CS 5011: Machine Learning Contest Report

Due on Sunday, November 18, 2014

Team 04

Pratik (cs12b023) and Sagar (cs12b039)

Missing Values:

We first tried to remove all the instances and columns of data which had missing values. Both ways didn't work as there were missing values in almost 2411 out of 3500 instances and 1606 out of 1897 columns.
We have tried the following methods to fill in the missing values.
1. Mean.
2. Median.
3. KNN for different values of k(4,8,10).
4. Used weka directly without filling values.

We initially used mean and median. The scores we got were almost same for different classifiers. When we used KNN we got significantly better scores.

Class Imbalance:

There was a huge class imbalance (700:2800) in the data. Due to this many classifiers were not working well.
Methods used to handle class imbalance:
1. Oversampling using our own algorithm.
2. Giving class weights in some classifiers.
3. Using cost sensitive classifiers available in weka.
4. Smote.

Some of the classifiers had an option of giving weights to the classes but most of them did not have this feature (like random forests etc). So we first tried to oversample the data on our own by repeating the values that were in the training set 3 times to make that class balanced. But this led the classifier to over fit the class A and was not giving good scores even compared to the score without oversampling.
We then tried cost sensitive classifier in weka. Finally we used smote for oversampling of data.

Dimensionality Reduction/Extraction:

We used the following methods for dimensionality reduction/extraction.
1. Principal Component Analysis.
2. Linear Discriminant Analysis.
3. Feature Agglomeration.

We used PCA to extract number of features(100, 200, 1000 etc) but nothing worked as better as using all the features directly. So finally we did not do any feature reduction/extraction.

Normalization:

We initially normalized the data using normal normalize function in sklearn. Then we used Normalizer function which actually regularizes the data in sklean which gave very good scores compared to normalize and almost the best of all till then (around 82%). But when we submitted we got very less score (26%) and we don't know the reason behind it till now.

Cross Validation:
These are the following methods we used for cross validation.

1. k-fold Cross Validation.
2. Stratified k-fold Cross Validation.

Stratified k-fold tries to preserve the class distribution in the train and test data set. Initially when we were using normal cross validation. Our scores were random because of this reason as some times the class distribution is captured in the train and test datasets and some times not. Then when we used stratified k-fold cross validation we got better scores and were able to distinguish which were correctly classifying and which were not.

Classifiers:
These are the different classifiers we used for the classification task.

1. SVM.
2. Random Forest.
3. Adaboost.
4. Bagging.
5. Gradient Boosting.
6. Gaussian Markov Model.
7. Linear Regression.
8. KNN.
9. Logistic Regression.
10. Random Tree.
11. Naive Bayes.

We found that all the classifiers we performing badly except gradient boosting and adaboost. We first started with n = 500 in gradient boosting and increased upto 2500. But we found out that after n = 1500 the scores came down again.

Majority Voting:

After using a lot of classifiers we then started voting using these different classifiers. This will reduce the variance and also error made by one type of classifier won't be made by other type of classifiers. This helped us get more scores at last.

Brief Summary of how we have done it:

I think that the data is not linearly separable. This might be because none of the linear models we used were working better. We used gradient boosting with n = 1000 on the mean filled missing values data for our first submission (76%). And then we increased n to 1500 and we got (77%). But then when as we

increased n further more our score decreased. Then we tried using median data. But this did not work. We tried KNN using different values of k = 4, 8, 10. We found that 10 gave better answers on the cross validation.

Then we shifted to weka from python.
You can find our work at `https://github.com/jpsagarm95/ML_Contest`