Location Classification from Twitter text

Chau, D. Thuan, SDSU Student, Lo, W. Donovan, SDSU Student

Abstract **– Dialect and the choice of words provide good indication of the individual's origin. Particular words are often used more frequently or exclusively than others . In this proposed work, we apply various classifiers on a collection of tweets from two locations and predict the users' origin.**

In this paper, we will examine the process of Twitter collection and data preparation. Following the second section, we will discuss the different classifiers applied on the cleaned data. From each of the classifiers, we will pick the best configuration and determine the most accurate model to be used on the testing set. Lastly, we will evaluate the performance of the model on the testing set.

We hope the results in this work will provide new findings of the area of location prediction with social media data.

Index Terms – adaboost, **classifier, decision tree, machine learning, neural network, pca, social media, support vector machine, Twitter**

I. INTRODUCTION

WITH the growing surplus of Twitter data produced each day by Twitter users, it makes data collection convenient and data sampling more diverse. But given such large amount of data, how can we utilize this factor to our advantage in learning something new. In this paper , we explore the concept of implementing classifiers to predict the users' location origin from the based the most frequent words that are used in those regions. This implementation will rely on large of sample data to correctly represent the population. If the sample size is too small , there may be words that are frequently used but is considered as infrequent, thus leading to misrepresentation.

In different locations, people may use exclusive words in their tweets respective to the topic of discussion or use particular words more based on the manner of which they expressive themselves. By capturing a large population of samples in different regions, we can create a word frequency table to model that location. If a collection of tweets from one particular user with an unlabeled location of origin is presented, the model can classify the user's location based on the words listed in the tweet.

II. DATA PREPARATION

The Twitter data collected in this paper uses Julia Language and the Twitter API. The work flow begins with programming environment and Twitter account setup, followed by Tweet collection, processing, and structuring.

In order to collect tweets from Twitter API, a Twitter account and Twitter Application needs to be setup. From the created Twitter Application, the access token, access secret, consumer key, and consumer secret can be extracted and used for the Twitter API. Julia language was chosen for its interactive property and Twitter API availability.

Once the programming is setup, we implement how Tweets is collected from two regions for our classification problem. We first decided to pick two Twitter accounts to extract Twitter IDs from, namely “KPBS” who represents San Diego and “FOXLA” who represents Los Angeles. From the two accounts, we gathered its followers in hopes that the users will be from the two locations. We then gathered tweets from the followers and removed any tweets whose user is not from San Diego or Los Angeles. The location San Diego and Los Angeles labels for each tweet is normalized to “SD” and “LA” for the classification ground-truth labels.

In order to build classifiers to predict the location of the user from a given tweet text, we formulated a Bag of Words table that holds the frequency count of each word used for each user. So in the table, the columns are words and rows are users.

The implementation of the column of words in the Bag of Words was first attempted by gathering all Tweets and gathering all unique words from our collection. However, from doing so, we encountered an estimate of 500,000 unique words. Majority of the word in the list was largely due to how words were tokenized and the large uncleaned unique words from structure of URL links, hashtags, and mentions. Further, in creating our own dictionary of words, the words we tokenized was checked for symbols, stemming letters, and article words. Due to the large effort that we will need to spend on cleaning the data to build a list of dictionary words in Bag of Words, we decided to use a preexisting list of 10,000 common words.

For each tweet that we gathered, we then increment the count of the words respective to the user, creating a profile to be used in MATLAB for classification. The final structure of the data from Julia code that will be used in MATLAB for Classification is a csv file, where the rows are user profile and columns are the word's frequency.

The Julia Code is provided in the attached folder “Julia_Code” which contains a README file.

III. METHODS

This paper propose to utilize a Bag of Words system with respective to the Twitter User as the profile for the sample sets used in our classification problem. In building our classifiers, we considered attempting various classifiers and selecting the best classifiers based on the performance of the model on the validation sample set. The classifiers that we will be using are SupportVector Machine, Neural Network, Decision Tree, and AdaBoost.

In the next section, we will analyze the effect that parameters in each of the classifiers will have on the prediction performance. From the analytical section, we will then select the best configuration from each classifier. In evaluation process, we will compute the confusion matrix and calculate the accuracy, precision, and recall rate.

IV. ANALYSIS

A. Support Vector Machine

We are using 2 types of SVM models, which are Linear SVM and RBF SVM. On each model, we try to change tolerance of each model.

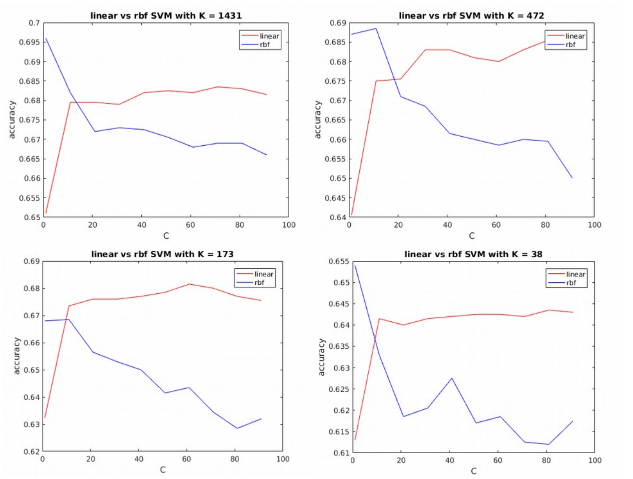

Figure 1. Linear and RBF SVM, accuracy vs C plots. From top-down, left-right, K = 1431, 472, 173, and 38 respectively.

In Figure 1 ,accuracy of linear SVM and RBF SVM over changing tolerance C. In general, accuracy of SVM model falls between 60% to 70%, increasing C would lead to reduce accuracy of RBF SVM but increasing accuracy of Linear SVM. For C bigger than 20, increase C does not affect or increase much to accuracy of linear SVM. The appropriate C for SVM model would be smaller than 20. Depending how many dimension we used for training, the best model would be affected for SVM (3 out of 4 cases the best model is RBF SVM).

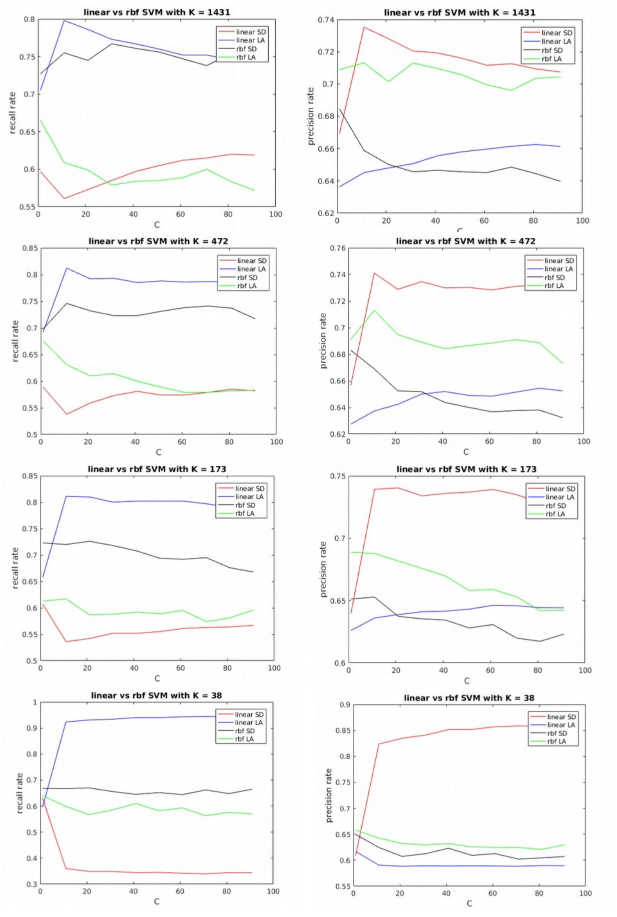

Figure 2. Linear and RBF SVM, recall vs C plots on the left. Precision vs C plots on the right. From top-down K=1431, 472, 173, 38.

For Linear SVM, recall rate for LA is much higher than recall rate for SD but precision for LA is much lower than precision for SD. However, RBF SVM has an opposite situation where recall rate for LA is lower than recall rate for SD and precision for LA is higher than precision for SD. This pattern is persistent throughout many cases as C increased.

In general, for Linear SVM, recall rate for SD is 65% or lower (from 35% to 65%) and recall rate for LA is 60% or higher (from 60% to 95%), precision for SD is 60% or higher (from 60% to 86%) and precision for LA is 67% or lower (from 60% to 67%). For RBF model, recall rate for SD is 60% or higher (from 60% to 80%) and recall rate for LA is 67% or lower (from 56% to 67%), precision for SD is 69% or lower (from 60% to 69%) and precision for LA is 63% or higher (from 63% to 72%).

B. Neural Networks

We are using different combination of activation function and output function for neural network.

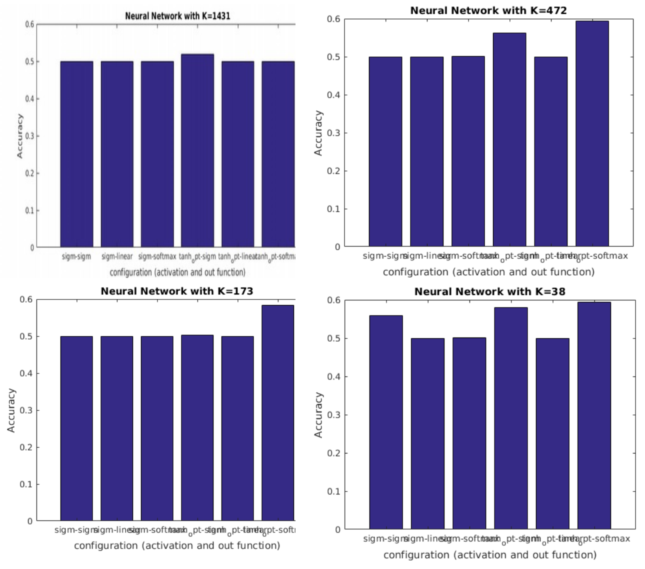

Figure 3. Neural Network, Accuracy vs configuration. From top-down and left-right, K=1431, 472, 173, 38.

The overall pictures for neural network is that neural network does not work well for this classification. Accuracy are 50% for most cases and the best accuracy is about 60% (activation function is tanh opt and output function is softmax). The reason why accuracy is 50% is that all of the prediction are all SD or all LA.

C. Decision Tree

We create decision tree by using Matlab function which optimize all the parameters. These are output figures of decision tree corresponding to number of dimensions 1431, 472, 173 and 38).

D. Adaboost

We run adaboost with different number of iteration and number of dimension. We did not run adaboost with K = 1431 due to our hardware problem.

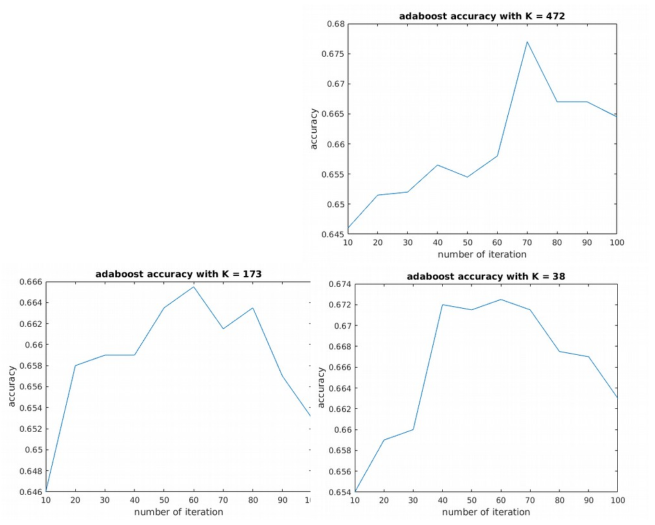

Figure 4. Adaboost. Accuracy vs number of iteration. Top-down, left-right K=472, 173, and 38.

Accuracy for this model is ranged from 64% to 68%. Moreover, changing number of iteration does not affect much

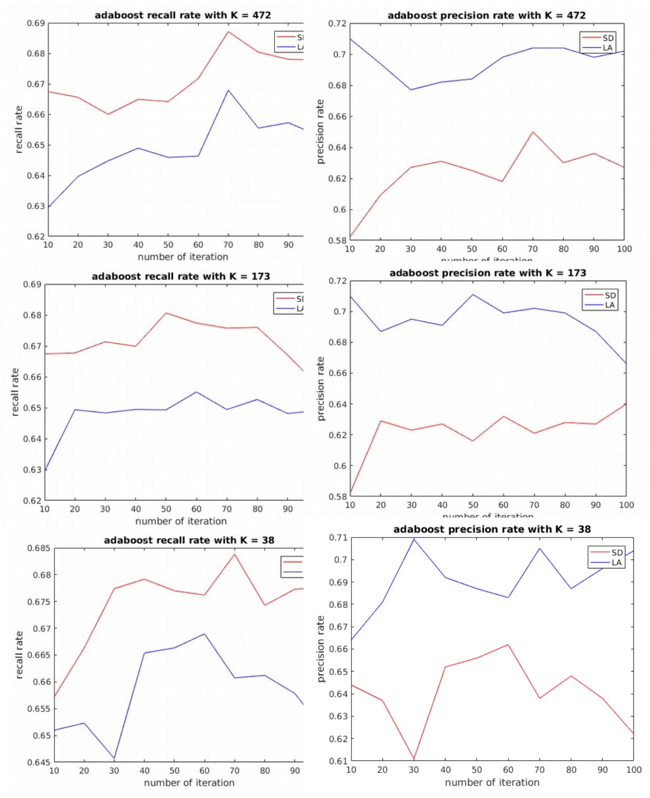

Figure 5. Adaboost. Recall Rate on left-side. Presicion Rate on right-side. Top-down is K=472, 173, and 38.

to accuracy and increasing or reducing number of dimensions does not help to increase significantly the best accuracy of this model (only 1% difference between best accuracy). For all cases, recall rate for SD (ranged from 65% to 69%) is higher than recall rate for LA(ranged from 63% to 67%) and precision for LA(ranged from 66% to 71%) is higher than precision for SD(ranged from 58% to 66%)

V. RESULTS

These are comparison of accuracy of best model of each type of model (SVM, neural network, decision tree and adaboost) with different training number of dimensions on evaluation sample. The best accuracy we get is 70% with K = 1431 for RBF SVM. Depending on how many dimensions we use, the best type of model would be different. For big number of dimension (K = 137 or more), best model would be SVM and for small number of dimension (K = 37), best model is adaboost. It is clear that decision tree and neural network does not work well on this classification. Again, we did not run adaboost for K = 1431, this is why accuracy for adaboost which K = 1431 is 0.

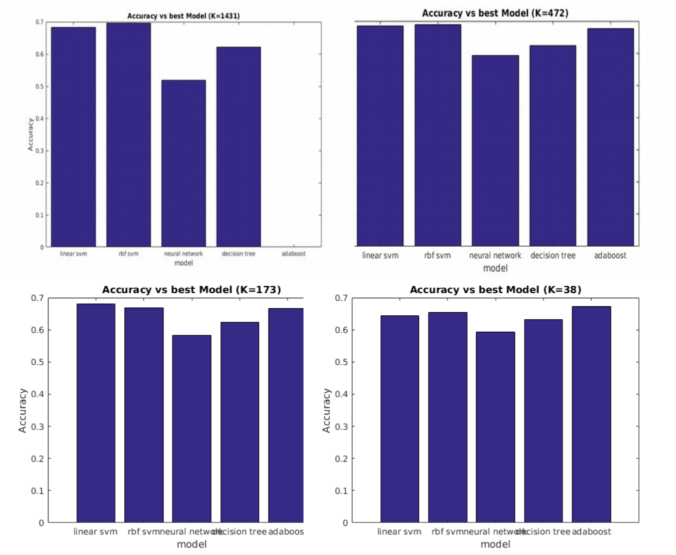

Figure 6. Accuracy vs Models. Top-down and Left-Right is K=1431, 472, 173, and 38.

Best Model	RBF SVM	
K (dimension)	1431	
Accuracy	0.6755	
Confusion Matrix	715	285
	364	636
	SD	LA
Precision Rate	0.6627	0.6906
Recall Rate	0.7150	0.6360

Figure 5. Best Model is RBF SVM performance results.

VI. CONCLUSION

In this project, we proposed a system to predict a Twitter user's location region by building a collection of profile of Bag of Words for Twitter users and applying various classifiers on the profiles. In the process, we evaluated the effects that different configuration of the classifiers have on the prediction 's performance.

From the comparison of different classifiers and their configurations, we identified that the best model is RBF SVM. In different instances with various word feature size, RBF SVM has the highest accuracy, which is marginally better than other models.

From applying the best model that we have found on the test set, we have an accuracy of 0.675, precision rate of 0.66, 0.69 and recall rate of 0.71, 0.63 in San Diego and Los Angeles respectively shown in Figure 5.

Items that we could have improved on would to construct our own dictionary for Bag of Words. The 10,000 common words that we used may not have include slang or exclusive words relevant to its region. Another item that we could have done differently was to choose two locations that were more distant. San Diego and Los Angeles may not be a good representative of this developing proposed work. Two distant regions that could have been considered may include a location on the west coast and another on the east coast. Additionally, we would like to run more configurations for each of the classifiers to select the best model. For instance, considering more layer structure, learning rate, learning rate scaling and batch size.

If there was extra time, other implementations that we would like to attempt are bagging and other classifiers such as Rule-based method and Naive Bayes ,Nearest Neighbor.

We hope that the work and the findings will provide new insight for those considering studies in the area of prediction for social media data.

References

[1] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence
in twitter: The million follower fallacy. In 4th International AAAI Conference on
Weblogs and Social Media (ICWSM), 2010.

[2] M. Abbasi, and H. Liu, Measuring User Credibility in Social Media, poster paper, International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction, April 2-5, 2013. Washington, D.C