Comparison of Supervised Learning Algorithms

Junpeng Yao

Abstract

Due to the overwhelming amount of supervised learning algorithms, we always face the problem that what is the best algorithm to choose for a certain data set. Inspired by *An Empirical Comparison of Supervised Learning Algorithms* written by Caruana and Niculescu-Mizil, I conduct a comparison between three supervised learning algorithms: K-Nearest Neighbors, SVMs and random forests. For each of these three classifiers, I train and test it on three data set — Isis Data Set, Adult Data Set and Bank Data Set.

1. Introduction

In this research, I compare the classification accuracy of three supervised learning algorithms on three data sets. I chose K-Nearest Neighbors, SVMs and random forests. To test the classification accuracy exhaustively, I tested these three algorithms on three data sets and with three different partitions. Moreover, I obtain the best parameter for each algorithm on the certain data set. Besides finding out the difference in performance, I also generalize the best partition choice for all data sample.

2. Method

2.1 Learning Algorithms

Before diving into the actual experiment, I want to give a brief background of these algorithms. This section summarizes these three algorithms and the parameter we chose for each algorithm in our implementation.

K-Nearest Neighbors(KNN): KNN algorithm is one of the simplest supervised learning algorithms. It is very easy to implement and understand. It simply calculates the distance of a data point to all other training data. The distance I used in my experiment is Euclidean distance, but it can also be other distance such as Manhattan distance. It then selects the K-nearest “neighbors” and assigns the data point to the class which the majority of the K data points belong to. In my experiment, I use 20 values of K ranging from 1 to 20 and grid search cross-validation to find the best hyper-parameter K.

Support Vector Machine (SVM): A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. In my experiment, I use 5 values of C ranging from 10^-1 to 10^-5 and grid search cross-validation to find the best hyper-parameter C.

Random Forest(RF): Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It creates a forest and makes it somehow random. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. In my implementation, the size of the feature set considered at each split is 1,25,50,100 or 150.

2.2 Performance Metrics

In this research, I mainly use the classification accuracy as the metrics. Classification accuracy is defined as the percentage of correct prediction for a certain classifier. I first trained the classifier using the training data and grid search cross-validation method and then gained the proper parameter which can minimize the training error. Then, I calculated the classification accuracy by running such classifier on the test data and measuring the percentage of correct prediction. Such procedure was repeated for three times for each data set and each partition. Finally, I took the average of these three results and got the final classification accuracy for this classifier.

3. Experiment

In this experiment, I use three algorithms to run on three data sets. For each data set, I train and test the data using three different partitions: 80% training and 20% testing, 30 % training and 30% testing and 20% training and 80% testing. The following are three tables that summarize the result of each data set.

Iris Data Set	KNN	SVM	Random Forest	Average partition
80/20 Training	0.972	0.956	1.0	
80/20 Testing	**1.0**	**0.978**	**0.967**	**0.981**
50/50 Training	0.964	0.964	1.0	
50/50 Testing	**0.964**	**0.956**	**0.942**	**0.954**
20/80 Training	0.967	0.911	1.0	
20/80 Testing	**0.947**	**0.869**	**0.953**	**0.923**
Average Testing	**0.970**	**0.934**	**0.954**	
Best Parameter	**K=10**	**C=0.1**	**N=1**	

Adult Data Set	KNN	SVM	Random Forest	Average partition
80/20 Training	0.805	0.424	0.988	
80/20 Testing	**0.797**	**0.413**	**0.849**	**0.686**
50/50 Training	0.797	0.416	0.989	
50/50 Testing	**0.779**	**0.424**	**0.836**	**0.679**
20/80 Training	0.806	0.426	0.993	
20/80 Testing	**0.766**	**0.413**	**0.829**	**0.669**
Average Testing	**0.780**	**0.416**	**0.838**	
Best Parameter	**K=6**	**C=0.001**	**N=150**	

Bank Data Set	KNN	SVM	Random Forest	Average partition
80/20 Training	0.973	0.969	0.998	
80/20 Testing	**0.976**	**0.972**	**0.983**	**0.977**
50/50 Training	0.977	0.972	0.998	
50/50 Testing	**0.972**	**0.970**	**0.978**	**0.973**
20/80 Training	0.981	0.978	0.998	
20/80 Testing	**0.972**	**0.971**	**0.973**	**0.972**
Average Testing	**0.973**	**0.971**	**0.978**	
Best Parameter	**K=6**	**C=0.0001**	**N=25**	

For the Iris data set, I found out that in average KNN has 97.0% test accuracy while random forest and SVM have 95.4% and 93.4% respectively. For Adult data set, random forest in average has 83.8% test accuracy, but KNN and SVM has 78.0% and 41.6% respectively. For bank data set, random forest also performs better than the other two algorithms. Therefore, overall the random forest still has the best performance in terms of the classification accuracy, which is consistent with the result of Caruana and Niculescu-Mizil.

By partitioning the data using three different ratios, we can compare the result and find out the best ratio between the number of training and test data. For the Isis data set, 80/20 partition in average performs 98.1% accuracy, while the other two partitions perform slightly worse than it. In fact, the result that 80/20 partition surpasses the other two partitions is consistent in all these three data sets. On the other hand, the partition 20/80 always performs the worst among these three partitions. Therefore, we can conclude that there is an increase of test accuracy with more training data and less test data.

4. Conclusion

In this study, I compare three algorithms using three different large data sets. The result I found is that random forest performs the best in the terms of classification accuracy and SVM performs slightly worse. Also, the partition where training data takes 80% and test data takes 20% has the best accuracy in every case. One interesting thing I notice is that SVM performs very badly on the Adult data set, in which the accuracy is less than 50%. I think it is because the parameter that was generated is not the best choice and linear SVM is not very stable for such a large data set.

Work Cited

Caruana, Rich, and Alexandru Niculescu-Mizil. “An Empirical Comparison of Supervised Learning Algorithms.” *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, 2006.