Stats202 Final Project

By Fang Lin (Stanford ID # 06166564)

August 8, 2016

Abstract

The main goal of this final project is to classify whether the given URLs are related to inputed search queries based on the 10 attributes, and return relevant URLs for search queries that users enter. To make relevance predictions for each row of the test data set deciding whether URLs are relevant for the query. Totally, 80046 observations are provided with the 10 attributes and class labels (1: relevant or 0: irrevalent). Also, addition 30001 unlabeled test data are given for validation. I applied five different types of classifier taught in the class including Logistic Regression (Linear and Quadratic), Naive Bayes, K-nearest neighbor, Support Vector Machine, Decision Tree, and Random Forest.

1. Introduction and Data Observation

Going through the traning and testing data set. Both of them don't have any missing value or duplicated value rows (According to the duplication checking). There are two nominal attributes, query_id and url_id, which are repr, esenting search query and URLs accordingly. There are totally 10 attributes for each URL besides the "query_id" and "url_id". Attributes like "query_length" and "is_homepage" are self-explanatory, while attributes such as "sig1", "sig2", "sig3", "sig4", "sig5", "sig6", "sig7", "sig8" remain to be explored. About 93.9% records have a unique url_id, while 15.5% recordes have a unique query_id. query_length is to represent the number of items in a query which actually have effects on the number of relevant URLs. The correlation analysis shows "query_length" and relevance's correlation coefficient is -0.0005, which as expected query length has nothing to do with relevance. The binary value is_homepage is able to represent whether they are the homepage. If the queries are from homepages, they are appear to have a higher number of relevant URLs (49%) than the queries are not on homepages (42%). Other 8 variables from "sig1" to "sig8" are all continuous ones with zero as minimum values, but their maximum values range from 0.86 for "sig2" to 673637 for "sig3". Though, correlation analysis among them shows "sig3" and "sig5" have an unusually high correlation of 0.815, we judged non were high enough to leave out of the model. So, we decided to select 9 variables except "query_length" for the following analysis. It can be noticed that correlation among a number of signals, and with URL relevance, increased with logging. No signal attribute showed a high correlation with URL relevance however correlation does increase slightly with logging on signals 3 through 6.

2. Approaches Evaluations

To use the data set efficiently, we split it into five segments with same sizes. Every segment is used as test set with other segments as training set in the cross validation steps. Thus, the entire training set will be tested and classification error will be calculated and compared. As discussed before, we choose 9 attributes as maximum to all classifiers. Although we know that cut one variable (sig3 or sig5) might have good effects on some model's performance, we choose to only use this way on one of the best classifier among all six.

4. Candidate Solutions and Data Mining

a. Logistic Regression

cv.glm() function from **boot** package is used for the logistic regression classifier. Through 10 fold cross validation approach the calculated misclassification rate is about **34.8%**.

b. Discriminant Classifier

Applying linear discribinant analysis, the cross-validation error rate is **35.0%**. Applying quadratic discriminant analysis, the cross-validation error rate is about **40.0%**. According to this comparison, it's more reasonable to use linear modle.

c. K-nearest Neighbor Classifier

Then, I apply the K-nearest Neighbor Classifier. Choosing K from 1 to 10 with knn() function from package "class". The following table shows the cross-validation error rate from K=1 to K=10.

	Cross-validation Error
K=1	45.2%
K=2	45.7%
K=3	44.4%
K=4	44.4%
K=5	43.8%
K=6	43.8%
K=7	43.3%
K=8	43.6%
K=9	43.4%
K=10	43.3%

d. Naive Bayes Classifier

Naive Bayes is the most straightforward model, which is able to predict both numerical attributes and categorical attributes. naiveBayes() from package "e1071" is used for the model. With 10-fold cross validation method, I got the incorrection probablity of the model is **40.00%**.

e. Support Vector Machine Classifier
With the same "e1071" package, through svm(), I applied radial kernel and linear kernel. About first 16010 rows are picked as the test set due to the super long running time needed for SVM algorithm. As we know that the complexity of SVN is O(n^2), it takes about 27 minutes for running on my MacBook. The error rate of using SVM with kernel method is **34.5%**, and using linear SVM is **35.0%**

f. Decision Tree Classifier
Through the tree() function, we fit the variables to a size=3 decision tree, becaused the cross validation shows the standard devidation becomes lowest when tree size=3. Thus, there is no need to prune anymore. The misclassificatio rate is **36%**, which is slightly larger than the following method.

g. Random Forest Classfier
Using "randomForest" package and its randomForest() function, with 500 trees, I got the OOB estimate of error rat is about **35.1%** according to the summay. It seems there is no need for cross validation to get an unbiased estimate of the test set error in random forests since the out-of-bag error is estimated internally.

5. Summary of Result Evaluation

As we can noticed from the results above, radial SVM Classifier performed the best among all the classifiers following by linear SVM, Logistic Regression, Random Forest Classifier, Discriminant Classifier, Naive Bayes Classifier. K-nearest Neighbor Classfier performs worst even using K=10. Thus, I used radial SVM model to predict the test set. Please see attached test file "DecisionResults.txt".