An Exploration of Twitter Feed Sentiment Analysis Methods

Christian Gao
Student Department of Statistics
University of California
Los Angeles, California 95616
Email: christiansgao@ucla.edu

1 Introduction

Microblogging today has become a very useful source live information. Particularly for large companies and celebrities, it is a great place to mine for data regarding people's opinion on a certain matter. To make use of this data, we must be able to at least extract the polarity of a tweet so that we can see people's views of certain matters in aggregate. In this project we attempt to reproduce some of the basic methods to classify sentiment. The data set we are using consists of 1,500,000 classified tweets aggregated from multiple sources.

2 About the Data

The data we are using for this project is from Twitter, a Microblogging sevice. Each Tweet is limited to 140 characters. Because of its casual nature, tweets are often prone to acronym usage and misspellings. They also include features such as emoticons: facial expressions depicted with letters and punctuation, targets: an symbol followed by the senders target, and hashtag: an # symbol followed by the topic of the tweet.

There are only two levels of labels in this dataset- positive and negative sentiment. 1 for positive 0 for negative. Since this data is aggregated from multiple sources, some of the data was manually labeled while other tweets were labeled by assuming sentiment based on emoticons and then having the emoticons removed. Everything apart from the sentiment and the raw tweet used in this study are generated features from the tweet itself and not included in the original dataset.

Table 1. Features for sentiment anaysis

Feature	Example
First Negative Word	hate
# of Negative Words	1
1st Positive Word	love
# of Positive Words	2
1st Appearance of Emoticon	:)
# of Emoticons	2

3 Feature Generation

For this analysis we generated a list of simple features to aid us in sentiment analysis. For an example tweet- "Hi! :) XD This is an example. Love is abundant, do you like winter? Winter is coming. I hate winter. @JustinBeiber #WinterIsComing" in Table 1. we list of some of the basic features we used:

4 Logistic Regression Modeling

We shall first explore the logistic regression model as a good baseline for our predictions. We try to dupicate the methods described in [**?**][1]. We have experimented with parameter tuning in order to optimize our predictions. In table two we have our results. It seems that the model responds better to lower lambda values meaning our features are not as noisy as well originally thought. Even though there are more than one million observations in our training data, logistic regression using the H2o package is still very fast. The prediction rates from this test are not very different from the rates mentioned in the paper.

Table 2. Prediction results for Logistic Regression

Lambda Parameter	AUC	elapsed time
0	.77	4 seconds
.001	.74	3.5
.4	.5	3 seconds

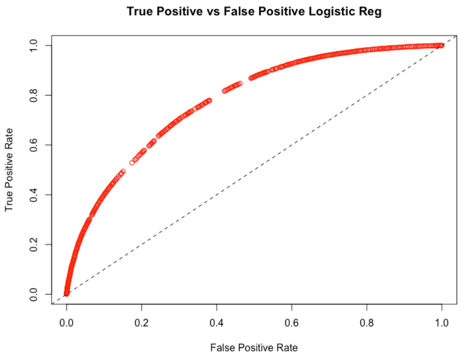

Fig. 1. ROC curve for logistic regression.

Table 3. Prediction results for GBM

Param	AUC	elapsed time
Base	.76	101 seconds
Trees Incr	.77	101 seconds
Rate Incr	.77	110 seconds
Depth Incr	.75	303 seconds
Total	.78	380 seconds

5 GBM Modeling

The second model we shall try is Gradient Boosting Machine (GBM). In this section we shall try and imitate the methods mentioned in [?][2]. The parameters that we will try to tune are the number of trees, the learning rate, and the depth of the trees. We have also introduced stoppping metrics to the larger trees in an effort to save time. After looking at the results. It seems that the GBM with the best AUC is the model with many trees, a low learning rate, and mid sized depth cap. Even though GBM outperformed logistic regression, the increase in AUC was very disproportional to the amount of time it took for the model to train. The base model grows 50 trees, has a max dept of 5, and a learning rate of .1

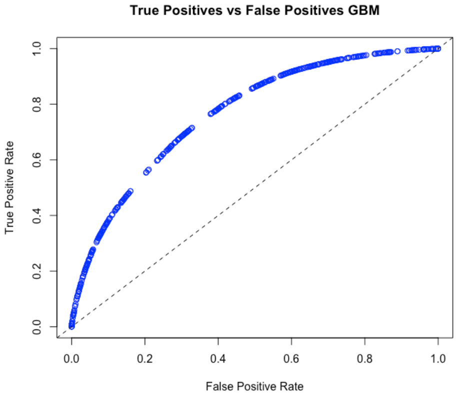

Fig. 2. ROC curve for logistic regression.

Table 4. Prediction results for RF

Param	AUC	elapsed time
Base	.64	101 seconds
Trees Incr	.77	406 seconds
Depth Incr	.71	213 seconds
Total	.77	625 seconds

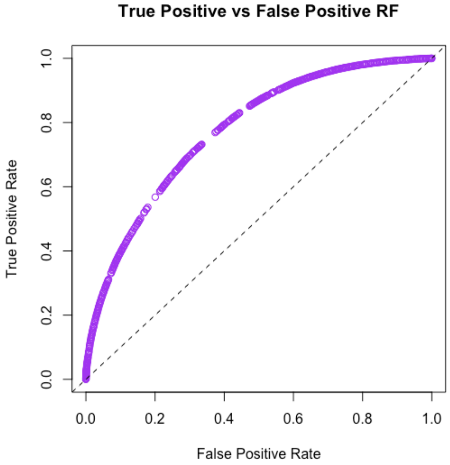

Fig. 3. ROC curve for logistic regression.

6 Random Forest Modeling

The third method we tried is Random Forest. The parameters we have tuned are the tree's max depth, number of trees to select from, and stopping parameters. It seems that out of all the algorithms, random forest is by far the slowest and it has not been able produce the best results.

6.1 Conclusion

In all of the experiments it seems like there is an auc cap of about 80% this is not inconsistent with most of the results of publications we have seen. However, it seems like most of the problem lies with identifying negative tweets. In most cases our ability to identify negative tweets is very high but our ability to identify positive tweets is low. This may be due to the imbalances of positive and negative tweets in our dataset. For future work, perhapes we could engineer some features specifically with that in mind.

References

[1] Huang, C. L. Twitter sentiment analysis.

[2] Vasileios Athanasiou. A Novel, Gradient Boosting Framework for Sentiment Analysis in Languages. See also URL www.mdpi.com/1999-4893/10/1/34/pdf.