Model Comparisons in Machine Learning

Yuexiang Zhang
University of California, San Diego
yuz719@ucsd.edu

Abstract

There is a old saying "Unity is strength", meaning that the power of crowd is larger than the power of a single person. To investigate whether this old saying also applies to machine learning, I explore three ensemble learning classifiers: Bagging , Boosting and Random Forest. I compare these three classifiers with Decision Tree. I also compare these three classifiers mutually to find out which one is the best ensemble learning classifier.

Keywords- Ensemble learning, Bagging, Boosting, Random Forest

1. Introduction

Ensemble learning refers to a collection of methods which train several individual classifiers and combine their predictions. The purpose of ensemble learning is that when we need to make important decisions, we need to combine the results of many base classifiers into one learning model to make sure the final prediction is precise. Sometimes, the prediction of a single classifier is not accurate enough. There are many real-world examples to show the reason why we need ensemble learning classifiers. For instance, any professional academic competitions will invite several judges to give the final grade for one participant to make sure the final grade is accurate and fair. Furtherly, different combinations should have different prediction results. Therefore, it is also necessary to compare different ensemble learning classifiers. In my research, I compare Bagging, Boosting, and Random Forest these three ensemble learning classifiers. To compare their effects, I tune parameters through GridSearch to find their best parameters and then record the training, validation and testing accuracy for each classifier. To reduce the influence caused by dataset, I choose four datasets from UCI repository. To reduce randomness, I conduct three trials for each dataset and each classifier.

2. Methods

2.1. Bagging

Bagging is a simple ensemble learning algorithm but with excellent output. The bagging algorithm bases on bootstrap, which is a common method we use in statistic area. Bootstrap means that it draw different training subset from the whole dataset with replacement. Each of these datasets then is used to train a different classifier of the same type. For example, if we choose decision tree as the small classifier and set the parameter n_estimators to be 10, there will be 10 small decision trees. Then the bagging algorithm will draw 10 subsets from the whole datasets and then use one subset to train one small decision tree. After training each small classifier, the bagging algorithm then combine their seperate results by taking a simple majority vote of their decisions.

After knowing concepts after bagging, how to choose the basic classifier used in bagging. After doing some research, I know that bagging works best with unstable basic classifiers. Unstable classifiers refer to classifiers which produce differing generalization behavior with small changes to the training data. Unstable classifiers also refer to those with high variance. For example, Decision Trees and Neural Networks are this kind of classifier.

2.2. Boosting

Boosting is also a ensemble learning method which combines weak classifiers to obtain a strong classifier. In my research, I use Adaboosting which stands for adaptive boost. Right now, Adaboosting is the most popular boosting algorithm. The most important characteristic of Adaboosing is that it updates the weights of training data by improving the weights of wrong classified data. Finally, it combines these weak classifiers linearly to form a strong classifier.

Conceptually, each iteration of boosting creates three weak classifiers. First of all, it draw a subset randomly from the whole dataset and train the first classifier with this subset. Secondly, it choose the training set for the second classifier by picking a subset that only half of the data points in this subset are classified correctly by the first classifier. Thirdly, the third classifier is trained with instances on which the first and second classifier disagree. Finally, Adaboosting algorithm combine the results of these three weak classifiers by taking a majority vote.

2.3. Random Forest

Generally speaking, Random Forest is a special case of bagging. Bagging can use any kind of weak classifiers while Random Forest can only use tree. Right now, the most popular type of tree we use in Random Forest is CART tree. It is also the default tree type in Sklearn.

One key point of Random Forest is that when training each small decision tree inside this forest, it does not need to use all the features of one data point. Let's say we have M features in total for one data point, we only choose m features randomly to train one small tree. We have m¡¡M. Additionally, to train a small tree within Random Forest, we do not need pruning because the choosing procedure of data points and features can make sure its randomness. This randomness can reduce the possibility of overfitting considerably.

3. Experiment

3.1. Step 1 Choosing dataset and classifiers

To make my research result comprehensive and convincing, I choose three data sets with differnet number of features from UCI repository.

I choose two weak classifiers, Decision Tree and SVM with Liner Kernal. The reason why I choose Decision Tree is that I use Decison Tree as the small classifier which is used in the ensemble learning classifiers. If I do Decision Tree singly, it can clearly show the difference between Single weak classifier and ensemble learning classifiers. The reason why I choose SVM with Liner Kernal is that it is usually a high-accuracy weak classifier. I want to see whether it can outperform ensemble learning classifier sometimes.

3.2. Step 2 Tuning Parameters

To get the best accuracy for each classifier, I use GridSearchCV to tune the parameters. For weak classifiers, I tune the parameters directly with GridSearchCV because it is very straightforward. For ensemble learning classifiers such as Random Forest, I tune the parameter of Decision Tree first, which is the small classifier being used in Random Forest. Then I tune the large parameters which is directly related with Random Forse. By doing this, I can make sure that I can get the best parameters when training with this specific data set.

3.3. Step 3 Comparising

I compare training, validation, and testing accuracy of the five classifiers. I will show the comparison results in a table.

All the accuracies are averaged value after 3 trials.

Table 1: Data Set1 50/50 partition

Type	Train	Validation	Accuracy
Bagging	0.954	0.965	0.937
Boosting	1.0	0.965	0.972
RF	0.993	0.951	0.988
SVM(Linear Kernal)	0.916	0.926	0.933
Decision Tree	0.923	0.931	0.912

Table 2: Data Set1 80/20 partition

Type	Train	Validation	Accuracy
Bagging	0.972	0.955	0.947
Boosting	0.989	0.969	0.945
RF	0.978	0.960	0.956
SVM(Linear Kernal)	0.928	0.936	0.894
Decision Tree	0.939	0.917	0.906

Table 3: Data Set1 20/80 partition

Type	Train	Validation	Accuracy
Bagging	0.984	0.974	0.940
Boosting	1.0	0.982	0.936
RF	0.982	0.965	0.927
SVM(Linear Kernal)	0.969	0.960	0.890
Decision Tree	0.953	0.932	0.886

Table 4: Data Set2 50/50 partition

Type	Train	Validation	Accuracy
Bagging	0.895	0.747	0.752
Boosting	1.0	0.817	0.892
RF	0.995	0.885	0.812
SVM(Linear Kernal)	0.803	0.788	0.774
Decision Tree	0.810	0.686	0.695

To next page
To next page
To next page
To next page
To next page
To next page
To next page
To next page

Table 5: Data Set2 80/20 partition

Type	Train	Validation	Accuracy
Bagging	0.980	0.853	0.752
Boosting	1.0	0.866	0.732
RF	0.995	0.813	0.799
SVM(Linear Kernal)	0.822	0.747	0.756
Decision Tree	0.878	0.796	0.780

Table 6: Data Set2 20/80 partition

Type	Train	Validation	Accuracy
Bagging	0.866	0.814	0.762
Boosting	0.923	0.817	0.766
RF	0.990	0.810	0.791
SVM(Linear Kernal)	0.822	0.711	0.799
Decision Tree	0.890	0.796	0.812

Table 7: Data Set3 50/50 partition

Type	Train	Validation	Accuracy
Bagging	0.892	0.831	0.768
Boosting	0.945	0.902	0.890
RF	0.991	0.951	0.979
SVM(Linear Kernal)	0.792	0.713	0.704
Decision Tree	0.801	0.782	0.789

Table 8: Data Set3 80/20 partition

Type	Train	Validation	Accuracy
Bagging	0.912	0.857	0.780
Boosting	0.959	0.913	0.890
RF	0.988	0.943	0.879
SVM(Linear Kernal)	0.792	0.780	0.704
Decision Tree	0.790	0.724	0.769

Table 9: Data Set3 20/80 partition

Type	Train	Validation	Accuracy
Bagging	0.901	0.814	0.762
Boosting	0.923	0.817	0.766
RF	0.990	0.810	0.791
SVM(Linear Kernal)	0.722	0.711	0.759
Decision Tree	0.792	0.796	0.771

4. Conclusion

First of all, all the three data sets show that ensemble learning classifiers outperform single weak classifiers.

Secondly, based on the three data sets I choose, I find that among the three ensemble learning classifiers, random forest usually performs best. In most time, boosting works better than bagging. However, I know that it should not be absolute because of different data sets.

Thirdly, I find that as the features of data set increases, the differences between ensemble learning classifiers and single weak classifiers increases. The increasing features make ensemble learning classifiers outperform much more that weak classifiers.

Fourthly, I find that the larger the partition is, the higher the accuracy is. It means usually the training accuracy under 80/20 partition is higher that the training accuracy under 50/50 and 20/80 partition. This rule also applies to testing accuracy.

5. Bonus Points

I think I can get some bonus points. First of all, I not only investigate on weak classifiers but also do some research on ensemble learning classifiers. In total, I use 5 classifiers including 2 weak classifiers and 3 ensemble learning classifiers. Secondly, I spent a lot of time learning Latex to have a better format of my report. I highly value the ability of using Latex because I'm going to use it for a lot of times in the future. I'm so happy that COGS118A gives me motivations to learn how to use Latex.

6. Reference

[1] R. Caruana, "An empirical evaluation of supervised learning in high dimensions", Proceedings of the 25th International Conference of Machine Learning, pp. 96-103, 2008

[2] Zhuowen Tu, Lecture slides, University of California, San Diego, Cogs118A, 2019