Comparisons of Different Classification Algorithms based on Binary-outcome Data Sets of Various Dimensions

Zhiwei Wang, Bingdao Chen

1 Introduction

The different mechanisms of classification algorithms determine that their performance will change in different data sets. We want to explore the specific pattern of how their performances vary with the change of data sets' features.

In the project, our goal is to compare 5 different classification algorithms: Logistic Regression, K-nearest Neighbors, Decision Tree, Random Forest and Support Vector Machine according to their performances on 8 different data sets. The comparison is mainly based on the prediction accuracy on test data as well as the corresponding time cost.

We select the data sets according to their dimension features, the number of instances n and the number of attributes d. We believe that, for a given classification algorithm, its performance varies when these two dimensions change a lot. Generally, we divide the data set types into 4 different kinds: small n & small d, large n & small d, small n & large d and large n & large d. Two data sets are selected for each kind in order to make our conclusions more persuasive.

2 Methodology

2.1 Algorithms

In our study, we use five classification algorithms: logistic regression, K-nearest neighbor (KNN), decision tree, random forest, support vector machine (SVM). They both have their own operational rules. Logistic regression use maximum likelihood estimator method to predict. KNN fins K nearest points and assign the data to the class whose frequency is most in these K points. Decision tree uses tree to classify. Random forest contains bagging and random forest. Support vector machine divides separate categories by a clear gap that is as wide as possible. We will apply these algorithms to datasets of different types to see their performance.

2.2 Datasets

The datasets we use are shown in Table 1, and detailed description of datasets is in the following.

Data Set List				
Type	Small n Small d	Large n Small d	Small n Large d	Large n Large d
Data sets	**Forest Mapping**	**Skin Segment**	**Arcene**	**Gisette**
	Audit Risk	**HTRU2**	**Gene Expression**	**P53 Mutants**

Table 1: Dataset list

- **Forest Mapping (n=354, d=27)**: Mapping different forest types based on their spectral characteristics using ASTER satellite imagery. We extract the class “Sogi” and “Kinok” from the multiple classes. Numerical attributes and binary outcomes.

- **Audit Risk (n=777, d=18)**: Predict the fraudulent firm on the basis the present and historical risk factors. Numerical attributes and binary outcomes.

- **Skin Segmentation (n=245057, d=4)**: The skin dataset is collected by randomly sampling B,G,R values from face images of various age groups, race groups and genders. Numerical attributes and binary outcome.

- **HTRU2 Data Set (n=17898, d=9)**: HTRU2 is a data set which describes a sample of pulsar candidates collected during the High Time Resolution Universe Survey. Each candidate is described by 8 continuous variables, and a single class variable (binary).

- **Arcene (n=900, d=10000)**: Distinguish cancer versus normal patterns from mass-spectrometric data. This is a two-class classification problem with continuous input variables.

- **Gene expression cancer RNA-Seq (n=801, d=20531)**: It is a random extraction of gene expressions of patients having different types of tumor. We extract two (KIRC & LUAD) of the five tumor classes. Numerical attributes.

- **Gisette (n=7000, d=5000)**: Gisette is a handwritten digit recognition problem. The problem is to separate the digits ’4’ and ’9’. Numerical attributes and binary outcomes.

- **P53 Mutants (n=16772, d=5409)**: Model mutant p53 transcriptional activity (active vs inactive) based on biophysical simulations. Numerical attributes and binary outcomes.

3 Implementation Details

In this section, we will show how we process datasets, how to implement algorithms and how to evaluate the performance of these algorithms.

3.1 Data Pre-processing

- **Data loading**: Original data are stored in different file types. Also, some of them are stored in matrix form. At the same time, NA values are represented in different ways such as ”?” or na. We use different loading function to extract data.

- **Processing**: Deal with NA values according to data size (drop or replace with mean). Standardization. Check each attribute’s data type. Recoding the label attribute into numeric $\{-1, 1\}$.

- **Resampling and Splitting**: Balanced data often works better in classification. We use SMOTE to oversample the minority class and then split the data into training part and testing part.

3.2 Algorithms Implementation

1.Parameters tuning: For each data set and each algorithm, we use gridsearch to choose the best parameters among the candidate pool using validation.
2.Algorithms building: We define 5 different functions which take in train data, train labels, test

data and test labels from each processed data sets and output the prediction accuracy, confusion matrix and the running time.

Another point to illustrate here is that we choose not to do feature selection or dimension reduction for Logistic Regression algorithm. The reason is our comparison between algorithms is based on data sets of different type of dimension features. If feature selection or dimension reduction is conducted before training, actually we are making changes to the d dimension of the corresponding data set which is against to our initial purpose.

3.3 Performance evaluating

1.Running time: We record the duration of training model over different datastes by different algorithms. It can reflect the efficiency of an algorithm.
2.Accuracy rate: For each trained model, we calculate the accuracy rate for train data and test data. It not only can compare the predictive ability of algorithms, but also can find whether there is a over-fitting phenomena in some algorithms.
3.ROC and PR curve: For each data set, we draw ROC and PR curve separately for each classification algorithm and plot 5 curves onto one figure. The area under the ROC curve and above the PR curve could help us better compare their performances.

4 Result

We apply 5 classification algorithms to 8 date sets of 4 different types and the result are shown in Table 2. We mainly focus on the classification accuracy within the test set.

Algorithm	S-S-1		S-S-2		L-S-1		L-S-2		S-L-1		S-L-2		L-L-1		L-L-2	
Logistic	train 1.0	test **0.9378**	train 1.0	test 1.0	train 0.9190	test 0.9186	train 0.9416	test 0.9420	train 1.0	test 0.82	train 1.0	test 1.0	train 1.0	test **0.981**	train 1.0	test 0.9961
KNN	train 1.0	test 0.9253	train 1.0	test 1.0	train 0.9997	test **0.9996**	train 0.9929	test 0.9724	train 0.87	test 0.8	train 1.0	test 1.0	train 0.9777	test 0.949	train 1.0	test 0.9954
DT	train 1.0	test 0.8714	train 1.0	test 1.0	train 0.9999	test 0.9992	train 0.9955	test 0.9528	train 1.0	test 0.64	train 1.0	test 0.9885	train 0.9668	test 0.93	train 0.9994	test 0.9928
RF	train 1.0	test 0.9170	train 1.0	test 1.0	train 0.9998	test **0.9996**	train 1.0	test **0.9758**	train 1.0	test 0.78	train 1.0	test 1.0	train 0.998	test 0.97	train 0.9996	test **0.9983**
SVM	train 1.0	test 0.9253	train 1.0	test 1.0	train 0.9983	test 0.9983	train 0.9474	test 0.9470	train 1.0	test **0.83**	train 1.0	test 1.0	train 0.9865	test 0.978	train 0.9988	test 0.9974

Table 2: Summary of Accuracy (e.g. "L-S-1" represents the first large n small d data set)

Preliminary Findings:
1.Among the 8 data sets, we find that 2 of them demonstrate an optimal prediction accuracy both in training set and testing set. Perfect prediction performance possibly results from good feature selection as well as decent data condition (no abnormal points).
2.KNN, Random Forest and SVM indicate a pretty stable performance among all data sets (especially Random Forest).
3.The performance of Logistic Regression varies greatly according to data sets dimensions. It does very well in small data sets while worse performance happens in Large-n Small-d data sets.
4.As a general conclusion, Decision Tree is inferior to Random Forest.

Then, the running time results in seconds is shown in Table 3. We did not count in the parameter tuning time because different models acquire different number of parameters and we simply care about the exact cost in algorithm running process.

Algorithm/Data Set	S-S-1	S-S-2	L-S-1	L-S-2	S-L-1	S-L-2	L-L-1	L-L-2
Logistic	<0.01	0.0156	0.6093	0.2656	0.8437	0.5469	7.0156	34.25
KNN	<0.01	<0.01	0.4375	0.0781	0.0781	0.3906	9.1719	56.6563
DT	<0.01	<0.01	0.3281	0.3594	0.2031	0.4219	12.4375	233.9688
RF	1.0156	1.0156	9.8593	4.4844	1.4844	2.1875	11.875	1134.6719
SVM	0.0156	0.0156	155.5	22.6406	0.7969	1.8594	344.3906	1750.2186

Table 3: Summary of Running Time (e.g. "L-S-1" represents the first large n small d data set)

Preliminary Findings:
1.Generally speaking, Logistic Regression and KNN has the least time cost. Even if in Large n Large d data sets, they still runs fast. As a contrast, Random Forest and SVM's time cost have an immense increase in large data sets.
2.Among all classification algorithms, compared to number of feature d, number of samples n has a much greater effect on the time cost.
3.It's very intuitive that Random Forest cost much more time than Decision Tree because of the bagging process.
4.SVM's time cost has the most amazing growth rate when n is large.

From the ROC curves and PR curves, which are shown in the Appendix, random forest algorithm has a good performance in datasets of all types, and decision tree algorithm has a bit poor performance in datasets of all types. Logistic regression algorithm and support vector machine perform well in "Small n & Small d" datasets. KNN performs not well in "Large n & Large d" datasets.

5 Conclusion

- Logistic Regression performs pretty well in small data sets. However, although saving a lot of training time, it is not an optimal choice in large scale data sets. The first reason is that it does not have an outstanding performance especially when the number of features is small. Secondly, feature selection is also pretty time-consuming in Logistic Regression.

- KNN, Random Forest and SVM have very stable performance among all kinds of data sets. This advantage results from the originality and universality of the algorithms. However, the cost is the time. In reality, if we do not need to repeat the training process for many times and only care only about the accuracy, these universal algorithms are good choices.

- Over-fitting easily arises when applying Decision Tree algorithm and Logistic Regression,especially when the number of features is large.

- Generally speaking, Random Forest is superior to Decision Tree. Firstly, it improves the over-fitting problem because of the bagging process. Secondly, Random Forest is a collection of Decision Trees. The aggregation limits over-fitting as well as error due to bias and therefore yield useful results.

- SVM algorithm has a good performance for datasets of all types. However, it is not suitable for large size datasets because it takes a lot of time to train model.

- Separating from the perspective of algorithm comparison, features themselves matter. If we consider variables with greater importance in our models, algorithms' performance will be improved a lot. 2 of the 8 data sets we choose demonstrate such phenomena.

6 Appendix

ROC curves and PR curves are shown in the following.

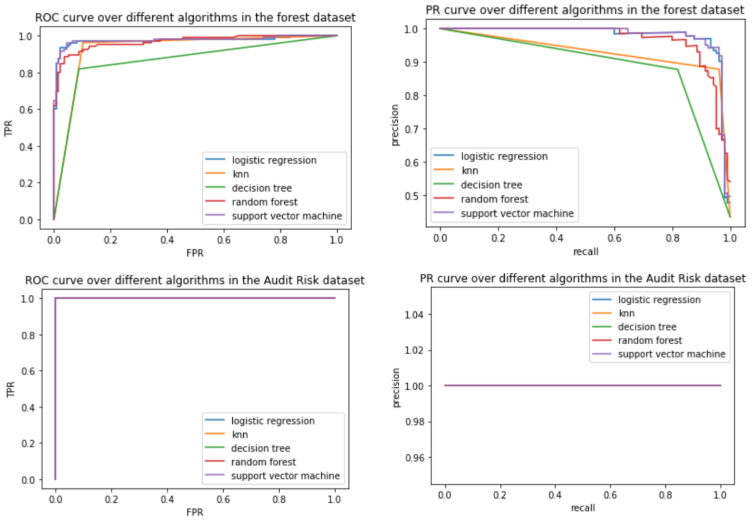

Figure 1: ROC curves and PR curves for Small n & Small d

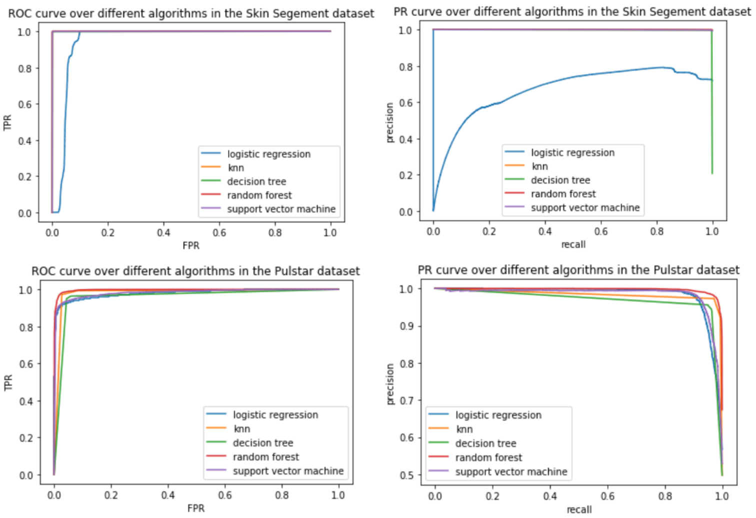

Figure 2: ROC curves and PR curves for Large n & Small d

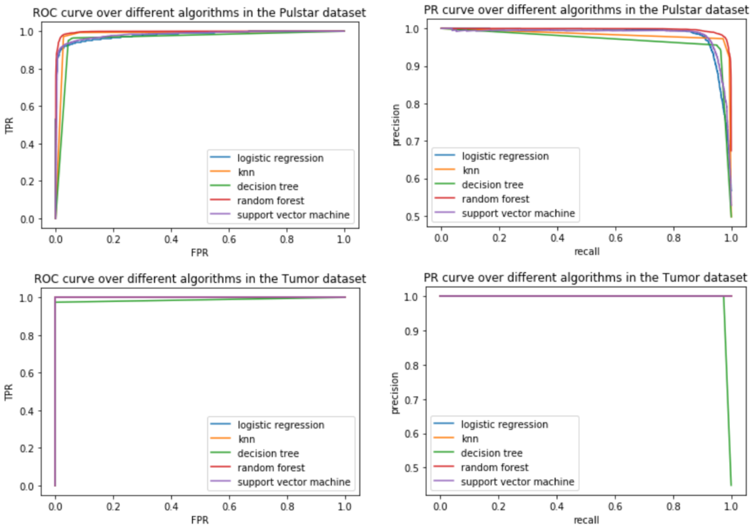

Figure 3: ROC curves and PR curves for Small n & Large d

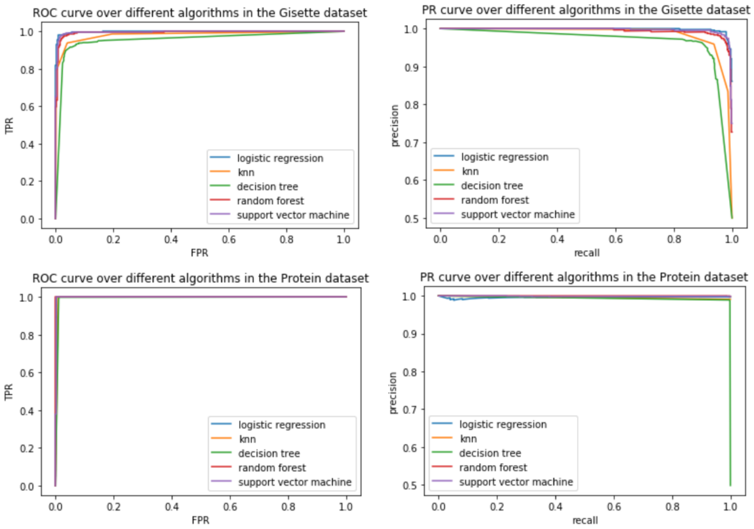

Figure 4: ROC curves and PR curves for Large n & Large d