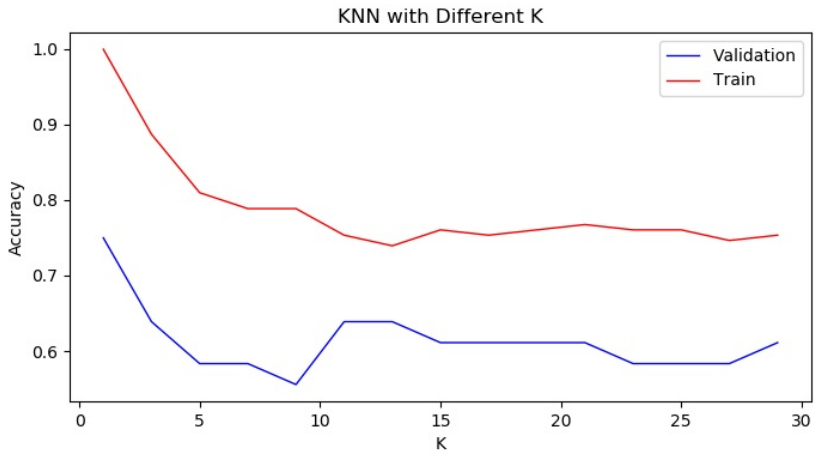

Figure 6: Accuracy with Different k

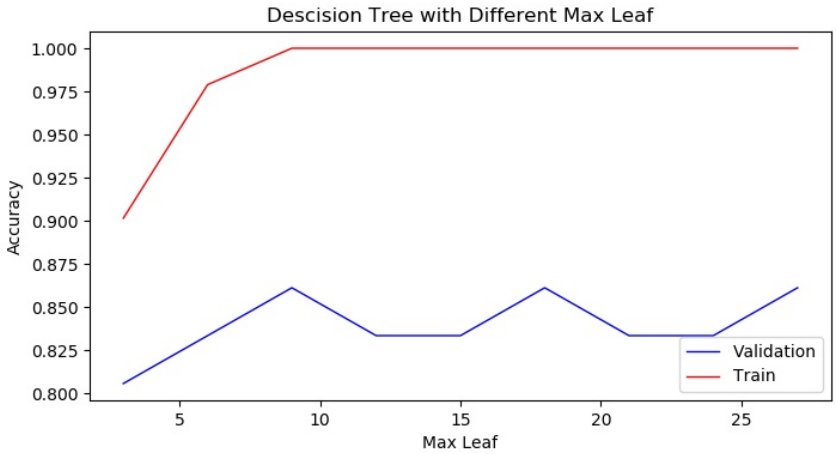

Figure 7: Accuracy with Different Max Number of Leaves

4.3 Boosting

Based on Fig.8

4.4 Support Vector Machine

Based on Fig.9

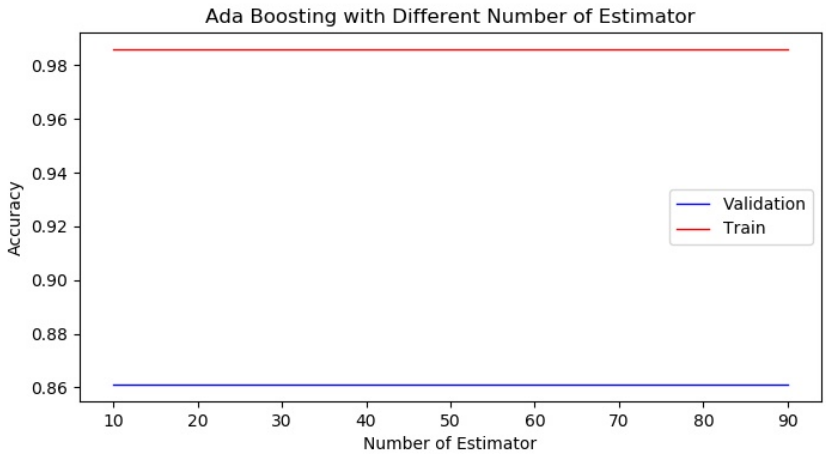

Figure 8: Accuracy with Different Number of Estimators

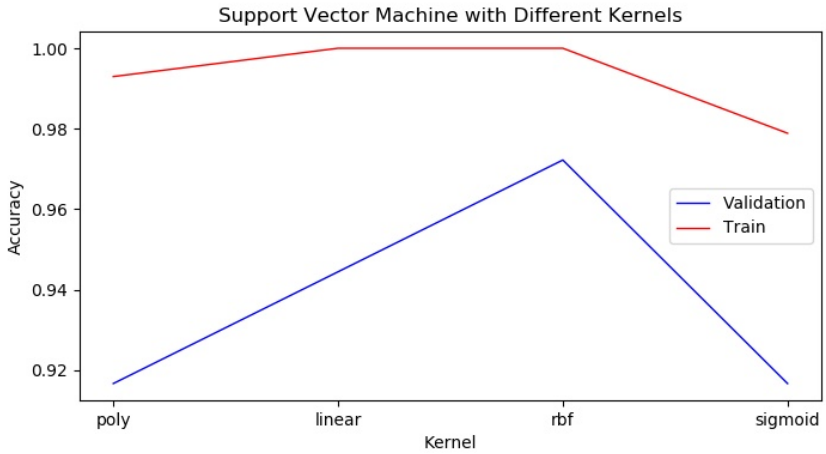

Figure 9: Accuracy Using Different Kernel Function

4.5 Neural Network

Based on Fig.10

4.6 Summary

5 Influence of Size

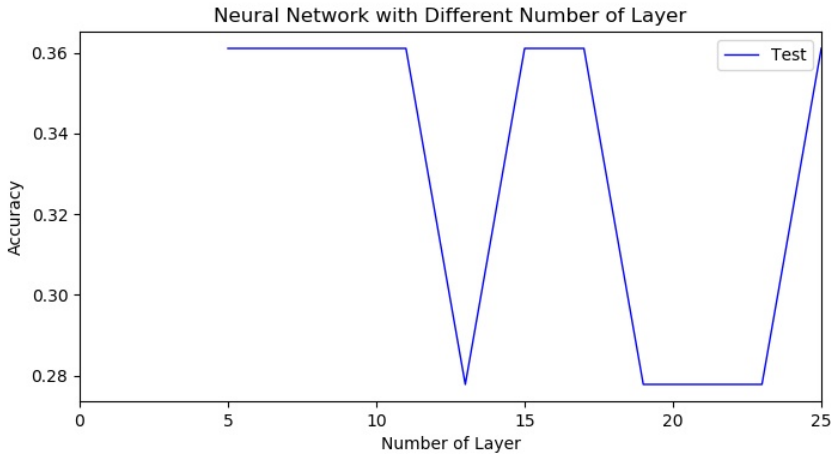

Figure 10: Accuracy with Different NUmber of Layers