Classifying Political Affiliation of Twitter Users using Sentiment Analysis and Topic Models

Saurabh Katkar, Mitali Nandargikar and Akshaya Hebbar

Department of Computer Science

Illinois Institute of Technology

Chicago, IL 60616

{skatkar, mnandarg, ahebbar1}@hawk.iit.edu

Abstract

The advent of Ebola virus in Africa and the news of a few cases being reported in the US has brought a cause of concern escalating to the level of panic among some people. Usually a person's views are shaped by the media and the personalities that he/she follows. Microblogs such as Twitter are subsequently being used as medium of depicting the feelings and reactions that a person harbors. Using sentiment analysis and topic modeling, we analyze the sentiment depicting the level of concern by examining the tweets of such users by taking into account their political ideology. We compare different classification algorithms used for predicting the political ideology given some by testing the accuracy, precision and error rate of the classifiers.

Introduction

As Ebola has taken more lives and crept into more countries, the virus has come to dominate both news headlines and social media conversation. Reports on news channels and views of political figures and popular media personalities on the disease have raised awareness about the disease amongst the people. However the concern about Ebola outbreak in Africa along with four confirmed cases of the disease in the U.S. has culminated into worry and panic among some sections of the population, while other sections are not comparatively less worried about the outbreak. Reports on the news media and news sites and opinions of media personalities aligned with a certain political party often sways a person's level of concern on the issue. Thus the study of a user's opinions inclined with a political ideology will help gauge the level of concern amongst the demographics of that political party. Thus for this report we consider the following **hypothesis**;

"A user's views and concerns about the Ebola virus might be influenced by the news the user is subjected to, the political, media personalities and close friends and their say on the issue. The said user's tweet will subsequently reflect the level of concern that he/she associates with the virus. "

The scientific community has made a great effort to provide effective solutions to analyse, structure, and process the large amount of on-line reviews in social media. A wide set of techniques of Sentiment Analysis (SA) are used in micro-blogging texts to extract the polarity (positive, negative, mixed or neutral) that users express in these texts. In this respect, Twitter has become a popular micro-blogging site in which users express their opinions on a variety of topics in real time. The texts used in Twitter are called tweets, which are short texts of a maximum of 140 characters and a language that does not have any restriction on the form and content. The nature of these texts poses new challenges for researchers in Natural Language Processing (NLP).

Thus, using sentiment analysis and topic modeling, we analyze the sentiment depicting the level of concern by examining the tweets of such users and coupled with lexicon analysis we determine the political affiliation of the user. For the purpose of classification we determine the association of a user with different political and media personalities that we know are aligned with a particular political party and we test the derived dataset and classify the data using Logistic Regression and Naïve Bayes to compare the two classification algorithms based on different performance measures.

Background

The Twitter Platform

Twitter is a popular social networking and microblogging site where users can broadcast short messages called 'tweets' to a global audience. A key feature of this platform is that, by default, each user's stream of real-time posts is public. This fact, combined with its substantial population of users, renders twitter an extremely valuable resource for commercial and political data mining and research applications.

The free-form nature of the platform, combined with its space limitations and resulting annotation vocabulary, have led to a multitude of uses. Some use the service as a forum for personal updates and conversation, others as a platform for receiving and broadcasting real-time news and still others treat it as an outlet for social commentary and critical culture. Of particular interest to this study is the role of Twitter as a platform for political discourse.

Data Mining and Sentiment Analysis

Owing to the fact that Twitter provides a constant stream of real-time updates from around the globe, much research has focused on detecting noteworthy, unexpected events as they rise to prominence in the public feed. The outbreak of Ebola virus is one such example with millions of tweets related to the disease were posted in the U.S. alone.

Another pertinent line of research in this area relates to the application of sentiment analysis techniques to the Twitter corpus. The informal form of communication that takes place on Twitter can be correlated to a person's mood states and help gauge their sentiments.

Data and Methods

The Python Twitter API

This is a Python wrapper around the Twitter API and the Twitter data model. This library provides a pure Python interface for the `Twitter API' (https://dev.twitter.com/). Twitter exposes a `web services API' and this library is intended to make it even easier for Python programmers to use. The API class provides access to the entire twitter RESTful API methods which would be needed for the purpose of this project to fetch user information and user tweets. Twitter offers two types of authentication: **OAuth 1** and **OAuth 2** authentication. For the purpose of this project, we use OAuth 2 as it is what is required to make read-only calls to Twitter, i.e. searching, reading a public user's timeline.

Fetching Tweets

This analysis focuses on three weeks of twitter data collected using Ebola as the primary search term. The data covers tweets produced during the period between October 19th and November 8th, 2014, when news about the Ebola outbreak was quite prevalent in the media.

Harvesting Common Followers

We took in consideration the association between the political personalities and their followers with the intention of harvesting the tweets of the followers. Initially we took into account 15 different political and media personalities that have a pre-defined alignment with a political party. In this regard the twitter users that were considered for examination had followers exceeding 200 thousand so that substantial amount of followers and their tweets would be derived. Subsequently the followers of these personalities were harvested using the followers' user ID and the followers common to 7 out of the 15 twitter users were considered. This was done so as to help derive the conclusion, based on association, the political alignment of the followers.

Subsequently, the **GET followers/ids** method of the Twitter API was used which returns a cursored collection of user IDs for every user following the specified user. Using this method, results are given in groups of 5,000 user IDs and multiple “pages” of results can be navigated through using the **next_cursor** value in subsequent requests.

Harvesting the Target Tweets

Once the common followers of the political personalities were harvested, we examined the user timeline for tweets related to the topic of this project’s interest. The **GET statuses/user_timeline** method of the Twitter API returns a collection of the most recent Tweets posted by the user indicated by the **screen_name** or **user_id** parameters. This method was used to fetch the most recent tweets of the twitter users derived. Once the tweets were fetched, they are stored in a list and the search method of regular expression (regex) applied to filter the tweets concerning Ebola. Thus this methodology was used to harvest the tweets from common users using the search term ‘Ebola’.

Explicit Labelling of the Followers

Since twitter users considered for examination do not have a pre-defined label determining their political ideology, we had to devise a scheme as to draw a conclusion on what political ideology the twitter user aligns to. For this purpose, we initially considered a set of twitter users who are political representatives of a specific party or media personalities who have vocally shown their support in favor of a political party. The followers of these personalities were derived and a subset of common followers, wherein the user should be following at least 7 of the political personalities, were taken into consideration. For example, if a user is following 10 out of the 15 republican personalities, then that user would be labelled ‘Republican’ (or 1 in our dataset). However if a user is following 5 out of 15 Democratic personalities, then that person would not be labelled ‘Democrat’ (or 0 in our dataset). Thus in this way by association we explicitly labelled the common followers.

Computing the Sentiment Score

On the Twitter corpus derived, we performed sentiment analysis to understand the nature of the tweet reflected through its score, which can be positive, negative or neutral which relates to the tweet’s expressed sentiment. For the purpose of sentiment analysis, the **AFINN** wordlist was used which is a list of English words rated for valence with an integer between -5 (negative) and +5 (positive). Using AFINN, the scores of the words used in the tweet were computed and correspondingly their sum derived to get the score of the tweet. This technique was used on all the tweets derived and scores were stored in a dataset for further analysis using classification and clustering.

Accordingly, various other attributes were computed for the dataset. They included deriving the gender of the twitter user which was done using the **GET users/lookup** method of the Twitter API in order to get the name of the twitter user and then extrapolating the gender information using **United States Census Data** (www2.census.gov/topics/genealogy/1990surnames).

Along with the gender, other data that was derived included the numbers of word in a tweet and frequency of different words in a tweet. Word frequency was considered because a contributor's perspective bias is displayed through their lexical choice. Someone who is showing a high level of concern would not use words such as ‘panic’, but mostly words such as ‘concern’, ‘worried’, ‘alert’ which depicts a measured tone of seriousness. Words such as panic depict a slightly exaggerated tone and as such would be generally used by a person depicting low level of concern on the issue.

Related Experiments

Predicting Labels using Generative and Discriminative Classifiers

Generative and discriminative learning are two of the major paradigms for solving prediction problems in machine learning, each offering important distinct advantages. These algorithm utilize a vastly different technique from each other in solving the classification problem and have their own advantages and drawbacks with respect to the other approach.

Generative Classifiers

Generative classifiers, such as Normal-based Discriminant Analysis and the Naive Bayes classifier, model the joint distribution P(x, y) of the measured features x and the class labels y factorized in the form P(x|y)P(y), and learn the model parameters through maximization of the likelihood given by P(x|y)P(y).

$$c_{map} = \arg\max_{c \in C} \left(P(c \mid d)\right) = \arg\max_{c \in C} \left(P(c) \prod_{1 \le k \le n_d} P(t_k \mid c) \right)$$

Fig: *Formula for the Naïve Bayes Classifier*

In other words, a generative classifier tries to learn the model that generates the data behind the scenes by estimating the assumptions and distributions of the model. It then uses this to predict unseen data, because it assumes the model that was learned captures the real model.

Discriminative Classifiers

Discriminative classifiers, such as logistic regression, model the conditional distribution P(y|x) of the class labels given the features, and learn the model parameters through maximizing the conditional likelihood based on P(y|x).

$$\theta_i = \frac{1}{1 + \exp\left[-\left(\beta_0 + \sum_{j=1}^{k} \beta_j x_{ij}\right)\right]}$$

Fig: *Formula for the Logistic Classifier*

A discriminative classifier tries to model by just depending on the observed data. It makes fewer assumptions on the distributions but depends heavily on the quality of the data

Algorithm Analysis

Generative models allow you to make explicit claims about the process that underlies a dataset. For example, generative graphical models allow you to describe conditional dependencies between model parameters. If your model has a good fit to your data set, it strengthens your claim that your model accurately reflects the generative process that actually created the data that you are modeling.

Generative classifiers learn about the conditional probability indirectly, they can get the wrong assumptions of the data distribution. Quoting Vapnik from Statistical Learning Theory –

"One should solve the [classification] problem directly and never solve a more general problem as an intermediate step [such as modeling P(X|Y)]."

An important contribution to this topic is from Andrew Ng and Michael Jordan presenting some theoretical and empirical comparisons between linear logistic regression and the Naive Bayes classifier. Their results suggested that, between the two classifiers, there were two distinct regimes of discriminant performance with respect to the training-set size. More precisely, they proposed that the discriminative classifier had lower asymptotic error rate while the generative classifier may approach its (higher) asymptotic error rate much faster.

In other words, the discriminative classifier performs better with larger training sets while the generative classifier does better with smaller training sets.

Implementation of Algorithm

With the aim of reporting the experimental results and inferring on the data by observing the performance measures of the algorithm, the aforementioned datasets were used and the results were derived using the Logistic Regression and Naïve Bayes Algorithm. The dimensions of the datasets used are given the following table:

Dataset	Instances(m)	Features(N)

Only Ebola	2000	8
Ebola+other sentiments	2000	18

Table: *Dimensions of the Ebola dataset*

We apply **gradient optimization** technique to find the local optima of the function in the algorithm. Taking an initial guess, the algorithm will go either to the negative or the positive direction of the gradient to optimize the gradient. This process is repeated till the algorithm converges

They are both first-order algorithms because they take only the first derivative of the function.

R already has built-in functions for performing Logistic Regression and Naïve Bayes classification on the provided datasets. Logistic regression is implemented by an R function glm from a standard package stats in R, and the Naïve Bayes classifier is implemented by an R function Naive Bayes from a contributed package e1071 for R. However, for the purpose of this experiment the classifiers were codified without the use of these classification packages and various performance measures such as Accuracy, Precision, Recall, and F-score were derived.

Experimental Observations

Upon executing the algorithm, the performance measures by which to gauge the efficiency of the algorithm are derived for comparing the asymptotic efficiency of Logistic Regression algorithm versus the Naïve Bayes Classifier.

For deriving the time taken by the classifiers, a built-in R function called ***system.time()*** is used. This function essentially times how fast R processes an expression. Using this function, the differences in speed of Logistic Regression and Naïve Bayes algorithm can be computed and compared and appropriate calculations can be made.

Logistic Regression Time Elapsed			
	user	**system**	**elapsed**
Ebola	25.41	0.08	25.64

Fig: *Time taken for the execution of Logistic Regression Algorithm*

Naïve Bayes Time Elapsed			
	user	**system**	**elapsed**
Ebola	15.02	0.1	15.9

Fig: *Time taken for the execution of Naïve Bayes Algorithm*

Upon executing the algorithm it was observed that Naïve Bayes approaches its asymptotic error without the need for a large number of training examples, and it does so very quickly. Logistic regression, on the other hand, is capable of outperforming naive Bayes, given the number of training examples is large enough.

However, if the number of training examples is very large, then using Logistic regression is comparatively costly, since the parameters of the algorithm are needed to be optimized using gradient optimization. As a rule of thumb, Naive Bayes will almost surely outperform logistic regression if the number of training examples is small.

Inference

Upon executing the algorithm it was observed that Naïve Bayes approaches its asymptotic error without the need for a large number of training examples, and it does so very quickly.

Logistic regression, on the other hand, is capable of outperforming naive Bayes, given the number of training examples is large enough. However, if the number of training examples is very large, then using Logistic regression is comparatively costly, since the parameters of the algorithm are needed to be optimized using gradient optimization.

As a rule of thumb, Naive Bayes will almost surely outperform logistic regression if the number of training examples is small. Further, it will reach its asymptotic error very quickly, making it much more desirable in such a scenario than logistic regression.

Thus, if Naïve Bayes is outperforming Logistic Regression by a large margin, then it is a safe bet to stick with Naive Bayes until a much larger amount of training examples are available. However, if it is observed that Naive Bayes is not outperforming logistic regression by much, then it can be deduced that Logistic Regression will outperform Naive Bayes within a relatively small amount of new training examples.

The efficiency of an algorithm can be characterized by its performance measures such as *Accuracy, Precision, Recall, F1 Score* etc. The different performance measures for Logistic Regression Algorithm were reported as follows:

Logistic Regression Performance Measures							
Dataset	**Accuracy**	**Precision1**	**Precision2**	**Recall1**	**Recall2**	**F1**	**F2**
Only Ebola	0.612	0.837	0.736	0.889	0.816	0.872	0.726
Many Senti	0.814	0.833	0.901	0.935	0.855	0.820	0.961

Fig: *Logistic Regression characterized by its performance measures.*

Naïve Bayes Performance Measures							
Dataset	**Accuracy**	**Precision1**	**Precision2**	**Recall1**	**Recall2**	**F1**	**F2**
Only Ebola	0.35	0.855	0.215	0.543	0.765	0.687	0.335
Many Senti	0.65	0.409	0.616	0.891	0.747	0.687	0.561

Fig: *Naïve Bayes characterized by its performance measures.*

For calculating the mean error rate, we take into account the number of wrongly classified data against the number of training examples considered to fit the data.

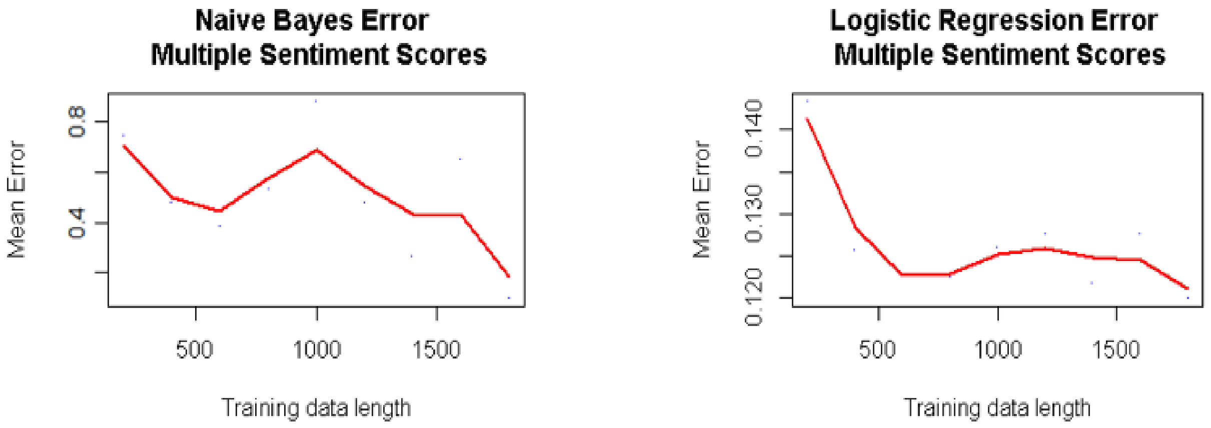

Fig3:.*Plots of misclassification error rate vs. training-set size m for Ebola dataset, with regards to Naïve Bayes and Logistic Regression.*

Conclusion

We found that only using 'Ebola' as a search term used for classifying users by their political affiliations is not as effective as using terms that reflects on ideologies of a political party. As such we found that including a number of term imbibed in the lexicon of a political ideology gave an improvement in the accuracy and precision scores of the Logistic Regression and Naïve Bayes classification algorithms.

Other attributes considered such as gender, and frequency of certain 'catch terms' played an important role in classification as they were quite reflective of the demographic and the lexicon of a political pary. For example we found that there were more women in the Democratic side as compared to the Republican side in our dataset.

Naive Bayes and Logistic Regression are a "generative-discriminative pair," meaning they have the same model form (a linear classifier), but they estimate parameters in different ways.

For feature x and label y, **Naive Bayes** estimates a joint probability **p(x,y) = p(y)*p(x|y)** from the training data (that is, builds a model that could "generate" the data), and uses Bayes Rule to predict p(y|x) for new test instances. On the other hand, **logistic regression** estimates **p(y|x)** directly from the training data by minimizing an error function (which is more "discriminative").

These differences have implications for error rate:

When there are very few training instances, logistic regression might "overfit," because there isn't enough data to estimate p(y|x) reliably. Naive Bayes might do better because it models the entire joint distribution.

When the feature set is large (and sparse, like word features in text classification) naive Bayes might "double count" features that are correlated with each other, because it assumes that each p(x|y) event is independent, when they are not. Logistic regression can do a better job by naturally "splitting the difference" among these correlated features.

If the features really are (mostly) conditionally independent, both models might actually improve with more and more features, provided there are enough data instances. The problem comes when the training set size is small relative to the number of features. Priors on naive Bayes feature parameters, or regularization methods (like L1/Lasso or L2/Ridge) on logistic regression can help in these cases.

Future Work

For this experiment, AFINN wordlist was considered, which only offers scores of about 3000 words. In future experiment it would be beneficial to use a more vast wordlist or a sentiment analysis API which would provide for a more fine-tuned score of the harvested tweets. This could in return result in a better classification accuracy on the classifiers used.

More attributes such as ethnicity, geolocation, age derived may also help in the process of improving the prediction of classification.

On the empirical side, combinations of discriminative and generative methodologies have been explored in many fields such as natural language processing, speech recognition, and computer vision.

In particular, the recent "deep learning" revolution of neural networks relies heavily on a hybrid generative-discriminative approach: an unsupervised generative learning phase ("pre-training") is followed by discriminative fine-tuning. Given these recent trends, a workshop on the interplay of generative and discriminative learning seem especially relevant.

Hybrid generative-discriminative techniques face computational challenges. For some models, training these hybrids is akin to the discriminative training of generative models, which is a notoriously hard problem. Alternatively, the use of generative models in predictive settings has been be explored.

References

1. *On Discriminative vs. Generative classifiers: A comparison of Logistic Regression and Naive Bayes*, by Andrew Y. Ng , Michael I. Jordan, 2001, Neural Information Processing Systems Conference (NIPS)-14

2. *Modeling Microblogs using Topic Models* by Kirti Puniyani, 2007, Carnegie Mellon University

3. *Predicting the Political Alignment of Twitter Users* by M Conover, B Goncalves, J Ratkiewicz, A Flammini, and F Menczer. Proceedings of 3rd IEEE Conference on Social Computing SocialCom, (2011)

4. *Political Tendency Identification in Twitter using Sentiment Analysis Techniques* by Pla Ferran and Lluís-F. Hurtado, The 25th International Conference on Computational Linguistics (COLING 2014)

5. *Discriminative vs. generative learning: which one is more efficient?* by Philip M. Long, Rocco Servedio and Hans Ulrich Simon, Information Processing Letters, 103 (4), 2007

6. *Discriminative versus Generative Parameter and Structure Learning of Bayesian Network Classifiers,* by Franz Pernkopf, Jeff Bilmes, ICML '05 Proceedings of the 22nd international conference on Machine learning, 2005