Comparison Between Different Classification Algorithms

Tianyu Zhang

Abstract

There are many classification algorithms, either supervised or unsupervised. It's hard to say which algorithm is better because each of them has advantage in different type of problems. This article discusses five different classification algorithms which are SVM, K-nearest neighbors, Logistic Regression, Bagging, and MLP (multilayer perceptron) and check which algorithm works best for specific classification dataset.

Introduction

Classification works to identify which of a set of categories a new observation belongs. In this particular paper, all datasets are transferred to classification between two categories. During the development of different classification algorithms, many of them have been proved to be very useful, such as Linear classifiers, SVMs, Decision Trees, Random Forest, KNN, and neural networks. Binary classification problems are very important in many industries. Other than the dataset used in this article, binary classification could be used in simple object classification or the classic cat or dog classification. When we increase the number of categories of the classification problem, we can expand the usage to almost all fields such as object detection, help detect diseases and so on.

Among all the classification algorithms, it usually turns out that neural networks work the best among all algorithms. Neural network consists of neurons, arranged in layers which convert the input vector to output. Compared to other classification algorithms, neural networks can usually catch small details of the dataset and make correct classification. There are also algorithms that require large amount of computing power such as linear kernel of SVM and logistic regression. When the dimension of dataset increases, it would take exponential of time to get the classification results. Therefore, some dimension reduction methods are introduced when the dimension of dataset reaches more than a thousand. There is also a special algorithm, KNN. The computing power of KNN is correlated to the number of entries in the dataset since for every new point that needs to be classified, it needs to go through all the points to find the nearest point to it. Even though different algorithms all have their advantages and disadvantages, for this article, we try to find the best setting for each algorithm and test which one works the best for binary classification problems.

Method

Learning Algorithms

Together 5 learning algorithms are used in the experiment of three datasets.

SVM: Support Vector Machine is supervised learning model which is used to analyze data used for classification and regression. Different kernels could be applied to the model such as linear, sigmoid. In this case, we take rbf kernel and look for best parameters such as radial and C.

KNN: K-nearest neighbors' algorithm is non-parametric method used for classification and regression. For each point that needs to be classified, it calculates its distance to all the training sample and check the output of those training samples which are closest the point. In this case, we try to find the best number of neighbors to use in KNN model.
Logistic Regression: Logistic regression is used to model a binary dependent variable which could only work in the binary classification problem since the model could not determine between three outputs. In this case, we try to find the best solver to use in logistic regression model.
Bagging: Bagging is also called bootstrap aggregating which reduces variance and avoid overfitting in classification and regression problems. It creates individuals for its ensemble by training each classifier on a random redistribution of training dataset.
MLP: Multilayer perceptron is a class of feedforward artificial neural network, which consists of at least three layers of nodes (including the input and output node). Each node passes forward using a nonlinear activation function. Compared to other algorithms, it could distinguish data that is not linearly separable.

Datasets

Three datasets are used in the experiments for comparison. They are ADULT, COV_TYPE and LETTER which is also used in the paper from Caruana (2008). I copied how the original author deals with the dataset to make it a binary comparison problem. In ADULT dataset, the output variable is chosen between <=50k or >50k. Therefore, it is already a binary classification problem. In COV_TYPE, the cover type has seven different options. After running through the entire dataset, we find the most occurrence of a cover type and the classification is between this cover type and all other cover types. As for the LETTER dataset, I also followed the way Caruana deal with it by putting all letters from A-M as positive and the rest as negative. Therefore, after some cleaning of data, we got three binary classification problems.

Experiments

For each of the three datasets, the first step is cleaning by putting the target value to either positive or negative. Then for variables that are not numerical values, we use one hot encoding for all the non-numerical values. For instance, in the first dataset ADULT, the original data has 15 columns, after some cleaning, there're 109 columns to be analyzed. After cleaning data, we put data into 3 different splits, 20% training and 80% testing, 50% training and 50% testing, and 80% training and 20% training which is the conventional way. For each split, we did three trials on it and for each trial, 5000 random data are picked from the entire dataset. The reason why this is plausible is because in both COV_TYPE and LETTER dataset, the ratio between positive output and negative output is close to 50%. In the ADULT dataset, the ratio is about 1:2. Therefore, picking 5000 samples from it randomly will not affect too much about the ratio between positive outputs and negative outputs. The benefit of that is it could save us some computing power. I've tried train a single linear SVM model on 80% of the ADULT dataset and it took more than 2 hours. After picking out 5000 random samples, we divide them into two parts using three different partitions. And for each partition, we apply five different classification algorithms on it.

For each algorithm, GridSearch is used so that we could find the best hyper-parameter for each model. Finally, we use the hyper-parameter found in GridSearch to train the training dataset and see the accuracy of testing dataset. Then we repeat this process for trial No.2 and trial No.3. After finishing all three trials under one split and one data frame, we record the average testing accuracy of each model and here are some results.

	split1_average_accuracy	split2_average_accuracy	split3_average_accuracy	Average
SVM	0.759667	0.760000	0.761000	0.760222
KNN	0.764167	0.777200	0.780333	0.773900
Logistic	0.781333	0.793733	0.795000	0.790022
Bagging	0.831667	0.838667	0.844000	0.838111
MLP	0.792167	0.785067	0.793333	0.790189

(Table 1. Average accuracies of different classification algorithms in the ADULT dataset, split 1 represents 20% training + 80%testing, split 2 represents 50% training + 50% testing, split 3 represents 80% training + 20% testing. Same for other tables listed below.)

	split1_average_accuracy	split2_average_accuracy	split3_average_accuracy	Average
SVM	0.513000	0.510550	0.519333	0.514294
KNN	0.682500	0.723525	0.775333	0.727119
Logistic	0.584167	0.601742	0.617000	0.600969
Bagging	0.744250	0.764463	0.790000	0.766238
MLP	0.515833	0.554958	0.646333	0.572375

(Table 2. Average accuracies of different classification algorithms in the COV_TYPE dataset.)

	split1_average_accuracy	split2_average_accuracy	split3_average_accuracy	Average
SVM	0.659083	0.865600	0.894667	0.806450
KNN	0.890333	0.932267	0.941000	0.921200
Logistic	0.721333	0.726400	0.704333	0.717356
Bagging	0.861083	0.901733	0.914333	0.892383
MLP	0.811500	0.853733	0.869000	0.844744

(Table 3. Average accuracies of different classification algorithms in the LETTER dataset)

These tables represent the average accuracies of each algorithm applied onto different datasets under different splits. The specific trial accuracy is given in the code section. We discover that in

both ADULT dataset and COV_TYPE dataset, bagging algorithm is the leading algorithm among these five in every single partition and the grand average. However, when it goes to the LETTER dataset, bagging loses to KNN algorithm. My thoughts on this is that the LETTER dataset has the simplest features and since its letter recognition, these features almost represent pixels from the written letter. Therefore, KNN could outperform other algorithms by its ability to find similar categories in the training set. Also, it's due to the classification being between letters A-M and N-Z. Letters in these two categories can be vastly different and maybe Bagging algorithm could outperform other algorithms if it comes to single letter classification. The surprising point here is that neural networks doesn't do well in all three datasets, not even close the being good in the COV_TYPE and LETTER dataset. This is probably because I didn't tune to the suitable number of hidden layers in the neural networks. Also, in all three datasets, increasing the proportion of training data would always lead to increasing the average testing accuracy.

Conclusion

As we can see from the test results, Bagging works better than other algorithms beside the LETTER dataset, which shows that each algorithm has its own strength and disadvantage when facing different datasets and features. Besides, in SVM, there're other types of kernels, and for other algorithms, there're also many other hyper-parameter settings to run. Due to the restriction of my local computing power, I'm not able to go through all the parameter settings to find the best one. However, with some adequate settings of each algorithm listed in this article, the accuracy rate for datasets with large number of features like ADULT, COV_TYPE and LETTER could return some good results with high accuracies.

References

Caruana, Rich, and Alexandru Niculescu-Mizil. "An Empirical Comparison of Supervised Learning Algorithms." *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, 2006, doi:10.1145/1143844.1143865.