An Empirical Study of Three Modern Machine Learning Algorithms

Yijie Fan YIF063@UCSD.EDU
University of California San Diego
9500 Gilman Drive
La Jolla, CA 92093, USA

Abstract

With the development of modern machine learning, many new algorithms are created and applied to many areas. One important thing in application is how to decide which algorithm to choose in different situations. In this paper, we mainly try to compare the effectiveness of three algorithms: k-nearest neighbors (KNN), Support Vector Machine (SVM) and Random Forest. We test three algorithms on three different datasets with different partitions of training and testing samples to evaluate the power of three algorithms.

1 Introduction

There have been many empirical studies evaluating effectiveness of machine learning algorithms, but since the development in this area is fast, and lots of new algorithms are invented, there is a need to do some empirical studies on algorithms emerged recently. Learning algorithms are widely used in many areas such as financial trading, data security and health care. Such evaluations would be helpful for users who are unfamiliar with these algorithms yet would like to apply them to some situations.

This paper presents results of three machine learning algorithms on three datasets. We evaluate the performances of KNN, SVM with radical basis function kernel and Random Forest. We use only accuracy to compare the classification results of three algorithms. Before applying three algorithms to classify data, we use grid search method to find the optimal hyperparameters of each algorithm.

It turns out the result somewhat coincides with the result in the known study (Caruana R., & Neculescu-Mizil A., 2006) that Random Forest usually gives out the best prediction. One thing is that, unlike the empirical results from the past, SVM with radical basis function kernel does not seem to outperform KNN method. In fact, SVM method fails to perform a decent job in one dataset we use. However, it does not mean that SVM with rbf kernel is not a reliable algorithm, since it does give out decent results on the other two datasets. It only indicates that we need to be alerted not to use SVM abusively without evidence showing that SVM suits the data set we would like to classify well.

2 Methodology

2.1 Learning Algorithms

We try to explore the optimal parameter of each algorithm within the constraint of our computational ability. This section summarizes the machine learning algorithms we use as well as parameters we try.

KNN: We use 18 values of K ranging from K=1 to K=18. We use KNN with unweighted Euclidean distance to calculate the distance between any pair of training samples.

You should use Times Roman style fonts. Please be very careful not to use nonstandard or unusual fonts in the paper. Including such fonts will cause problems for many printers.

SVM: We use the kernel with radical basis function and with width {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2}.
Random Forest (RF): The number of trees in the forest we tried are 1,2,4,6,8,12,16,20. We would like to check how many trees would give out the best classification.

2.2 Performance Metrics

We use only the accuracy (ACC) to determine the effectiveness of the machine learning algorithm. We choose ACC because for the classification problem, the most important thing is how well our algorithm classifies the dataset. It is both easy to calculate and straight forward. Thus, for the simplicity, we just choose ACC as our performance metric in this case.

2.3 Data Sets

We compare the algorithms on three binary classification problems. IRIS, WINE and CAR EVALUATION are from the UCI Repository (Blake & Merz, 1998). IRIS has been converted to a binary problem by treating "Iris-Setosa" as "1" and other kinds of flowers as "0". We converted CAR EVALUATION into binary problem by changing conditions "unacc" and "acc" to "0" and conditions "good" and "vgood" to "1", meaning we only would like to consider cars that are at least "good" to be buyable. In addition, we use LabelEncoder to apply one hot coding method to CAR EVALUATION data set since it consists of non-numeric variables. For the data set WINE, we convert it to a binary problem by denote wines from location 1 as "1" while wines from other places as "0".

3 Experiment

For each data set, we partition it by three ways: 20% training + 80% testing; 50%training + 50%testing and 80% training + 20% testing. We first use Grid Search to find the best parameters of each model, and then train the model, from which we get a training accuracy. Then, we do 3-fold cross validation to verify our models and get a validation accuracy. Eventually, we use our test data to gain a test accuracy and use it to see if our models work well for all data. To prevent extreme cases, we repeat the procedure for three times, and average over three trials to get a more reliable accuracy. We would like to run more trials, but this is a computational expensive set of experiments. Fortunately, even with only three trials, we are able to find differences between the results given by three algorithms.

For KNN method, when we use K=1 or 4 gives out the best classification result. For SVM, the best parameters are different for different data sets. For CAR EVALUATION and WINE, the best parameter is 0.001 while the best parameter for IRIS data set is 0.05. As for the Random Forest method, the best parameter for CAR EVALUATION and WINE is mostly 12 or 20 trees, while the best parameter for IRIS is just 1 tree.

The tables below show the training accuracy, validation accuracy and testing accuracy for three algorithms and three data sets. By looking through the accuracy we get, we notice several interesting things. One thing is that, when number of training samples increases, we are likely to get a better classification result, which does not surprise us. From CAR EVALUTION data set, we can see that Random Forest gives out higher accuracy than KNN and SVM, while KNN is better than SVM for some cases but worse in some other cases. If we look at the WINE data set, Random Forest still gives out the best testing accuracy except for 20% training and 80% testing case. However, SVM gives out poor results in this case. As for the IRIS data set, both SVM and KNN give out perfect accuracy in all cases while Random Forest only gets perfect result when training set is 80% of the total data set, while it makes few mistakes in some cases.

Car Evaluation		80% training	50% training	20% training
KNN	Training	0.988	0.977	0.959
	Validation	0.967	0.956	0.957
	Testing	0.965	0.944	0.928
SVM	Training	0.972	0.972	0.948
	Validation	0.965	0.957	0.948
	Testing	0.977	0.958	0.916
Random Forest	Training	1	1	0.996
	Validation	0.979	0.968	0.948
	Testing	0.981	0.977	0.943

Table 1 Car Evaluation

Wine		80% training	50% training	20% training
KNN	Training	1	0.966	1
	Validation	0.923	0.943	0.914
	Testing	0.917	0.921	0.902
SVM	Training	0.683	0.618	0.714
	Validation	0.683	0.618	0.716
	Testing	0.611	0.719	0.657
Random Forest	Training	0.998	1	0.962
	Validation	0.967	0.981	0.927
	Testing	0.963	0.955	0.881

Table2 Wine

Iris		80% training	50% training	20% training
KNN	Training	1	1	1
	Validation	1	1	1
	Testing	1	1	1
SVM	Training	1	1	1
	Validation	1	1	1
	Testing	1	1	1
Random Forest	Training	1	0.982	1
	Validation	1	1	1
	Testing	1	0.964	0.983

Table3 Iris

4 Conclusion

According to the excellent performances of Random Forest on the three data sets above, it is the best estimator among the three machine learning algorithms above. However, when we are using Random Forest, we need to try to provide it with as many data as possible since for some cases that data are limited, its performance is outperformed by KNN method. In addition, it actually performs the worst for the IRIS data set. The reason is that IRIS data set is almost the easiest data set to classify such that almost every algorithm could classify it nearly perfect. Therefore, the advantage of Random Forest is not obvious in this case. As for KNN and SVM, from the data sets we have, KNN seems to perform better than SVM, which is inconsistent with the result from the paper of Caruana and Niculescu-Mizil, possibly because the data sets we use and the insufficient parameters we test.

References

Caruana, R., Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. ICML '06 Proceedings of the 23rd international conference on Machine learning, 161-168.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases.