Machine Learning Nano Degree Project 2

Kai Xiong

June 2016

1 Introduction

In this report, we analyse student intervention dataset. We extract 48 features to identify if a student needs early intervention. Thus this is a classification problem due to its binary outcome.

2 Statistical Analysis and Data Exploration

The general statistics of the Boston house price dataset is summarized as follows:

Statistics	Value
Total number of students	395
Number of students who passed	265
Number of students who failed	130
Graduation rate of the class (%)	67%
Number of features (excluding the label/target column)	30 (without the label column)

Table 1: General statistics

3 Preparing the Data

We use all columns but the last column as features. The last column which denotes if a student pass or fail is the target column. There are a number of columns that have non-numeric values. For the columns only have "Yes" or "No", we convert them to 0/1 (binary) values. For other columns that have more than two values, we create as many columns as possible values and assign 1 to one of them and 0 to others. We then shuffle the data and split them into training and test sets. We use 300 data points, which is about 75% of the data, as the training set and others as the test set.

4 Training and Evaluating Models

We have used three supervised learning models, namely, SVM, Naive Bayesian, Boosting classifier to solve this problem.
SVM is a popular model that can be applied to classification or regression problems, e.g. classification of facial expressions and texture classification. It uses kernel to transform data into higher dimensions and

then separate them using a margin as wide as possible in the higher dimensional feature space. New data points are also casted to the high dimensional space, where they are more easy to be separated. Therefore, SVM is good at classifying both linear and non-linear problems. Nonetheless, the disadvantages are that SVM is fairly computationally costly when using sequential minimal optimization, which is what libsvm and scikit learn uses. The training time complexity is $O(n^3)$. Scikit learn's implementation of SVM's can thus only handle up to 10,000 samples. As this problem has high dimensional feature space and the features and labels have non-linear relationship, SVM is good for transforming existing features to another space for better classification accuracy. Also we have a sample size of 395, which is much less than scikit's bottleneck. Thus SVM can handle this dataset well.

We also apply Naive Bayesian classifier to the problem as NBC is fast in terms of training and making predictions. NBC is part of the probabilistic classifiers based on Bayes theory with the assumption that all features are conditionally independent of each other. NBC is very efficient in linear classification problems. And it only requires small amount of data to train the model. The disadvantage of this algorithm is that its precision and recall are low when using a small dataset. Problems involving large datasets have been solved well with NBC such as spam detection in the emal and news article classification [4]. We choose naive bayes in our problem setting as it is fast to train and predict.

Boosting is an ensamble learning algorithm primarily for reducing bias and variance. It is known for converting weak classifiers to strong classifiers. Usually, it weights multiple realizations of classifiers and sum them up to get a more accurate results. In this project, Gradient boosting algorithm is applied as it allows optimization of an arbitrary differentiable loss function. The disadvantage of Gradient boosting is that its training time is generally longer as it builds tree models sequentially and it is hard to parallel the algorithm. Gradient boosting has been applied to marketing problems [2] and political problems [3]. Gradient boosting concludes each student's label on a higher picture. After some iterations, it will have a good weighing scheme to weigh weak and strong data points and thus make accurate predictions.

Performance of the supervised learning algorithms are presented below. From the tables, we see that Naive bayesian is the most time efficient algorithm. However, its F1 score on training set and test set is the lowest. SVM has intermediate speed and performance. It does well in both training set and test set. Boosting is the slowest algorithm. It has the highest score in the training set however, performance drops significantly in the test set.

Training set sizes	Training time	prediction time	F1 score on training set	F1 score to test set
100	0.002	0.005	0.8481	0.8115
200	0.005	0.004	0.8868	0.8013
300	0.007	0.003	0.8761	0.7838

Table 2: Statistics for SVM

Training set sizes	Training time	prediction time	F1 score on training set	F1 score to test set
100	0.001	0.003	0.8308	0.6935
200	0.004	0.005	0.8102	0.7472
300	0.002	0.001	0.8037	0.7633

Table 3: Statistics for Naive Bayes

Training set sizes	Training time	prediction time	F1 score on training set	F1 score on test set
100	0.090	0.003	1	0.7335
200	0.112	0.001	0.9929	0.7735
300	0.150	0.002	0.9739	0.7794

Table 4: Statistics for Boosting

5 Best model

SVM has moderate F1 score and running time. Naive Bayesian has poor accuracy but it is the fastest among the three algorithms. However, we observe that all the running time are pretty negligible as the dataset is small. So out of sample accuracy, namely high testing F1 score is the most important metric we are looking at. Conclusively, Gradient boosting is the optimal classifier in terms of accuracy.

The best parameter is with learning rate equal to 0.02, max depth equal to 1 and min samples leaf equal to 3. The F1 score for the training set is 0.9738 and the F1 score for the test set is 0.8382. The prediction time is 0.003 for test set. Thus we see Gradient boosting works well both in sample and out of sample. And the performance in the test set is significantly improved.

In Layman terms, the Gradient boosting algorithm achieves its goal sequentially. Whenever a new student's profile comes in, it is included in the new model and considers all students' data collected so far to come out a solution that fits everyone. Thus, it solves the problem sequentially. Say the algorithm is a board of experts each of provides an opinion whether a given student needs intervention. The first expert will make his own decision. And remaining experts provide opinion in sequential order such that each one tries to improve the overall judgement of all experts that came earlier in the sequence. When we have a new set of students' profile, we assume the new students' profiles are similar to what we have already examined. The group of experts will judge the new students' profile and make conclusions based on what they know about the group of students they previously judeged.

6 Conclusion

In this report, we study basic statistics of the boston student intervention dataset. We compare SVM, Naive Bayes and Boosting learning algorithm and find Gradient boosting is the optimal solution to this problem. We then use grid search to tune the model. Results show that Gradient boosting works well both in sample and out of sample.

7 Reference

1. Lipo Wang, Support Vector Machines: Theory and Applications, Springer, Berlin, 2005

2. Lemmens, A. and Croux, C. (2006), Bagging and boosting classification trees to predict churn. Journal of Marketing Research, 43, 276–286

3. King, G. and Zeng, L. (2001), Explaining rare events in international relations. International Organization, 55, 693–715

4. Vicent Alabau, Jesus Andres, Francisco Casacuberta, Jorge Civera Jose Garcıa-Hernandez, Adri‘a

Gimenez, Alfons Juan, Alberto Sanchis, Enrique Vidal, The naive Bayes model, generalisations and applications