The Influence of Dataset on Accuracy of Multiple Supervised Classifiers

Xin Li

June 10, 2017

Abstract

Nowadays more and more efficient supervised methods for classification has come out. In terms of the great difference of characteristics among these classifiers, **Caruana** and **NiculescuMizil** has compared empirical learning algorithms(Caruana and Nicu,2006)[1]. In this paper, I am looking forward to explore the accuracy of Random Forest, K-Nearest Neighbors and Boosted Decision Tree on different kinds of datasets and find out the accuracy of these algorithms in predicting data.

1 Introduction

Throughout 1980s till 2020s, there comes out Neural Network, SVM, Random Forest, Boosting and Deep Learning. Nowadays, the most advanced Deep Learning has been implemented in lots of areas, like Artificial Intelligence, Interpolation Between Domains(Sumit,Suhrid,Raghuraman,2013)[2]. However, in the fields of classification and data prediction, the algorithms developed during these time still have their significant advantage over each other based on different types of dataset. In the paper, I'll go through different types of dataset, and use Random Forest, KNN as well as Boosted Tree(BST-DT) to set up models and compare and analyze the accuracy for these classifiers.

2 Data and Problem

I am using 3 sets of data `Adult`, `Letter` and `Bank Marketing` from http://archive.ics.uci.edu/ml/ [3] UCI repository. For each set of data, I take 5000 sample as training set and the rest as the testing set. For these 5000 samples, use 5-fold cross validation to find the desired model by splitting the 5000 samples into 5. Each time take 4000 to train and 1000 to test. Then compute the average accuracy for one classifier.

dataset vs type	Training Set	Testing Set	Features	No.Class
adult(cleaned)	5000	27321	15	2
adult(source)	5000	27561	15	2
Letter(binary)	5000	15000	171	2
Letter(26)	5000	15000	17	26
Bank Market	5000	40211	15	2

2.1 Adult

For Adult dataset, there are in total of 32561 data points and 15 features. This is the most complex dataset of the 3 dataset selected for the test. Of the features in the table,9 of them are categorical and 6 of them are numeric. We are taking the first 14 as input and last one **K** value as output. There exists 7% missing data in the input. To handle the missing data in the set, there are mainly two ways to deal with that. One safe way is delete the rows where there are missing data. The second way is to give a difference label of missing data. It is notable that, Random Forest are more ideal to deal with missing data. (https://www.stat.berkeley.edu/ breiman/RandomForests[4] I implemented both ways of dealing with these data. To take care of categorical part, I am using the one-hot encoding to show all characteristics on features. So after the cleaning, the dimension of the dataset is 32321*74.

2.2 Letter

For Letter dataset, there are 20000 data points and 17 features. The first one is the output which is `Letter`. The rest of them are numbers which can be easily computed using classifiers. For 26 letters, there are two ways to do the classification, the first one is setting each letter to a specific number. For example, 789-A 766 -B based on UCI website(https://archive.ics.uci.edu/ml/machine-learning-databases/letter-recognition/letter-recognition.names)[3]. The second way is simpler, using binary case to differentiate 26 letters into two sets, A-M to 1, and N-Z to -1 (Caruana and Nicu,2006)[1]. I've implemented both ways to do the data-cleaning part.

2.3 Bank Marketing

For Bank Marketing dataset, there are 45211 data points and 17 features. The last feature `y` is the output which determine whether client subscribed a term deposit. For the rest of 7 features are numbers and 9 features are categorical. Again, I am using one-hot encoding to process the categorical features. As there are no missing data in this case. The dimension of dataset becomes 45211*58 after the encoding.

3 Method

The algorithms I am implementing in this to do the classification are K-nearest neighbors, Random Forest and Boosted Trees(BST-DT). Each of these method is implemented to work on each of the three datasets.

3.1 K-nearest neighbors

K-nearest neighbors is one of the most basic ways to implement classification. It is basically classify the current point based on the class of its closest K points. For a set of points,

$$\mathbf{S}{=}x_1, x_2, x_3 \ldots\ldots x_n$$

It assigns the label of K closest points to S. In handing the numeric features for KNN, I am setting 0 mean and normalize the dataset by subtracting each by average and divided by standard deviation.

3.2 Boosted Trees

For boosted decision tree, for each round, the classifier will adjust its weight by putting more weight on the misclassified point which helps the classifier correct the wrong point faster.I am using [2,4,8,16,32,64,128,256,512,1024,2048] boosting steps for this problem.

- Given a set of points

  $$((x_1, y_1), (x_2, y_2) \ldots\ldots (x_n, y_n)), y_i \in Y$$

  which Y = -1,1 Then do the update

  $$D_{t+1}(i) = \frac{D_t(i)exp(-\alpha_t y_i h_t x_i)}{Z_t}$$

  (L. Reyzin and R. E. Schapire 2006)[5]

From here

$$D_t(i)$$

stands for the a distribution of weights, which is varied each time to help with the more efficient classification

3.3 Random Forest

In terms of Random Forest, it has some similarities to Decision Tree. The difference is Random Forest is implemented only base on a set selection of random features. I am implementing Breiman-Cutler (Caruana and Nicu,2006)[1] according to yielded better results. The forests have 1024 trees. The size of the feature set considered at each split is 1,2,4,6,8,12,16 or 20.

4 Experiment

This table includes an implementation of three classification methods and three datasets.

dataset vs algorithm	KNN	Random Forest	Boosted Tree
adult(cleaned)	83.58%	85.44%	85.54%
adult(source)	83.09%	85.49%	85.53%
Letter(binary)	94.96%	94.43%	90.74%
Bank Market	87.38%	87.27%	87.44%
Letter(26)	89.73%	91.97%	70.79%

Of the three algorithms provided, Random Forest have a overall best accuracy compared to the other two. Since the datasets provided are all large datasets with lots of features. In this way, the random forest is balancing the error in various features and provides a more suitable model for these datasets. Thus, it creates highest accuracy. Comparing the accuracy of Source and Cleaned Dataset, only using Random Forest can increase the accuracy. This shows that in handling missing data, Random Forest is more efficient compared to KNN and Boosted Decision Tree. For KNN, as one of the most basic algorithms to do classification, is able to handle classification in a quite direct way. However, as for adult dataset, different number categories have different meaning, weight, so it is a bit harder to judge only based on distance. As a result, for complex dataset such as adult, KNN could only have a poor performance compared to Random Forest and Boosted Tree. When the dataset is simpler, say letter which each number feature is more similar, KNN can provide a much better performance. By doing 0 mean and normalizing the dataset, the accuracy improves 4% for adult dataset, 0.2% for Bank Market dataset, and 0.04% for Letter dataset. It seems that the more complex the numeric features of the dataset, the more helpful normalization helps with improving the KNN performance. For Boosted Tree, it maintains a not-bad prediction in datasets except Letter(not binary). Its accuracy for binary/not binary Letter Dataset differs about 20% because for output with only two categories, it is easy to implement with binary classification and boost the critical part. Also, boosted tree maintain a not bad performance in predicting dataset with missing data.

5 Conclusion

Although in terms of real-life utilization, the most advanced deep learning is one of the most helpful algorithm. In terms of Boosted Decision Tree, Random Forest and KNN classifiers, comparing the overall performance, Random Forest maintains overall best performance compared to the two other classifiers. For various datasets as Random Forest has its advantage in handing missing data as well as multiple classes. This result indeed provides us an insight in implementing Random Forest when the dataset is large and complex. Also, if we are sure that we aim to use binary classification for the output,boosted decision tree is also a good choice. For number categories are not hard enough, using KNN would be an option to provide an accurate prediction. Overall, different supervised algorithms have various advantages in terms of doing classifications, we need to determine optimal classifier based on the selection of dataset.

References

[1] Caruana and Niculescu-Mizil:An Empirical Comparison of Supervised Learning Algorithms, 2006.

[2] Sumit,Suhrid,Raghuraman:DLID: Deep Learning for Domain Adaptation by Interpolating between Domains, 2013.

[3] UCI Archive: http://archive.ics.uci.edu/ml/

[4] Leo Breiman and Adele Cutler, Retrived from: https://www.stat.berkeley.edu/ breiman/RandomForests.

[5] L. Reyzin and R. E. Schapire: How Boosting the Margin Can Also Boost Classifier Complexity,2006.