Empirical Evaluation of Supervised Learning Algorithms

Soumya Agrawal
University of California San Diego
Department of Cognitive Science
soagrawa@ucsd.edu

Abstract

Over the last few years, machine learning has gained importance and recognition in many fields. Used to predict variables businesses all the way to health, accuracy is an important aspect of the algorithms being used. This is primarily driven by feature engineering, which can be used to train a classification model and obtain the most optimal results. This paper will present an empirical comparison of five supervised learning algorithms over four data sets. By varying the training and testing data size as well as the number of trials, we are able to obtain results that can be compared across trials to find trends. Overall, highest performance was seen in RBF-SVM and Random Forest, with KNN proving to be a feasible solution but only for limited variations of the data.

Introduction

Over the last few years, machine learning has gained importance and recognition in many fields. Used to predict variables in businesses all the way to health, accuracy is an important aspect of the algorithms being used. Many associate this accuracy with the type of classifier that is chosen to be implemented, such as SVM or KNN. However, within these classifications are important parameters capable of controlling the accuracy obtained. This paper will present an empirical comparison on Linear SVM, RBF SVM, K Nearest Neighbors, Decision Tree, and Random Forest on four datasets. Following the process of (Caruana and NiculescuMizil 2006) (CNM), the goal is to compare and rank the performance of different classifiers. With the datasets havying varying details and sizes, it is clear that there might not be one classifier that outperforms all, but we will perform multiple trials and test on different partitions to obtain an accurate depiction and ranking of the different classifiers. For each classifier, we use GridSearch and cross validation to find optimal hyperparameters. Hopefully, this paper will be able to educate readers on machine learning and the factors that are able to influence accuracy.

Methods

Libraries and Packages

All classification methods were programmed and performed in Python using Jupyter Notebook. Packages like numpy, pandas, sklearn, seaborn, and matplotlib were used to import classifiers as well as functions to help analyze the data and results obtained. Sklearn and pandas contain many packages that have previously implemented classifiers as well as functions to help analyze the data initially. Seaborn and matplotlib contain packages that help build the heatmaps that depict the variety of accuracies obtained for different hyperparameters.

Classification Algorithms

Five classification algorithms were evaluated on the data. These five include: Linear Support Vector Machine (SVM), Radial Basis

Function SVM, K Nearest Neighbors, Decision Tree, and Random Forest. Parameters were explored using GridSearch Cross Validation, with the ranges for the parameters based off CNM (varying slightly depending on the efficiency of the algorithm). Below are details on the classifiers used:

Linear SVM Variations of the regularization parameter, C, which ranged from 10^-3 to 10^3 in factors of ten.

Radial Basis Function SVM Variations of the regularization parameter, C, which ranged from 10^-3 to 10^3 in factors of ten, as well as variations of the length of radial width, gamma, which had the following values: {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1,2}

K Nearest Neighbors Variations of the number of neighbors, K, ranging from K= 1 to K= 10, with increments of 1

Decision Tree Variations of the maximum depth, D, which had the following values: {1,2,3,4,5}, with the criterion set to entropy to measure the quality of a split.

Random Forest Variations of the number of features considered at each split, which had the following values: {1,2,4,6,8,12,16, 20}, and a constant number of 1024 trees

Given the size of the datasets and time limitations, subsets of the dataset were taken, where we took a maximum of 2000 data points from each dataset (unless the original dataset had less). For each of these given subsets, three partitions were created for the testing and training data. These three include: 80% of the dataset (the subset of 2000 points) for training and 20% for testing, 50% of the dataset for training and 50% for testing, and 20% of the dataset for training and 80% for testing. Furthermore, three trials were conducted for each partition (where the X and Y subset of data was reshuffled) after which accuracies were averaged. To increase performance runtime, parallel computation was enabled for each classifier when performing cross validation.

Datasets

These classifiers were then assessed over four datasets that were obtained from the UCI Machine Learning Repository. An overview of the datasets can be found in Table 1.

Population Income (ADULT) The dataset is responsible for classifying the income of individuals in the population above or below $50,000 based on career and demographic information.

Car Evaluation (CAR) The dataset is responsible for classifying a car as good, very good, acceptable, and unacceptable based on characteristics of the car as well as price. Given that this data is not binary, we will be considering acceptable, good, and very good as one classification (all come under acceptable), and unacceptable as the other classification.

Spam Filter (SPAM) This dataset classifies emails as spam or non-spam on the basis of words or character frequencies occurring in the mail.

Mushroom (MUSH) This dataset classifies mushrooms as edible or poisonous based on cap, gill, stalk, veil, and ring characteristics of the mushroom.

Table 1. Initial Characteristics of Dataset

DATASET	SAMPLES	FEATURES
ADULT	30162	14
CAR	1728	6
SPAM	4601	57
MUSH	8124	22

Experiment

The datasets were loaded from the website directly, with data frame headers given their names based on the description provided on

the websites. The data was then checked for null values, and any, if present, were removed. The data types for the different columns were then analyzed to explore if there were categorical values. If there were categorical values present, those were converted to dummy variables using a function provided in pandas. Any additional classification columns that resulted from the conversion to dummy were dropped and the final shape was recorded. Measures were also taken to increase algorithm time when running the different classifiers. A subset of the dataset was taken, with the maximum number of 2000 points being taken from all the datasets, and then being split in the partitions and trials performed. Table 2 below is representative of the final shape of the data before it was used to perform classification functions.

Table 2. Characteristics of Dataset After Manipulation

DATASET	SAMPLES	FEATURES
ADULT	2000	104
CAR	1728	21
SPAM	2000	57
MUSH	2000	117

In order to improve the classifier's time to fit the dataset, the data was scaled to only include values ranging from -1 to 1 to assist functions like SVM and have them perform equally for different features. This scaled data was then used for all the classifiers, and the training, validation, and testing accuracy was recorded in addition to the hyperparameters that were most optimal for each trial.

Results

With the datasets varying in difficulty and the amount of features required to classify, results among the different datasets can vary greatly. Table 3 shows the average accuracy obtained over 3 trials for each dataset, partition, and classifier. The accuracy was obtained by using the splits discussed before (80/20, 50/50, 20/80) and then running three trials for each split. We then take the training and validation accuracy associated with the best parameter, and compute the test accuracy using the given parameter. These accuracies are then averaged over three trials and reported in the table.

We can see that while the car, mushroom, and spam datasets had high testing accuracies, the adult dataset had a significant decrease in testing accuracy, with a drop of about 10%. However, training accuracy was exceptionally high for all datasets when using the random forest classifier. This algorithm had 100% accuracy across all datasets, which is very outstanding for the adult dataset where the other classifiers were not able to achieve anything above 90%.

Some common trends across all the classifiers were that as the size of the training set decreased, the accuracy obtained decreased, whether than be validation accuracy or testing (some exceptions to this can be seen in the adult dataset). In addition, we can see that training data maintains a high accuracy, which slowly decreases as we enter validation, and then testing. A closer look at the ADULT dataset, however, reveals there may be a problem of overfitting. While the training data achieves a 100% accuracy, the validation and testing drop about 15% to the lower 80s. We can infer that although the classifier is successful during training in predicting the data, the results are not as generalizable, possibly a result of the chosen hyperparameter.

Furthermore, as we can see from the table, RBF SVM performs the highest in most cases, with Random Forest being a close second and performing the best in some cases. Decision Tree performed the worst across all cases, with the training accuracy being significantly lower than other classifiers, and the validation and testing accuracy falling behind compared to

		ADULT			*MUSH*			*SPAM*			*CAR*		
Model	Train Size	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test
Linear SVM	0.2	0.891	0.775	0.822	1.000	0.999	0.998	0.924	0.888	0.904	0.951	0.933	0.943
	0.5	0.897	0.856	0.831	1.000	0.999	1.000	0.932	0.916	0.923	0.958	0.943	0.953
	0.8	0.871	0.845	0.829	1.000	0.999	1.000	0.942	0.920	0.912	0.961	0.959	0.947
RBF SVM	0.2	0.884	0.793	0.822	1.000	0.999	0.998	0.947	0.902	0.906	1.000	0.954	0.975
	0.5	0.874	0.849	0.814	1.000	1.000	1.000	0.946	0.921	0.926	1.000	0.992	0.997
	0.8	0.891	0.842	0.842	1.000	1.000	1.000	0.949	0.926	0.919	1.000	0.999	0.997
KNN	0.2	0.826	0.803	0.817	1.000	0.999	0.999	0.865	0.847	0.841	0.939	0.904	0.901
	0.5	0.860	0.844	0.817	1.000	1.000	1.000	0.924	0.854	0.877	0.959	0.933	0.954
	0.8	0.850	0.819	0.813	1.000	1.000	1.000	0.907	0.879	0.870	0.981	0.962	0.973
Decision Tree	0.2	0.840	0.813	0.840	0.999	0.998	0.999	0.949	0.851	0.876	0.912	0.890	0.889
	0.5	0.890	0.873	0.835	0.999	0.998	0.999	0.918	0.885	0.896	0.938	0.934	0.934
	0.8	0.849	0.834	0.853	0.999	0.999	0.999	0.912	0.891	0.884	0.936	0.937	0.936
Random Forest	0.2	1.000	0.805	0.821	1.000	0.999	0.999	1.000	0.929	0.909	1.000	0.945	0.941
	0.5	1.000	0.879	0.823	1.000	0.999	0.999	1.000	0.939	0.932	1.000	0.978	0.976
	0.8	1.000	0.841	0.830	1.000	1.000	1.000	1.000	0.946	0.923	1.000	0.991	0.991

Table 3. Average accuracies over 3 trials across all classifiers, data partitions, and data sets (rounded to 3 decimal places)

others. When looking at the hyperparameters chosen (reported in Table 4-7 in the appendix), we can see that the algorithm consistently chose a depth parameter closer to the end of the range. Further analysis could be done on whether increasing the range of the depth parameter from 1-5 leads to a better accuracy for Decision Trees. KNN, although not in the top 2, performed rather well in some datasets, but seemed to fall behind in the ADULT and SPAM dataset, possibly due to a more difficult classification problem. Overall, the MUSH and CAR dataset had accuracies greater than 90%, which is impressive and can possibly prove the simplicity of the datasets.

Conclusion

To summarize the paper, the performance of four classifiers (with linear and RBF kernel included in SVM) was evaluated on four datasets. In order to obtain the most thorough results, three trials were conducted for each classifier for each partition and results were averaged across these trials. Cross validation and GridSearch was used to determine the most optimal hyperparameter and determine the accuracy using this argument. Overall performance proved that RBF SVM and Random Forest achieve high results, with KNN obtaining high results at times but very sensitive to arguments and certain datasets. Decision tree posed to be a weak classifier, but possibly due to the hyperparameters' limit of range.

Limitations of this project include the hardware capacity of the equipment at hand and lack of knowledge of advanced algorithms. With the project being done on a laptop without any advanced computing, the capacity of data the laptop could hold was significantly lower than what I had expected. I had to take a portion of the dataset to evaluate

the classifiers due to time constraints of each algorithm. In addition to that, there are factors other than accuracy that can determine the best classifier, but we focus specifically on accuracy, which would prevent our data from not being consistent with findings in other papers. Improvements when building on this research could be to include more performance metrics and have a wider range of classifiers (including boosting algorithms).

Bonus Points

I believe I have earned bonus points for this project due to the thorough experiments I have conducted. I went beyond the requirements to provide 4 (5 if you count the RBF SVM) classifiers that were tested on 4 different datasets, providing almost double the amount of original statistics. The extensiveness of this empirical comparison is reflective of what was required.

Acknowledgements

I would like to thank the teaching staff of the Cognitive Science 118A: Introduction to Machine Learning I course at UC San Diego. Everybody, especially Professor Zhouwen Tu, has contributed to my understanding of the concepts covered in this paper and I am thankful for the knowledge gained through this class. I would also like to acknowledge the UCI Machine Learning Repository, which provided a great source of classification data to be used in this project!

References

[1] Rich Caruana and Alexandru Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning (ICML '06). ACM, New York, NY, USA, 161-168. DOI: https://doi.org/10.1145/1143844.1143865.

Appendix

Train Partition	C-Linear SVM	C-RBF SVM	Gamma-RBF SVM	K-K Nearest Neighbors	Depth-Decision Tree	Features-Random Forest
80	0.1	1.0	0.050	1.0	5.0	1.0
80	0.1	1.0	0.050	1.0	5.0	1.0
80	0.1	1.0	0.050	1.0	5.0	1.0
50	1.0	1.0	0.050	1.0	5.0	1.0
50	0.1	1.0	0.050	1.0	5.0	1.0
50	1.0	10.0	0.005	1.0	5.0	1.0
20	0.1	1.0	0.050	1.0	5.0	1.0
20	0.1	1.0	0.010	1.0	5.0	1.0
20	0.1	1.0	0.050	1.0	5.0	1.0

Table 4. Hyperparameters chosen at each trial for each classifier for the MUSH dataset

Train Partition	C-Linear SVM	C-RBF SVM	Gamma-RBF SVM	K-K Nearest Neighbors	Depth-Decision Tree	Features-Random Forest
80	10.0	100.0	0.050	9.0	5.0	1.0
80	100.0	1000.0	0.010	3.0	5.0	4.0
80	1000.0	100.0	0.050	6.0	4.0	1.0
50	100.0	1000.0	0.010	7.0	5.0	2.0
50	10.0	1000.0	0.005	3.0	5.0	4.0
50	10.0	1000.0	0.010	3.0	5.0	4.0
20	10.0	10.0	0.050	9.0	5.0	2.0
20	10.0	100.0	0.050	2.0	5.0	1.0
20	1.0	10.0	0.100	10.0	5.0	2.0

Table 5. Hyperparameters chosen at each trial for each classifier for the SPAM dataset

Train Partition	C-Linear SVM	C-RBF SVM	Gamma-RBF SVM	K-K Nearest Neighbors	Depth-Decision Tree	Features-Random Forest
80	100.0	100.0	0.010	8.0	5.0	16.0
80	1000.0	10.0	0.050	8.0	5.0	16.0
80	1.0	10.0	0.050	8.0	5.0	16.0
50	1.0	10.0	0.050	7.0	5.0	16.0
50	1.0	1000.0	0.010	10.0	5.0	16.0
50	0.1	10.0	0.050	10.0	5.0	16.0
20	10.0	100.0	0.005	10.0	5.0	20.0
20	1.0	10.0	0.050	10.0	5.0	12.0
20	0.1	10.0	0.050	8.0	4.0	12.0

Table 6. Hyperparameters chosen at each trial for each classifier for the CAR dataset

Train Partition	C-Linear SVM	C-RBF SVM	Gamma-RBF SVM	K-K Nearest Neighbors	Depth-Decision Tree	Features-Random Forest
80	10.0	1000.0	0.001	7.0	3.0	16.0
80	1000.0	1000.0	0.001	4.0	3.0	12.0
80	10.0	1000.0	0.001	10.0	3.0	16.0
50	10.0	1.0	0.010	10.0	3.0	16.0
50	10.0	1.0	0.050	7.0	4.0	20.0
50	1000.0	10.0	0.005	8.0	4.0	20.0
20	0.1	10.0	0.005	9.0	5.0	20.0
20	0.1	1.0	0.050	9.0	3.0	20.0
20	10.0	10.0	0.010	7.0	3.0	20.0

Table 7. Hyperparameters chosen at each trial for each classifier for the ADULT dataset