An Empirical Comparison of Supervised Learning Algorithms

Shuibenyang Yuan shy166@ucsd.edu
Department of Mathematics, University of California, San Diego
Halıcıoğlu Data Science Institute, University of California, San Diego

Abstract

In machine learning field, there are many supervised learning methods have been introduced and applied into industrial fields. In this paper, we will present an empirical comparison among three supervised learning methods to get some ideas of methods we have in the machine learning field. That is, Random Forests, Logistic Regression, and Decision Tree. We compare the performance of each by using three different datasets through three trials to analysis which method performs better under what circumstance.

1. Introduction

In the last decade, there were significantly large amount of different kind of machine learning methods being popular because of the increase demand of machine learning techniques upon different fields. While, there are always a trade-off among the methods we introduced, Thus, model selection procedure is very necessary when we do machine learning to receive the best prediction results. In this paper, we will use empirical study as a way to get better understanding the difference of different methods.

This paper will use three different supervised learning methods by the performance of their prediction accuracy (ACC). We will evaluate Random Forests, Logistic Regression, and Decision Tree on three binary classification problems for three trials to gain the average accuracy scores. The accuracy scores are: mean training accuracy, mean validation accuracy, and mean testing accuracy to understand the different performance of each supervised machine learning method at different scenarios.

The empirical results are supervising. Preprocessing and performing feature engineering the dataset by cleaning the missing values and standardizing the dataset improve the accuracy for prediction for all methods significantly. To preview: with preprocessed dataset, Random Forests and Logistic Regression have the same level performance, while Random Forests performs a bit better. Almost all cases, Random Forests outperforms than Logistic Regression with significantly higher training accuracy and 0.01 more accuracy of validation accuracy and training accuracy. Decision Tree has a lower level performance because of the overfitting of Unpruned Decision Tree. The best parameter that we found for each classifier using cross-validation is almost consistent for three datasets. Also, with more samples involved, the higher performance of each model.

2. Methodology

2.1 Algorithms and Time Complexity

We are using scikit-learn as our implement of machine learning methods. Due to the advanced data structure of scikit-learn's Random Forests, Logistic Regression, and Decision Tree and the optimization of numpy and scipy package, each machine learning procedure is computationally feasible without sampling the datasets. This section summarizes the parameters used for each learning algorithm, and may safely be skipped by readers who are easily bored.

Random Forests (RF):
We use the default implementation of scikit-learn *'Gini impurity'* as our method of Random Forests. The forest have 10 trees. The size of the feature set considered at each split is 2, 4, 6, 8, 12, 16 and 20.

Logistic Regression (LOGREG):
Our Logistic Regression uses L2 penalty with primal formulation. We train our model with varing the ridge (regularization) parameter by factors of 10 from $10^{-4}$to 10 10^{8}.

Decision Tree (DT):
The pruning option for decision tree we use is the max feature for each dataset. We use 6 values of max feature ranging from 10 to the dimension of the data set.

2.2 Performance Metrics

We use accuracy (ACC), one of the threshold metrics, as our criteria to measure the performance of the classifiers. The accuracy we use contains train accuracy, validation accuracy, and test accuracy. And we use test accuracy to rank classifiers on each dataset. For each accuracy, we compute it by counting the numbers of the successfully predicted labels and divide it by the total number of data points in the train/validation/test set.

The ACC has a fixed threshold 0.5. The train accuracy reflects how well the supervised method fits the dataset. If we only look at the train accuracy, it will not tell us the performance of each model in predicting labels. Thus, we add both validation accuracy and testing accuracy to see the predictability for each classifier.

2.3 Data Sets

We compare the algorithms on 3 binary classification problems. All of them are from UCI Repository (Blake & Merz, 1998). They are ADULT, CARD_DEFAULT, and MUSHROOM. We first clean the data by dropping all data points with null value. After that, we find that all of them contain nominal attributes, thus we need to convert them into Boolean using one-hot encoding method. In ADULT, we treat people who can earn more than 50k salary as 1, and who earn less or equal to 50k as 0. We also normalize the data in order to compute the relative Euclidean distance from the decision boundary.

3. Performance by ACC

For ADULT and CARD_DEFAULT, since the implement of algorithm in sikit-learn is optimized enough, we will not sample the data points to perform the study. We preprocessed by cleaning the null value first and scale the data set as standardized centered at the mean. For each of the data set, we use three different kinds of partitions, 20% training and 80% testing, 50% training and 50% testing, and 80% training and 20% testing. For each hyper parameter, we use 5-fold cross validation on the train set to fine-tune our model. For all the procedure above, we also try for three times in order to eliminate any biasness or unintentional mistakes. We then average the mean accuracy for train, validation and test and put them into Table 1 for each different partition.

The data for average performance on three classification for three different partitions is included in Table 1. Several things we found interesting in the ACC data. Since our dataset is very large, we can see a sign of convergence to some accuracy points among each classifier. Based on Table 1, we observe that Random Forests outperformed all the classifier, and Logistic Regression is in the second place. Decision Tree, on the other hand, have a significantly noticed phenomenon of overfitting: training accuracies are nearly to 1 to all the dataset, but validation accuracy and testing accuracy are in the last place. It shows that the best decision tree will cause overfitting effect among all three datasets, and if given more data in the training set, the lower the testing accuracy will be, which means that overfitting issue is even worse. To resolve overfitting, we can try decision tree in a dataset of smaller sample size to prevent overfitting by observing the trend of learning rate. Or alternatively, we can perform a post prune based on the validation set. Random Forests outperform the Logistic Regression with a better training accuracy because of the advantage of its decision tree based, it also has a bit higher performance in validation dataset and testing dataset.

Table1 Average Performance on Three Classifiers for three different Partitions

(Train/Test) Classifiers	Mean train acc	Mean validation acc	Mean test acc
80/20 RF	0.93046667	0.88556667	0.8862
50/50 RF	0.9388	0.88866667	0.88736667
20/80 RF	0.93143333	0.88866667	0.88933333
80/20 LOGREG	0.88746667	0.88306667	0.8837
50/50 LOGREG	0.88706667	0.88526667	0.885
20/80 LOGREG	0.8869	0.8856	0.88626667
80/20 BST-DT	0.99993333	0.84586667	0.8419
50/50 BST-DT	0.9999	0.84706667	0.8441
20/80 BST-DT	0.99986667	0.84706667	0.84463333

4. Performance by Datasets

We first use Grid Search with 5 cv to find the best parameter of the classifier. The results are posted in Table 2. From Table2, we observe that the parameters do not keep consistent. All performances for all classifiers for all datasets are included in Table 3. Observing the Table 3, we can see Decision tree dominantly have high accuracy in training accuracy. It is because of overfitting property of unpruned trees as mentioned in section 3. Then we should no consider it as a good estimator compared to others because of low performance of validation accuracy and training accuracy of decision tree: it constantly. While, comparison of Random Forests and Logistic Regression is very interesting. Both Random Forests and Logistic Regression perform very well with Random Forests won with a bit accuracy in general as mentioned in section 3. From Table 3, we can see that they are very close with a minor distance: Random Forests always wins with a less than 0.05 accuracy. While Logistic Regression took relatively longer time than Random Forests when we were training the model. We suggest the lower performance of Logistic Regression is because of the disadvantage of binary dataset of Logistic Regression. Random Forest, on the other hand, runs faster if it has 10 trees in our case because of the advantage of tree data structure. We should consider Random Forests be the best estimator in our comparisons.

Table 2 the best parameter over different Partitions

(Train/Test) Classifier	ADULT best_params	CARD DEFAULT best_params	MUSHROOM best_params
80/20 RF	{'min_samples_split': [20]}	{'min_samples_split': [20, 12]}	{'min_samples_split': [4, 6, 2]}
50/50 RF	{'min_samples_split': [20, 16]}	{'min_samples_split': [16, 8, 12]}	{'min_samples_split': [2, 16]}
20/80 RF	{'min_samples_split': [16, 20]}	{'min_samples_split': [16, 20]}	{'min_samples_split': [2, 4]}
80/20 LOGREG	{'C': [10.0, 1.0, 0.1]}	{'C': [0.001, 0.1]}	{'C': [0.01, 0.1]}
50/50 LOGREG	{'C': [10.0, 1.0, 0.1]}	{'C': [0.0001, 1.0, 10.0]}	{'C': [0.01]}
20/80 LOGREG	{'C': [1.0, 0.1]}	{'C': [1.0, 0.1]}	{'C': [0.01, 0.1]}
80/20 BST-DT	{'max_features': [70, 50, 30]}	{'max_features': [30, 70]}	{'max_features': [30]}
50/50 BST-DT	{'max_features': [30, 50]}	{'max_features': [30, 70]}	{'max_features': [50, 10]}
20/80 BST-DT	{'max_features': [50, 30]}	{'max_features': [70, 10]}	{'max_features': [10, 30]}

Table 3 Performance on Three Classifiers for three different Partitions

(Train/Test) Classifier	Adult mean train acc	Adult mean validation acc	Adult mean test acc	Card Default mean train acc	Card Default mean validation acc	Card Default mean test acc	Mushroom mean train acc	Mushroom mean validation acc	Mushroom mean test acc
80/20 RF	0.9023	0.8493	0.8495	0.8892	0.8077	0.8104	0.9999	0.9997	0.9987
50/50 RF	0.9065	0.8555	0.8527	0.9099	0.8105	0.8094	1	1	1
20/80 RF	0.9088	0.8543	0.8537	0.8855	0.8117	0.8143	1	1	1
80/20 LOGREG	0.85	0.8418	0.8447	0.8125	0.808	0.8083	0.9999	0.9994	0.9981

50/50 LOGREG	0.8504	0.847	0.8475	0.8109	0.8094	0.8086	0.9999	0.9994	0.9989
20/80 LOGREG	0.8497	0.8475	0.847	0.8111	0.8097	0.8124	0.9999	0.9996	0.9994
80/20 BST-DT	1	0.8061	0.8024	0.9998	0.7318	0.7239	1	0.9997	0.9994
50/50 BST-DT	1	0.8085	0.8054	0.9997	0.7329	0.7277	1	0.9998	0.9992
20/80 BST-DT	1	0.8122	0.8078	0.9996	0.729	0.7261	1	1	1

5. Conclusion

After being developed for several decades, we can see from the performance of all three classifiers that they are all well-developed with average testing accuracy around 0.8. We have to carefully pick the model and choose the correct hyper parameter to fine-tune in order to reach the highest prediction results. Some models are also sensitive to the preprocessing process in the dataset, while some others do not. In our study of Random Forests, Logistic Regression, and Decision Tree, under the scenario that the datasets are binary and categorical. we observed that Random Forests and Logistic Regression generally perform better than Decision Tree (unpruned), and Random Forests wins Logistic Regression with a bit higher accuracy without costing extra time.

6. Related Work

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

R. Caruana, A. Niculescu-Mizeil, *An Empirical Comparison of Supervised Learning Algorithms*, 2008

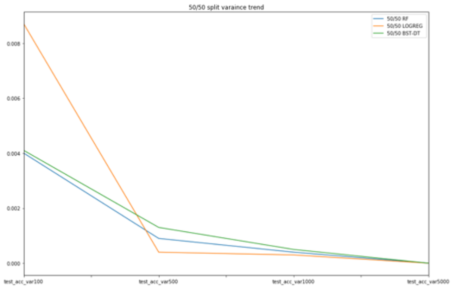
R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. LIBLINEAR: a library for large linear classification. The Journal of Machine Learning Research, 9:1871–1874, 2008.

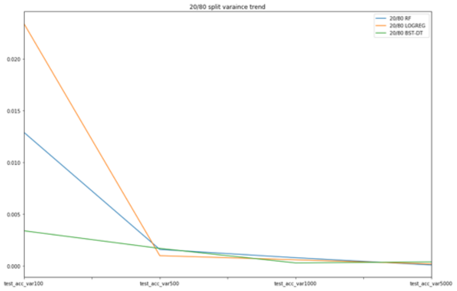
T. K. Ho, "*Random decision forests,*" in International Conference on Document Analysis and Recognition, pp. 278–282, 1995.

T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning: data mining, inference and prediction,* Springer, 2001

7. Bonus Points

We are also interested in how fast the variance of the training accuracy converge to 0. We sample the datasets as 100, 500, 1000, and 5000 samples.





Based on the plots and the table, we observe that Logistic Regression converges fastest when more and more data feed in, it has largest max variance in 50/50 and 20/80. In long term with large dataset, we can rank by variance with 1. Logistic Regression, 2. Random Forest, and 3. Decision Tree. In short term with small dataset, we can rank by variance with 1. Decision Tree, 2. Random Forest, and 3. Logistic Regression. We can conclude that Logistic Regression needs more data to get lower variance, while Decision tree does not acquire much data to get lower variance.

Table 4 Variance of training accuracy and test accuracy among different classifier over different partition

	train acc var100	test acc var100	train acc var500	test acc var500	train acc var1000	test acc var1000	train acc var5000	test acc var5000
80/20 RF	0.0044	0.0058	0.001	0.0008	0.003	0.0001	0.0021	0
50/50 RF	0.0061	0.004	0.0036	0.0009	0.0011	0.0004	0.0007	0
20/80 RF	0.0006	0.0129	0.0003	0.0016	0.0018	0.0008	0.0002	0.0001
80/20 LOGREG	0.0018	0.0029	0.0033	0.0007	0.0002	0.0001	0.0001	0
50/50 LOGREG	0.0032	0.0087	0.0018	0.0004	0.0002	0.0003	0	0
20/80 LOGREG	0.003	0.0234	0.0001	0.001	0	0.0006	0	0.0002
80/20 BST-DT	0	0.0042	0	0.001	0	0.0008	0	0.0002
50/50 BST-DT	0	0.0041	0	0.0013	0	0.0005	0	0
20/80 BST-DT	0	0.0034	0	0.0017	0	0.0003	0	0.0004