|  |
| --- |
| **An Empirical Comparison of Supervised Classification Methods** |

**Zichao Wu**

Department of Computer Science

University of California, San Diego

9500 Gilman Dr, La Jolla, CA 92093

[*ziw255@ucsd.edu*](mailto:ziw255@ucsd.edu)

**Abstract**

A number of supervised learning methods have been introduced in the last decade and used extensively to solve classification problems in empirical contexts. This paper compares the performancesof different supervised classifiers on different empirical datasets. Classifiers in this comparison are KNN, linear SVM, logistic regression, and random forest. This paper also examines influences of different partition methods on classifier performance mentioned above.

**1 Introduction**

Learning algorithms are now used in many domains to build models that could give reliable predictions on given datasets. The qualities of models being trained are frequently examined via training and validation accuracies that determines model/hyper-parameter selection, while qualities of trained models are primarily measured in terms of their test accuracies.

This paper attempts to partially replicate the results obtained in Caruana’s paper [1] with the same dataset from UCL machine learning repository [2]. We evaluate the performance of KNN, linear SVM, logistic regression, and random forest using the accuracy metric. The accuracies are obtained on training, validation, and test sets.

**2 Methodology**

**2.1 Learning Algorithms**

**Linear SVM**: sklearn implementation of linear SVM classifier is used. Scaling is done on each input dataset prior to training to shorten training time. We also vary the regularization parameter by factors of ten from 10−7 to 103 with each kernel.

**KNN**: we use 26 values of K ranging from K = 1 to K = |trainset|. We use KNN with Euclidean distance.

**Logistic Regression**: we train regularized regression classifiers, varying the ridge (regularization) parameter by factors of 10 from 10−8 to 104.

**Random Forests**: sklearn implementation of random forest classifier is used. The forests have 200 trees. The size of the feature set considered at each split is 1,2,4,6,8,12,16 or 20.

**2.2 Data Sets**

We compare the algorithms on 3 binary classification problems. There are 5000 randomly pre-selected cases for each problem. ADULT, COV TYPE and LETTER are from the UCI Repository [2]. ADULT dataset is converted to a binary problem based on the last column of adult income: rows with “>=50K” is assigned positive while “<50K” is assigned negative. COV TYPE has been converted to a binary problem by treating the largest class as the positive and the rest as negative. LETTER uses letters A-M as positives and the rest as negatives, yielding a well-balanced problem. All categorical attributes had been one-hot encoded, increasing the number of features, particularly for the COV TYPE dataset.

**2.3 Partitions**

We varied the test dataset size for each classifier for each problem: 20%, 50%, and 80% of each input dataset are assigned training set, with the rest being test set, before each experiment.

**3 Experiment**

After splitting the dataset into training and test sets using method as described in 2.3 in each experiment, we use 3-fold cross validation on the training set. To avoid randomness of splitting dataset, each partitioning method is performed 3 times, yielding 3 trials × 4 classifiers × 3 datasets × 3 partitions = 108 experiments in total. Each time we always report the best accuracy under the chosen hyper-parameter. For each problem and dataset split we find the best parameter settings for each algorithm. Since for the accuracy is averaged among three 3 trials to rank order the classifiers, we report 4 classifiers × 3 datasets × 3 test/train partitions (20/80, 50/50, 80/20) × 3 accuracies(train, validation, test) = 108 accuracy values.

The raw average training, validation, testing accuracy scores are given in table 1, table 2, table 3 below, respectively. The higher the accuracy score, the better the model performance on given test sets. Ranking of each classifier is given in table 4. In a given column, the higher the ranking score, the superior the classifier is relative to other in terms of prediction performance.

Table 1. Training accuracies of different partitions on each problem set across classifiers

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dset**  **Clf** | **ADULT** | | | **COVTYPE** | | | **LETTER** | | |
| **20%** | **50%** | **80%** | **20%** | **50%** | **80%** | **20%** | **50%** | **80%** |
| **KNN** | 0.754 | 0.756 | 0.764 | 1 | 1 | 1 | 1 | 1 | 1 |
| **Lin.SVM** | 0.854 | 0.860 | 0.869 | 0.754 | 0.753 | 0.768 | 0.740 | 0.741 | 0.759 |
| **Lin.Reg** | 0.789 | 0.792 | 0.795 | 0.751 | 0.751 | 0.767 | 0.739 | 0.743 | 0.761 |
| **RF** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 2. Validation accuracies of different partitions on each problem set across classifiers

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dset**  **Clf** | **ADULT** | | | **COVTYPE** | | | **LETTER** | | |
| **20%** | **50%** | **80%** | **20%** | **50%** | **80%** | **20%** | **50%** | **80%** |
| **KNN** | 0.754 | 0.755 | 0.762 | 0.736 | 0.711 | 0.670 | 0.932 | 0.915 | 0.857 |
| **Lin.SVM** | 0.841 | 0.840 | 0.822 | 0.749 | 0.747 | 0.735 | 0.735 | 0.738 | 0.753 |
| **Lin.Reg** | 0.789 | 0.791 | 0.794 | 0.748 | 0.745 | 0.737 | 0.735 | 0.739 | 0.747 |
| **RF** | 0.845 | 0.849 | 0.839 | 0.807 | 0.785 | 0.760 | 0.926 | 0.910 | 0.862 |

Table 3. Testing accuracies of different partitions on each problem set across classifiers

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dset**  **Clf** | **ADULT** | | | **COVTYPE** | | | **LETTER** | | |
| **20%** | **50%** | **80%** | **20%** | **50%** | **80%** | **20%** | **50%** | **80%** |
| **KNN** | 0.757 | 0.755 | 0.754 | 0.783 | 0.733 | 0.695 | 0.951 | 0.937 | 0.885 |
| **Lin.SVM** | 0.856 | 0.842 | 0.836 | 0.739 | 0.739 | 0.735 | 0.737 | 0.737 | 0.732 |
| **Lin.Reg** | 0.796 | 0.791 | 0.788 | 0.737 | 0.737 | 0.736 | 0.742 | 0.740 | 0.732 |
| **RF** | 0.862 | 0.847 | 0.838 | 0.811 | 0.800 | 0.771 | 0.936 | 0.920 | 0.885 |

Table 4. Ranking of Testing accuracies of classifiers across datasets

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dset**  **Clf** | **ADULT** | | | **COVTYPE** | | | **LETTER** | | | **SUM** |
| **20%** | **50%** | **80%** | **20%** | **50%** | **80%** | **20%** | **50%** | **80%** |
| **KNN** | 0 | 0 | 0 | 2 | 0 | 0 | 3 | 3 | 3 | 11 |
| **Lin.SVM** | 2 | 2 | 2 | 1 | 2 | 1 | 0 | 0 | 1 | 11 |
| **Lin.Reg** | 1 | 1 | 1 | 0 | 1 | 2 | 1 | 1 | 0 | 8 |
| **RF** | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 24 |

**3.1 Interpretation of test results: classifiers**

By looking at the ranking table (table 4), it is easy to see that among all the classifiers, random forest has the best general accuracy (the highest sum of ranking scores across all models and partitions). Linear SVM is on par with KNN, and linear regression has the worst performance. Additionally, we can see from table 1 that Random Forest almost always has overfitting issues with all training datasets, so does KNN except the ADULT set.

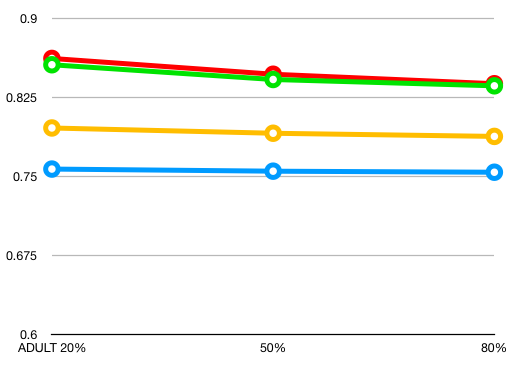
For the ADULT dataset, there is a clear ranking among all the classifier performance: RF > Linear SVM > Linear Regression > KNN. For the COV TYPE dataset, RF still has the best performance across all partition methods. However, performance of the other three classifiers varies with partition method. KNN performs well on 20% test size but not on larger test sizes. Linear regression and linear SVM have very similar performance in COV TYPE set. For the LETTER dataset, it is worth noting that, unlike the other two datasets, KNN beats all other 3 classifiers across every partition method, with random forest being the second. It could be due to the fact that the LETTER dataset has significantly smaller number of features than the other two datasets after being one-hot encoded, in which case overfitting would be less of an issue. Similar to the COV TYPE set, linear regression and linear svm have very similar performance in LETTER set (accuracy averages around 0.73 - 0.74).

**3.2 Interpretation of test results: partition size**

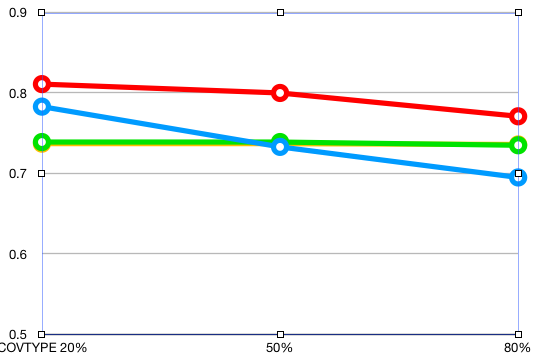
All classifiers accuracies show a very mild decreasing trend of test accuracy with the increase of test set size/decrease of training set size (graph 1-3). Among all the classifiers, Linear regression (yellow line) show the least variability in response to change in test size. Linear SVM (green line) is very close to linear regression, though its accuracy shows a greater variability in response to decrease in test size in ADULT set.



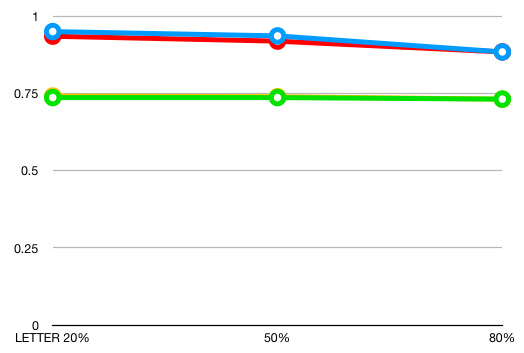
legend



Graph 1: Negative relationship, ADULT



Graph 2: Negative relationship, COVTYPE

****

Graph 2: Negative relationship, LETTER

**4 Conclusion**

Among all the classifiers presented above, Random forest classifier performs the best across all experiments. KNN excels in LETTER dataset, while performs worst in all others. Both KNN and random forest show a tendency to overfit the data. Linear regression and svm show a very similar testing accuracy scores across all experiments, with Linear SVM better in ADULT experiment. In general, RF performs the best, followed by Linear SVM, linear regression, and KNN. It is worth noting that even the best model RF could give lower accuracy score than KNN in experiments with small feature size (LETER set).

**5 Bonus**

Video Link in Github:

<https://github.com/CharryWu/COGS118Final>

**References**

[1] Caruana, R., & Niculescu-Mizil A. (2005). An Empirical Comparison of Supervised Learning Algorithms.

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