Youliang Liu

COGS 118A

Final Project

03/21/2020

Abstract

In this report, I will explore the efficiency of 3 different classifiers, Decision Tree classifier, K Nearest Neighbor classifier, and Random Forest classifier. I will also explore the effect of these three classifiers when given different dataset magnitude and different training set and testing set magnitude. Overall, the random forest classifier is very stable and have a very low training error and testing error on average, but the downside is it takes the longest time to run. And the decision tree classifier is also quite stable, and its biggest advantage is it is very fast in runtime. K nearest neighbor on the other hand is the most unstable classifier. It demonstrated a dramatic increase in testing error when the magnitude of the dataset increased dramatically.

Introduction

Since each classifier is constructed very differently, they have very different properties which leads them to perform good under some circumstances, and bad under some other circumstances. Since the beginning of machines learning and with its rapid development over the years, many different classifiers have been created to solve different kinds of classification and regression problems. Since those classifiers have very different properties, it is critical to test out the performance of different classifiers. In the article '*An Empirical Comparison of Supervised Learning Algorithms*', the author Rich Caruana and Alexandru Niculescu-Mizil constructed a research on the performance of majority of the supervised learning classifiers and listed the average accuracies of those classifiers. Similarly, I constructed my own research on three popular classifiers, Decision tree classifier, K Nearest Neighbor classifier, and Random Forest classifier and compared their accuracies on three popular datasets. I also conducted

research in how the size of the training data and testing data will affect the result. I also performed some preprocessing on the dataset to make it less complicated and not overly biased to minimize the random effects which may occur due to the selection of the dataset.

Methods

Dataset:

All three datasets I used in my research are from the scikit-learn library. The reason I choose scikit-learn library datasets is that those datasets are being tested by many other researches and are proved to be less biased and easy to use. And I especially choose the datasets that have a very different size in order to test out the effect the size of the dataset will have on different classifiers. The first dataset I chose is the digits dataset. It has 64 dimensions and 1797 samples. Since it has a relatively large dimension comparing to the other two datasets, I performed the PCA feature extraction to downsize the dimension to 32. In this way the dataset is not overly complicated so that it will not affect the performance of the classifier due to the side effects of high dimensions. The second dataset I used is the breast cancer dataset. It is a small dataset with 569 samples and has a dimension of 30 for each sample data. This dataset is the only one that is originally a binary dataset, which means there are only 2 labels for all the sample data. The third dataset is about the house pricing in California. It has the biggest data size, which includes 20640 data samples with 8 dimensions for each sample. This dataset helps me to identify the effect of dataset's size on the performance of the classifiers.

Metric:

I used the training error and testing error to be the metric to evaluate my classifiers. I used the gridsearchcv to perform cross validation and tune the hyperparameters for my classifiers. Then I fit the training data by using the best estimator, and then calculate the training error and testing error base on the training data and testing data.

Classifiers:

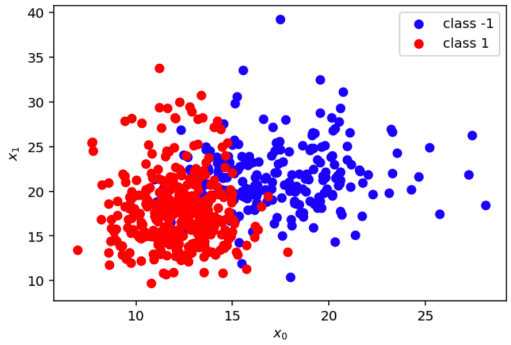
I used the Decision tree classifier, K nearest neighbor classifier, and the Random forest classifier as my classifiers. All three of them are imported from the scikit-learn package.

Experiment:

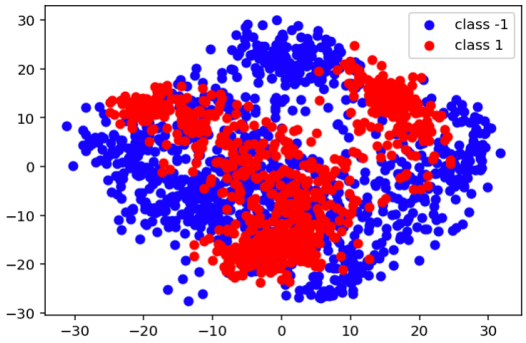
Label management:

Among the three datasets I used in my research, only the breast cancer dataset comes with a binary label. But the label for the breast cancer dataset is 0 and 1, so I changed to 0 label to be -1.

```
Visualizing breast_cancer data:
```

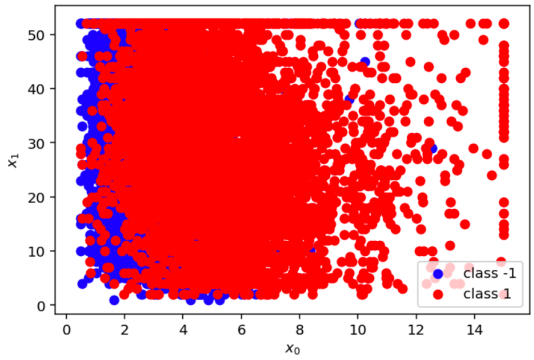


For the other two datasets I performed some tasks to transfer the labels to binary. For the digits dataset, the labels are from 0 to 9, which represents the 10 digits. I transformed all the label bigger than 5 to be 1 and all the labels smaller than 5 to -1 to make them binary. By managing the data in this way I can have an evenly distributed dataset.



Similarly, for the California housing datasets, the original labels are ranging from 0.15 to 5 representing the price of the house in unit of 100,000 dollars. I calculated the mean value of the price and rearrange the label to if the label is above the mean then I classify them as 1, if it is below the mean, then I classify them to -1.

Visualizing housing price data:



Data Splitting:

I split the dataset into training set and testing set 20/80 and 80/20. Which means 20% of the dataset to be training set and 80% of the data to testing set in the first splitting, and 80% of the dataset to training data and 20% to testing data in the second split.

Minimize randomness:

For each classifier and each splitting, I fit the training data three times after reshuffling the dataset and split again after each fit. Then I take the average of the training errors and testing errors of these three times to minimize any randomness that may occur during the training process.

Results:

Digits dataset

Classifier	**Decision Tree**		**Knn**		**Random Forest**	
Split	**20/80**	**80/20**	**20/80**	**80/20**	**20/80**	**80/20**
Training error	**0.046**	**0.077**	**0.007**	**0.002**	**0**	**0**
Testing error	**0.209**	**0.151**	**0.027**	**0.009**	**0.096**	**0.037**

Breast Cancer dataset

Classifier	**Decision Tree**	**Knn**	**Random Forest**

Split	**20/80**	**80/20**	**20/80**	**80/20**	**20/80**	**80/20**
Training error	**0.015**	**0.018**	**0.050**	**0.059**	**0.006**	**0.004**
Testing error	**0.074**	**0.051**	**0.074**	**0.056**	**0.054**	**0.032**

California Housing dataset

Classifier	**Decision Tree**		**Knn**		**Random Forest**	
Split	**20/80**	**80/20**	**20/80**	**80/20**	**20/80**	**80/20**
Training error	**0.152**	**0.154**	**0.249**	**0.217**	**0**	**0.003**
Testing error	**0.18**	**0.163**	**0.393**	**0.352**	**0.124**	**0.112**

Conclusion:

From the difference of training error and testing error in regard to splits, we can tell that 80/20 split has a smaller testing error on all cases. This result indicated that the increase of training dataset size can clearly train the classifier better and classify the testing data better.

Comparing three datasets, the California housing dataset has on average the highest training and testing error. That is caused by the large size of the data size. More training data is good, but when the size of the data exceeded certain range, they will have a negative effect on the classifier by causing overfitting.

Judging from the classifier perspective, the Random forest classifier is the most stable classifier which has a very low training error and testing error in three different datasets. The simultaneous increase of training error and testing error in all three classifiers for the large-scale datasets implies that that dataset might has a high variance and low bias which can leads to overfitting. And since Knn is especially sensitive to overfitting, Knn performed badly on the large scale dataset. The decision tree classifier also remained pretty stable throughout the classification process.

Reference:

Caruana, R., & Mizil, A. N. (2006). *An empirical comparison of supervised learning algorithms. In ICML '06 Proceedings of the 23rd international conference on Machine learning* (pp. 161-168). Pittsburgh, PA: Proceeding.