Empirical Comparisons of Supervised Learning Algorithms

Lulude Sun L1SUN@UCSD.EDU
Department of Cognitive Science
University of California, San Diego
San Diego, CA 92093, USA

Editor: Lulude Sun

Abstract

In the paper *An Empirical Comparison of Supervised Learning Algorithms* by Caruana and Niculescu-Mizil, they implemented and compared each of the existing supervised learning algorithms on different datasets. They presented a large-scaled empirical comparison between these supervised learning methods including SVMs, neural nets, logistic regression, naive bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees, and boosted stumps. Furthermore, they examined their performances and evaluated these learning methods by using a variety of performance criteria such as T-test. In this paper, I will follow a similar procedure and format where I will apply three different supervised learning methods on three different datasets obtained from the UCI respository and one dataset from Kaggle. In general, I will compare the three most often used learning methods: SVMs, KNN, and Decision Tree in a simpler fashion in comparison to that of Caruana's. I will also evaluate the performaces of each model using three different metrics: accuracy, precision, and F-1 Score. In the end, I will calculate t statistics and p-value between each algorithm to make further comparison.

1. Introduction

Supervised machine learning algorithms have been a significant and dominant method used in many fields and industries nowadays. These algorithms incorporate a variety of statistical, probabilistic, and optimization methods to detect useful pattern from datasets and used them to make inference and predictions. In the supervised variant, prediction models are developed by training a dataset where the label is known and the outcome of unlabelled ones can be predicted. As the algorithms obtain the training data, it produces more precise model based on that data. After training my machine-learning algorithm with data, I will compare the performance of these methods by comparing their accuracy and statistics. In this paper, extensive efforts are made to address empirical comparison of three commonly used supervised learning algorithms.

As different performance metrics measure different trade offs between classifiers, I will use t-test to measure how significant the differences between the three algorithms are. In other words, by obtaining t-test between each of the learning methods, I will know if those differences could have happened by chance. First, I performed hyperparameter selection to obtain the best parameter for each of the three supervised learning methods: SVMs, KNN, Decision Tree. Then by fitting the models using the best parameter selected, I will report the accuracy score. In the end, by calculating the t-test statistics, I will compare these methods. The results are similar to that of Caruana and Niculescu-Mizil's. Although SVM took the longest time during the hyperparameter selection process, it does prove to be the most accurate method in comparison to the other two. However, I do find my dataset covtype and letter have an overall lower accuracy across these algorithms.

2. Methodology

2.1 Learning Algorithms

I will explore the different parameters and variations used for each learning algorithm. For computational feasibility reasons, I selected some but not all parameters to obtain the best parameter set through hyperparameter selection process. This section summarizes the parameters I will use for each learning algorithms and the process of parameter selection.

SVMs:

I used the following kernels in SVM: linear, rbf, and poly and the following polynomial degree: 2, 3, and 4. I also varied the regularization parameter by factors of ten from 10^{-2} to 10^{2} and kernel coefficient by factors of ten from 10^{-3} to 10^{0} for each kernel. For each dataset, I performed hyperparameter to select the best parameter set for each dataset.

KNN:

I used the following weights parameter: uniform and distance. I also varied the nearest neighbors parameter from 1 to 600. After running 5-fold cross validation, the best parameter set is chosen for each dataset.

Decision Tree (DT):

I varied the criterion in the following two options: entropy and entropy; and the splitter parameter in the following options: best and random. I also varied the maximum depth of the tree by factors of 2 from 2 to 400. By running a 5-fold cross validation, I obtained the best parameter set for each dataest.

2.2 Performances Metrics

I used three different metrics to evaluate and compare the performances of different learning algorithms using the main classification metrics which would give me the scores of accuracy, precision, recall, and f1-score. Precision gives insight into the fraction of relevant instances among the retrieved instances while recall, also known as sensitivity, gives insight into the fraction of relevant instances that were retrieved. Understanding accuracy made us realize, we need a trade-off between precision and recall. F-score on the other hand, gives us the harmonic mean of precision and recall. I also obtain the mean of different performance metrics used to evaluate the three different learning algorithms.

2.3. Data Sets

I compared three different algorithms: SVMs, KNN, and Decision tree on four binary classification problems. ADULT, COV TYPE, and LETTER are from the UCI Repository. The fourth data set, which I call RAIN in my paper, is from Kaggle that contains about 10 years of daily weather observations from many locations across Australia.

Problem	ATTR	TRAIN SIZE	TEST SIZE	% POZ
ADULT	14	5000	27561	15.3%
COVTYPE	44	5000	576012	0.86%
LETTER	17	5000	15000	25%
RAIN	23	5000	18744	21.1%

Table 1: Description of Problems

ADULT

After reading in the adult data file as a dataframe, I dropped all missing values across all the features to avoid errors during the training process. After dropping all NaN values, I used label encoding to convert all categorical values to binary type values which is required for the training process. I normalized every entries except the last column and converted the resulting NumPy array back to a dataframe. The reason for converting NumPy array back to dataframe is due to its lesser time needed during the hyperparameter selection, especially for SVMs.

COVTYPE

This dataset has been converted to a binary classification problem by treating the largest class as the positive and the rest as negative. In a similar fashion as described above for the ADULT dataset, I label encoded and normalized the dataset and converted the dataset back to a dataframe for later training process.

LETTER

The LETTER dataset is converted to boolean type where letter A to M are positives and the rest are negatives, which yields a well balanced problem. By using label encoding and normalization, I cleaned the data and converted the dataset to a dataframe in a similar way for the other two datasets.

RAIN

This dataset has 23 attributes, each contributing to the RainTomorrow column which is the target variable for prediction. By training classification models on the target variable, we can predict next-day rain. As it has a significant amount of missing values, I dropped all missing values before the data clean process. After normalizing and label encoding, it is converted back to a dataframe for future training process.

2.4. Performance Metrics

I used three different metrics: accuracy, precision, and F score to evaluate the performances of the three different learning algorithms in comparison to the using eight different metrics in the original study. For each test problem, I shuffled and randomly selected 5000 cases for the training set and use the rest of the data as the final test set. I used 5-fold cross validation on the 5000 cases to train three different models and selected the best parameters through hyperparameter selection. After running five trials on four different dataset using three different models, I reported the the performances on the larger final test in table 2 to table 6 as well as the performances of the training sets in table 10 to table 13. Even using only three metrics and running only five trials, the differences between each model can already be easily detected.

3. Performances

Table 2 to Table 6 shows the performance for each algorithm on each of the 3 different test problems. As Caruana and Niculescu-Mizil also mentioned in their paper, as the No Free Lunch Theorem suggest, there is no universally best learning algorithms. It is also evident in the results I reported in the tables that even the best models such as SVMs can perform poorly on some problems. On the other hand, there are also models that have poor average performance perform well on other problems.

Data	Algorithm	Trial number	Accuracy	Precision	F Score
ADULT	SVM	1	0.850	0.84	0.84
ADULT	SVM	2	0.844	0.83	0.83
ADULT	SVM	3	0.844	0.83	0.83
ADULT	SVM	4	0.845	0.83	0.84
ADULT	SVM	5	0.842	0.84	0.84
ADULT	KNN	1	0.821	0.81	0.81
ADULT	KNN	2	0.822	0.81	0.82
ADULT	KNN	3	0.823	0.81	0.81
ADULT	KNN	4	0.825	0.81	0.81
ADULT	KNN	5	0.823	0.81	0.81
ADULT	DT	1	0.848	0.83	0.84
ADULT	DT	2	**0.851**	0.84	0.83
ADULT	DT	3	0.844	0.84	0.83
ADULT	DT	4	0.846	0.83	0.84
ADULT	DT	5	0.846	0.84	0.84

Table 2: Test Set Performance by Metrics

The COVTYPE dataset has a pool performance for all three different performance metrics using all three algorithms. This dataset has the most columns and attributes out of the four datasets I am using and I expected a well rounded performance using all metrics. However, the results reported in the tables shows otherwise. I believe this is due to an imbalance of the class. As the COVTYPE dataset has approximately 600000 rows yet I am only using the first 5000 to train my model, which I believe leads my datasets to class imbalance. A way to solve this low performance problem might be to weight the loss function based on the imbalance. Due to time restraint, I was unable to improve the accuracy, but this is definitely something I would try in the future to improve my model performances.

Data	Algorithm	Trial number	Accuracy	Precision	F Score
COVTYPE	SVM	1	0.648	0.63	0.62
COVTYPE	SVM	2	**0.651**	0.63	0.62
COVTYPE	SVM	3	0.644	0.63	0.62
COVTYPE	SVM	4	0.641	0.63	0.61
COVTYPE	SVM	5	0.645	0.63	0.62
COVTYPE	KNN	1	0.648	0.63	0.62
COVTYPE	KNN	2	0.649	0.63	0.63
COVTYPE	KNN	3	0.646	0.62	0.61
COVTYPE	KNN	4	0.644	0.62	0.62
COVTYPE	KNN	5	0.645	0.64	0.62
COVTYPE	DT	1	0.645	0.63	0.62
COVTYPE	DT	2	0.648	0.64	0.62
COVTYPE	DT	3	0.649	0.63	0.62
COVTYPE	DT	4	0.649	0.64	0.62
COVTYPE	DT	5	0.647	0.64	0.62

Table 3: Test Set Performance by Metrics

Data	Algorithm	Trial number	Accuracy	Precision	F Score
LETTER	SVM	1	0.632	0.62	0.62
LETTER	SVM	2	0.633	0.63	0.63
LETTER	SVM	3	**0.639**	0.63	0.63
LETTER	SVM	4	0.633	0.63	0.63
LETTER	SVM	5	0.635	0.63	0.63
LETTER	KNN	1	0.638	0.62	0.63
LETTER	KNN	2	0.626	0.62	0.62
LETTER	KNN	3	0.622	0.61	0.61
LETTER	KNN	4	0.624	0.62	0.62
LETTER	KNN	5	0.623	0.62	0.63
LETTER	DT	1	0.547	0.57	0.57
LETTER	DT	2	0.570	0.59	0.59
LETTER	DT	3	0.573	0.59	0.59
LETTER	DT	4	0.568	0.58	0.58
LETTER	DT	5	0.578	0.57	0.57

Table 4: Test Set Performance by Metrics

The LETTER dataset, in comparison to ADULT and RAIN, also has a relatively poor performance across all algorithms. In the case of this dataset, I believe that the problem might be due to the multi-class characteristic of the y variable. For this dataset, Decision Tree classifier performed the least well with accuracy, precision, and f score all ranging in the 0.55s. According the accuracy performance metrics, COVTYPE, LETTER, and RAIN dataset all have the highest accuracy using SVMs while the ADULT dataset, surpringsly, performed the best using the Decision Tree classifier. With that being said, I believe SVMs performed pretty well overall with relatively high accuracy, precision, ans f score across all testing problems.

Data	Algorithm	Trial number	Accuracy	Precision	F Score
RAIN	SVM	1	0.855	0.84	0.84
RAIN	SVM	2	0.852	0.85	0.84
RAIN	SVM	3	0.854	0.84	0.84
RAIN	SVM	4	0.854	0.85	0.84
RAIN	SVM	5	**0.856**	0.84	0.84
RAIN	KNN	1	0.828	0.82	0.81
RAIN	KNN	2	0.825	0.82	0.81
RAIN	KNN	3	0.823	0.82	0.82
RAIN	KNN	4	0.829	0.82	0.81
RAIN	KNN	5	0.831	0.82	0.81
RAIN	DT	1	0.838	0.83	0.83
RAIN	DT	2	0.837	0.83	0.83
RAIN	DT	3	0.833	0.82	0.82
RAIN	DT	4	0.836	0.83	0.83
RAIN	DT	5	0.839	0.83	0.83

Table 5: Test Set Performance by Metrics

Data	Algorithm	Accuracy	Precision	F Score
ADULT	SVM	0.844	0.834	0.836
ADULT	KNN	0.823	0.810	0.812
ADULT	DT	**0.846**	0.836	0.836
COVTYPE	SVM	0.646	0.630	0.619
COVTYPE	KNN	0.646	0.628	0.620
COVTYPE	DT	**0.648**	0.634	0.620
LETTER	SVM	**0.634**	0.628	0.628
LETTER	KNN	0.626	0.618	0.622
LETTER	DT	0.572	0.580	0.582
RAIN	SVM	**0.854**	0.844	0.840
RAIN	KNN	0.827	0.820	0.815
RAIN	DT	0.837	0.828	0.828

Table 6: Mean of Performance of Algorithms by Dataset

In Table 7, I reported the mean test set performance across all five trials for each algorithm and each datset combination with separated columns for each metrics. As the results shown in the table, Decision Tree classifier performed the best for the ADULT and COVTYPE dataset on average with the highest average accuracy while SVMs performed the best for the LETTER and RAIN dataset.

Algorithm	Accuracy	Precision	F Score
SVM	**0.744**	**0.736**	**0.731**
KNN	0.730	0.719	0.717
DT	0.726	0.720	0.717

Table 7: Mean of Overall Performances

In Table 8, I reported the mean of overall performance across all five trials for each algorithm with separated columns for each metrics. As I have expected, SVM classifier performed the best overall with the highest average accuracy, precision, and f score in comparison to KNN and DT. Althought SVM did take the longest the run 5-fold cross validation and hyperparameter selection, I believe it is one of the most accurate model used in the four different testing problems. A remainder that accuracy = $\frac{correctpredictions}{allpredictions}$, precision = $\frac{TruePositives}{TruePositives+FalsePositives}$, and F-score = $\frac{2\times precision\times recall}{precision+recall}$. Although I am only comparing the performances using these three different metrics, I believe it is sufficient to draw conclusion on the performances between the three different learning methods: SVMs, KNN, and DT.

4. Conclusion

The field of machine learning and supervised learning algorithms has made significant and important progress in the last decades and is continuing becoming an invaluable field. Learning methods such as SVMs, K Nearest Neighbors, Decision Tree, Random Forest, Logistic Regression are all efficient in turning data into real, actionable insights. It enables us to gain more insight and understand the outcomes of desired and undesired target variable.

Based on the performances of all three models after selecting the best parameters through hyperparameter selection and running five trials, I will summarize the results and comparison in the following ways. SVM gives the highest average accuracy, precision, and f score overall even though it takes the longest running time during the training process. KNN also produces very high average accuracy, precision, and f score overall that are also above average. Decision Tree has above average accuracy, precision, and f score, but does perform relatively poor in comparison to the other two algorithms. Although SVMs has the best performance overall, I believe it is also important to choose learning methods according to specific datset in order to achieve the most accurate result in the most sufficient way.

Appendex

I will report six tables in this section. Table 8 shows the T test results including the t test statistics and p-value between different algorithm combination. Table 9 shows a similar T test results between each algorithm and each dataset combination. Table 10 to Table 13 shows each algorithm's training set performance.

Algorithm	Statistics	P-Value
SVM vs. KNN	0.433	0.667
SVM vs. DT	0.517	0.607
KNN vs. DT	0.136	0.892

Table 8: T test of Algorithms

Data	Algorithm	Statistics	P-Value
ADULT	SVM vs. KNN	24.928	7.173×10^{-9}
ADULT	SVM vs. DT	-1.800	0.110
ADULT	KNN vs. DT	-16.138	2.183×10^{-7}
COVTYPE	SVM vs. KNN	0.243	0.814
COVTYPE	SVM vs. DT	-1.302	0.229
COVTYPE	KNN vs. DT	-2.446	0.040
LETTER	SVM vs. KNN	3.42	0.009
LETTER	SVM vs. DT	30.095	1.613×10^{-9}
LETTER	KNN vs. DT	23.515	1.137×19^{-8}
RAIN	SVM vs. KNN	19.556	4.860×10^{-8}
RAIN	SVM vs. DT	13.780	7.426×10^{-7}
RAIN	KNN vs. DT	5.903	0.00036

Table 9: T test of Algorithms by Datasets

Data	Algorithm	Trial number	Accuracy
ADULT	SVM	1	0.857
ADULT	SVM	2	0.862
ADULT	SVM	3	0.846
ADULT	SVM	4	0.858
ADULT	SVM	5	0.843
ADULT	KNN	1	**1.0**
ADULT	KNN	2	**1.0**
ADULT	KNN	3	**1.0**
ADULT	KNN	4	**1.0**
ADULT	KNN	5	**1.0**
ADULT	DT	1	0.854
ADULT	DT	2	0.842
ADULT	DT	3	0.846
ADULT	DT	4	0.854
ADULT	DT	5	0.855

Table 10: Train Set Performance

Data	Algorithm	Trial number	Accuracy
COVTYPE	SVM	1	0.646
COVTYPE	SVM	2	0.650
COVTYPE	SVM	3	0.661
COVTYPE	SVM	4	0.655
COVTYPE	SVM	5	0.654
COVTYPE	KNN	1	0.657
COVTYPE	KNN	2	0.653
COVTYPE	KNN	3	0.662
COVTYPE	KNN	4	0.654
COVTYPE	KNN	5	**0.660**
COVTYPE	DT	1	0.660
COVTYPE	DT	2	0.654
COVTYPE	DT	3	0.643
COVTYPE	DT	4	0.664
COVTYPE	DT	5	0.649

Table 11: Train Set Performance

Data	Algorithm	Trial number	Accuracy
LETTER	SVM	1	0.834
LETTER	SVM	2	0.844
LETTER	SVM	3	0.845
LETTER	SVM	4	0.836
LETTER	SVM	5	0.835
LETTER	KNN	1	0.997
LETTER	KNN	2	0.998
LETTER	KNN	3	0.998
LETTER	KNN	4	**0.999**
LETTER	KNN	5	**0.999**
LETTER	DT	1	0.9986
LETTER	DT	2	0.998
LETTER	DT	3	**0.999**
LETTER	DT	4	**0.999**
LETTER	DT	5	0.998

Table 12: Train Set Performance

Data	Algorithm	Trial number	Accuracy
RAIN	SVM	1	0.866
RAIN	SVM	2	0.872
RAIN	SVM	3	0.861
RAIN	SVM	4	0.866
RAIN	SVM	5	0.870
RAIN	KNN	1	**1.0**
RAIN	KNN	2	**1.0**
RAIN	KNN	3	**1.0**
RAIN	KNN	4	**1.0**
RAIN	KNN	5	**1.0**
RAIN	DT	1	0.863
RAIN	DT	2	0.857
RAIN	DT	3	0.863
RAIN	DT	4	0.858
RAIN	DT	5	0.854

Table 13: Train Set Performance

References

[1] Caruana, R., Niculescu-Mizil, A. (2006). *An Empirical Comparison of Supervised Learning Algorithms*. ICML'06.

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

[3] Young, J., (2017, December). Rain in Australia, Version 2. Retrieved February, 2021 from https://www.kaggle.com/jsphyg/weather-dataset-rattle-package