Spring 5-1-2015

Twitter Bot Detection

Bryan Holster

TWITTER BOT DETECTION

An honors paper submitted to the Department of Computer Science
of the University of Mary Washington
in partial fulfillment of the requirements for Departmental Honors

Bryan Holster
May 2015

By signing your name below, you affirm that this work is the complete and final version of your paper submitted in partial fulfillment of a degree from the University of Mary Washington. You affirm the University of Mary Washington honor pledge: "I hereby declare upon my word of honor that I have neither given nor received unauthorized help on this work."

Bryan Holster
(digital signature)

05/01/15

Honors Thesis

Twitter Bot Detection

Bryan Holster

Department of Computer Science

Stephen Davies, Honors Thesis Advisor

April 27, 2015

Abstract

In this thesis, I explore the identification of non-human Twitter users. I am interested in classifying users by behavior into the categories of either *bot* or *human*. My goal in this research is to find an accurate and efficient means of identifying and segregating non-human Twitter users from their human counterparts.

I use a two-stage data collection process to collect Twitter users suspected of being a bot and then obtain a majority vote on the suspected users to validate the suspicion. I gather, on average, 1000 tweets per user, on which I calculate 40 features characterizing the user. I explore the effectiveness of three different methods to most accurately classify users as either a bot or a human based on these features.

The results of this work show that bots can be classified efficiently and with a high degree of accuracy. I show that certain features play a larger role in the classification process than others. The applications of Twitter bot identification include: (i) protecting users from malicious content (ii) spam filtering, and (iii) bot removal from Twitter data for other research.

Table of Contents

Abstract i

1.Introduction **1**

1.1 Perspective 2

1.2 Objective 3

2. Existing Work **4**

2.1 Methods 4

3. The Data **8**

3.1 What it Means to Be a Bot 8

3.2 Gathering the Data 9

4. Analysis **12**

4.1 Feature Extraction 12

4.2 Methods 13

4.3 Performance 15

5. Conclusions **20**

References **21**

1. Introduction

The popularity of social media has created many avenues for sharing, as well as acquiring, information. Twitter, a popular social media tool that facilitates sending short text-based messages, has roughly 288 million active users, with an average of 500 million tweets per day in 2013[1]. This represents a six orders of magnitude increase from the average of 5,000 tweets per day in 2007[2]. The amount of time per day some individuals spend communicating via social media easily exceeds the time they spend in face-to-face communication.

There are many advantages of ever growing online communities such as Twitter, however, they come at a cost. One advantage in particular is the reach a single user's tweet has the potential to have. The "Six Degrees of Separation" theory, created by Frigyes Karinthy in the early 20th century, states that through a chain of contacts each and every person on the planet is six or fewer introductions away from any other person [Frigyes 1929]. Modern research exploring this theory found that, on average, Twitter is a network with only five degrees of separation [Cheng 2010].

The cost of this reach is the enablement of rumors, general misinformation, and political slander to spread very quickly. Automated Twitter accounts, or bots, pose a threat to the general public. They have the potential to take advantage of the aforementioned negative uses of Twitter to cause panic during emergencies, or even hinder public policy. This is not to say that all bots have malicious intent, but instead to

[1] "About Twitter, Inc.", Accessed April 27, 2015. https://about.twitter.com/company
[2] "Twitter Usage Statistics", Accessed April 27, 2015. http://www.internetlivestats.com/twitter-statistics/

demonstrate why fast and efficient social media bot detection is of importance.

1.1 Perspective

This research is concerned with the classification of Twitter users into the categories of either bot or human. In order to accomplish this we must begin from the perspective of understanding user behavior.

The most central part of Twitter is the user base. Users interact with other users by generating content and taking other actions supported by Twitter. There are many different roles a user can assume within the Twitter community. There are those seeking to contribute information and those seeking information. There are also those seeking to create turmoil within the community and those trying to stop them [Pal 2012]. These all represent underlying reasons for any user to be involved in any online community such as Twitter.

These roles do not specifically define behaviors useable to classify users into the broader categories of *bot* and *human*. Instead the category in which a user belongs can be identified by looking at their underlying characteristics. We can look at how often a user tweets, the variance in content of their tweet, and how much interaction the user has with other users. The primary focus of this research is on feature extraction from user behavior that could lead to the proper categorical classification of Twitter users. I extract many behaviors, such as the mean time between tweets, number of retweets, and number of embedded links within their tweets. Behaviors such as these can provide insight as to which class the user belongs. These features are used to build a profile for each user, which can be used as input for computational algorithms to classify each user as a *bot* or a *human*.

An important step in this research was algorithm selection. In this research, I used several types of machine learning algorithms such as Random Forest [Breiman 2001], k-Nearest Neighbor [Cover et al. 1967], and Linear Discriminant Analysis [Fisher 1936]. These algorithms were chosen because each takes a distinctly different approach to classification. In this thesis, I focus on the application of these machine learning algorithms on the extracted user features.

1.2 Objective

While Twitter may be a source of entertainment for some, or a way to read the news for others, it is a wealth of data for a data scientist. The objective of this research is to provide a method for predicting the class of a user as being either *bot* or *human*.

In section 3, I assess what it means to be a non-human Twitter user and how I gathered our user data. How I define the difference between a bot and human user directly impacts how effective a classifier can be, as well as the quality of the training and test data sets.

In section 4, I will analyze different classifiers and their performances to determine which is most effective. I employ machine learning algorithms such as Random Forest [Breiman 2001], k-Nearest Neighbor [Cover et al. 1967], and Linear Discriminant Analysis [Fisher 1936]. I model the margin of error for each and present their advantages and disadvantages.

2. Existing Work

The strategies currently employed by Twitter to specifically detect bots are inadequate at best. The reason for this is that Twitter allows automated accounts. They look for outliers that break the rules and regulations in such ways as performing a large number of following and unfollowing requests and spamming. This could include both bot and human accounts.

2.1 Methods

In order to understand this growing phenomenon of bot activity, beginning in 2009 [Chu Z et al. 2012], the academic community began to study their behavior. Researchers have begun taking approaches such as: trying to reverse-engineer social bots to gain a better understanding of how they function [Freitas et al. 2014], while others are creating their own bots [Briscoe et al. 2014; Hwang et al. 2012] to test the susceptibility of human users to the influence of these bots [Wagner et al. 2012; Wald et al. 2013; Boshmaf et al 2013].

An important study validating choices I have made in our research expanded on the possibility of human detection of bots [Wang et al. 2013]. The authors created an Online Social Turing Test which assumed that bot detection is a rather trivial task for a human. The ability of a human to detect nuances and anomalies in language they assumed is still unmatched by machines. They demonstrated that the efficacy of humans to detect bots is high enough to be used in a majority vote system. They found that using this method resulted in a near-zero false classification rate in a 2-Class model. In this case the two classes being *bot* and *human.*

When the scope of social bots is focused on Twitter, a number of researchers prefer to expand from a 2-Class model to a 3-Class model [Tavares et al. 2013;Chu Z et al. 2010]. The research performed by Tavares et al focuses solely on temporal features, specifically inter-tweet delay and tweet time (time of day and day of the week). The authors developed two classification algorithms: a 2-Class algorithm distinguishing between human and managed accounts, and a 3-Class algorithm distinguishing between human, managed and bot accounts.

The managed accounts consist of only accounts maintained by prominent corporations, while the bot accounts were chosen from online lists of Twitter bots. Tavares et al. claim to have not filtered this data in any way after collection. They do, however, later have to throw out non-time zoned accounts from their dataset, reducing it to 86 human accounts, 91 managed accounts and 67 bot accounts. This resulted in a working set of 164975 tweets over 244 accounts.

Tavares et al. came to a number of conclusions: (a) inter-tweet delay distributions were statistically significantly different, (b) managed accounts have higher tweeting activity during work days, (c) human accounts exhibit homogenous behavior throughout the week, and (d) bot accounts show little correlation throughout the week.

The authors' research resulted in a 3-Class classifier with an accuracy of 75.8%$\pm$4.8% and a 2-Class classifier with an accuracy of 84.6%$\pm$2.2%. These accuracies are impressive, considering only two temporal features were used. The resulting accuracies, however, may have been influenced by a bias created by the manner in which they acquired their dataset. As concluded by Torralba et al, data acquired from similar locations (all bots are from online list), or data sharing similar

characteristics (all managed accounts from prominent corporations), will likely exhibit some amount of bias [Torralba et al. 2011].

Natural language processing has begun to play a larger role in bot detection on social media, especially as bots grow more advanced. Researchers have already begun using such methods for bot detection [Chu Z et al. 2010] in conjunction with features such as tweeting device, and the number of followers and friends. Chu Z et al. use a 3-Class model with the classes human, bot and cyborg.

The user is labeled as a human if they exhibit “original, intelligent, specific and human-like contents.” The user is identified as a bot if they: lack the aforementioned contents, display excessive automation of tweeting (automatic updates of blog entries or RSS feeds), display spam or malicious URLs in tweets, repeatedly post duplicate tweets, post links with unrelated tweets, or display aggressive following behavior. Lastly, a user is identified as a cyborg if the account exhibits both human and bot characteristics.

Chu Z et al. process approximately 60 tweets per Twitter account and conclude that bots generally have many friends but few followers. Interestingly, some bots maintain a near 1:1 ratio between friends and followers. The authors assert that this is a sign of an advanced bot, taking precautionary measures against Twitter’s imposed limit on a user’s follower to friend ratio. Lastly they show that bots tend to include URLs in their tweets in hopes of luring human users to external web sites. This is usually for either advertising purposes, or to infect users with malware.

One of the most recent studies done on bot detection was that of [Ferrara et al. 2015] under contract with the DoD. Ferrara et al. achieved an impressive detection rate

of 95%. The authors achieved this feat classifying on a staggering 1,150 unique features. Of these features the most predictive were found to be: number of retweets, age of account, number of tweets, and length of username.

The research resulted in the creation of a web application *Bot or Not?* Which, given a username, provides the likelihood of the user being a bot using six generalized feature categories: *Network, User, Friends, Timing, Content,* and *Sentiment*.

3. The Data

3.1 What It Means to Be a Bot

It is worth defining what exactly it means to be a bot on Twitter. There are three main categories of Twitter users. Researchers have labeled these categories as human, bot. A third category also exist which has been called cyborg [Chu Z et al. 2012]. The human category contains all users who type out their own tweets and post them to Twitter. The bot category contains all users who are fully automated programs. These bots generate tweets at some programmed time interval, or in response to some external trigger. The trigger could be anything from a breaking news article being posted, to a tweet containing the word “Obama”. Once a bot program has begun, there is minimal human interaction from the creator. This distinction is important to this research as it defines a stark contrast between a bot and a cyborg.

The cyborg category contains all users that rarely, if ever, manually post a tweet via Twitter. These users use programs to post the copies of the same message to Twitter, Facebook, and Google+ either for convenience or to share a link and the header of their new blog post. The key difference is that despite using a program to post tweets, each tweet is engineered by a human. The program generates no content on its own, therefore these human users, despite displaying some bot characteristics will be classified as humans in this 2-Class model. This ensures that our training data is populated with what will be referred to as certified bots throughout this thesis.

3.2 Gathering the Data

In order to obtain a sufficient data set, I employed a two stage system. In the first stage, University of Mary Washington students were asked to search for Twitter accounts that they suspected to be bot accounts. They then submitted the web address of the suspected accounts via a web application designed to accept and store only unique accounts. If a student attempted to submit an account previously submitted by another student, it would be rejected and a message would be displayed informing the student. I used this method to ensure that I gathered as many unique potential bot accounts as possible before moving onto stage two.

The second stage of this system utilized a web application that created a pool of Twitter accounts consisting of the 326 suspected bot accounts in stage one as well as a randomly chosen set of 500 unique Twitter accounts. The latter were chosen from collection of over 22 million tweets I had accumulated via a real-time Twitter feed.

University of Mary Washington students, once logged onto the web application, were presented with a URL to a random Twitter account out of the 826 account allotment. Upon visiting the account's Twitter page they had the option to vote either 'yes' or 'no' as to whether or not they determined the account to be a bot. After voting, they were then displayed another random account's URL, and the process repeated. Once an account accumulated enough votes from users it was removed from the available pool. This ensured that I had sufficient voting data across our 826 account allotment.

Once stage two was completed I divided the accounts into four categories: certified bot, majority bot, certified non-bot, and majority non-bot. The category in which

the account was placed was based entirely on the composition of votes the account had accumulated. For an account to have been labeled a certified bot, all votes for the account had to have been in favor of it being a bot. Conversely, a certified non-bot only received votes in favor of it not being a bot. All other accounts labeled as majority bot, and majority non-bot, received majority votes in favor of either being a bot or not being a bot respectively.

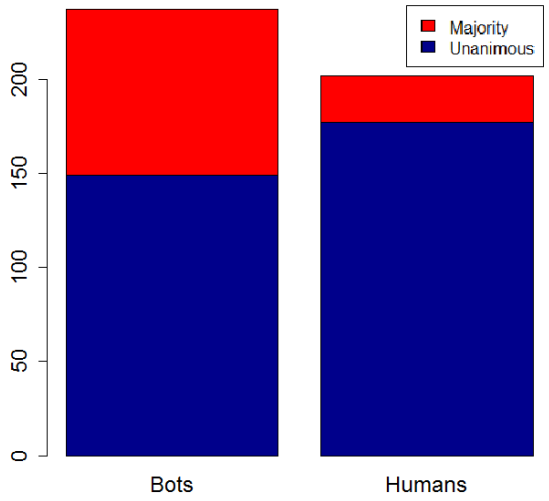

Fig. 3.1 The User Data Distribution stacked bar plot represents the voting results from stage two of our data collection system. The left column representing the bot class, and the right representing the human class. The blue regions represent accounts that were voted unanimously to belong to either the bot or human class. The red regions represent accounts that had at least a ⅔ majority vote to belong to their respective classes.

A common method of data collection for research such as this, as used by [Gianvecchio et al. 2008] and [Sarita et al. 2009], is to use a sufficiently large subset of tweets archived from a real-time feed. In this research, however, instead of analyzing a snapshot of data for a given time frame, I chose to collect the last ~1,000 tweets for each user I had identified.

I gathered this data by creating an application in Python script language that utilized the third party libraries python-twitter, oauth2, httplib and json. The application accessed Twitter via the Twitter Application Programming Interface (API), which allowed for its communication with Twitter. One downside to this approach, however, is that

Twitter only allows 180 requests per 15 minutes via the API. Furthermore, each request may only yield a maximum of 200 tweets, meaning a rate limit of no more than 36,000 tweets per 15 minutes. My application easily circumvents this rate limit by delaying its own requests to ensure accordance with Twitter's rate limit.

This method provided a working set of 441,776 tweets across 439 accounts. By obtaining tweets in this manner I aimed to reduce the likelihood that our extracted features would be skewed by temporal circumstance.

4. Analysis

4.1 Feature Extraction

In collaboration with Dahlgren scientists Dr. David Marchette and Dr. Elizabeth Hohman, I compiled a list of features to extract from the data. The list included many features they had found useful in prior Twitter research, such as the number of times a user says “lol” or “wtf.” The following features were extracted as a means of characterizing user behavior:

- **Account**:
  - *fromMob* - percentage of users tweets which come from a mobile phone
  - *meanF2F* - mean friend to follower ratio across all of users tweets
  - *N* - total number of tweets for the user
  - *nsources* - number of unique sources
  - *sdF2F* - standard deviation of friend to follower ratio
  - *maxFriends*, *minFriends* - max and min number of friends the user had over their last 1000 tweets.
  - *maxFollowers, minFollowers* - max and min number of followers the user had over their last 1000 tweets.
- **Content**:
  - *retweets* - total number of times the user retweeted
  - *alllower* - all letters in the tweet are lowercase
  - *lols* - total number of times “lol” appeared in one of the users tweets
  - *links* - total number of links in the user’s tweets
  - *mentions* - total number of mentions in the user’s tweets
  - *sdLength* - standard deviation in number of characters appearing in each of the user’s tweets
  - *allcaps* - all letters in the tweet are uppercase
  - *fwords, bwords, nwords* - total number of time the word each of these derogatory words were used in the user’ tweets
  - *wtf* - total number of times the user used “wtf”
  - *meanLengthC* - mean number of characters appearing in each of the user’s tweets after having unicode removed
  - *meanLength* - mean number of characters appearing in the user’s tweets
  - *digits* - *total* number of digits used in the user’s tweets
  - *hashes* - *total* number of hashes (#, hashtags) in the user’s tweets
  - *sdLengthC* - standard deviation in number of characters appearing in each of the user’s tweets after having unicode removed
  - *emoticons* – total number of emoticons in the user’s tweets

- **Time**:
  - *SHours* - standard deviation between first and last tweet of the day
  - *DHours* - mean absolute value of the distance between tweets
  - *NHours* - number of hours between first and last tweet of the day
  - *Monday, Tuesday*, *Wednesday, Thursday, Friday, Saturday, Sunday* - total number of times the user has tweeted for each day of the week
  - *deltaTSigma* - standard deviation of the distance between tweets
  - *deltaTMean* - mean distance between tweets

The three categories, *Account, Content*, and *Time* are used in this thesis as a means of conceptually grouping the features used in this research into broader categories. The *Account* features characterize each of the users independently from their tweeting behavior. The *Content* features characterize aspects of each user's tweets in respect to how they are organized and what they contain. The *Time* features characterize the temporal features relating to the user's tweets.

A total of 40 numeric variables were calculated for 439 Twitter users. The variables were calculated using R code which took CSV files, containing information on 441,776 tweets, as input. Each row of these CSV files contained the following: tweet id, datetime the tweet was created, text of the tweet, geolocation (coordinates) of the tweet (if any), place the tweet was posted from (city, state, if any), source URL of the tweet, language of the tweet, name of the user (if any), screenname of the user, users followersCount, and users friendsCount.

4.2 Methods

I use Random Forest (RF) [Breiman 2001], k-Nearest Neighbor (k-NN) [Cover et al. 1967] and Linear Discriminant Analysis (LDA) [Fisher 1936] on the features mentioned in the last section to classify the Twitter accounts in my data set.

Random Forest functions by creating *n* number of decision trees, upon beginning training, and outputs either the mean prediction, in the case of regression, or the mode of the classes, in the case of classification, of each decision tree. The method combines bootstrap aggregating [Breiman 1996] and a random selection of features to create an ensemble of decision tress with controlled variance. Bootstrap aggregating, also known as “bagging”, is the technique of repeatedly choosing a random sample of the training set, with replacement, and fitting trees to the samples. Random forest differs only slightly from “bagging” in that it selects a random sample of features at each decision split during the learning process. An advantage of random forest is its ability to measure variable importance in a data set. This technique is implemented in the R package randomForest [Liaw 2002].

The k-Nearest Neighbor algorithm is a method for classification that makes no assumptions about the probability distributions of the variables being evaluated [Cover et al. 1967]. The input consists of the *k* closest data points, of the training set, within the feature space. Each data point is assigned to the class that is most common among its *k* nearest neighbors. Typically the optimal value of *k* is an integer near the natural logarithm of *n* (where *n* is the number of data points in the training set). In the case of this thesis (binary classification), *k* should also be an odd integer as to avoid potential tied votes. The *k*-NN algorithm is considered to be one of the simplest machine learning algorithms. Its simplicity, however, does not degrade its potential. The algorithm is guaranteed to approach the Bayes error rate (the lowest possible error rate for a class of classifier) [Cover et al. 1967] for some value of *k* (where *k* increases as a function on *n*). This algorithm is implemented in the R package knn [Venables et al. 2002].

Linear Discriminant Analysis is a generalization of Fisher's linear discriminant [Fisher 1936]. This method finds a linear combination of features that separates the classes of objects. Once found, the linear combination can be used as a linear classifier. It can also be used for dimensionality reduction prior to classification. A fundamental assumption of LDA is that independent variables are normally distributed. In fact, LDA only works if continuous quantities are calculated for the independent variables on each observation. In a two class problem LDA assumes that the probability density functions are of Gaussian distribution with the respective mean covariance parameters [Venables et al. 2002]. LDA then assumes that the covariances have full rank (all rows and columns are linearly independent). The decision criteria become a threshold for a constant *c* on the dot product of vectors *w* and *x*. In geometric terms, input vector *x* is deemed to be in class *y* if *x* is on a specific side of a hyperplane perpendicular to *w*. This algorithm is implemented in the R package lda [Venables et al. 2002].

4.3 Performance

The initial data set of 40 features across 439 accounts was concatenated with the data set containing the feature *isaBot* for each account and randomly split into two data sets while closely maintaining class proportions. This binary feature states whether or not the account is a bot based on the outcome of the majority vote system as described in section 3.2. These two data sets are used as the training and test data. The training and test sets being comprised of feature data for 219 and 220 accounts respectively.

The R implementation of randomForest offers a method of plotting variable importance by measure of *Mean Decrease Gini* (result of using *Gini index* as an impurity function). The *Gini index* is a measure of statistical dispersion and is considered to be one of the most commonly used measures of inequality [Gini 1912]. As an impurity function it measures the frequency at which a randomly chosen element of the set *A* would be mislabeled if it were labeled randomly by the distribution of labels in the subset of *A* [Breiman 1996]. After calculating the aforementioned variables this method was used to gauge the importance of each variable. The varImpPlot method takes an ensemble of trees as input and outputs the dot plot displayed in figure 4.1.

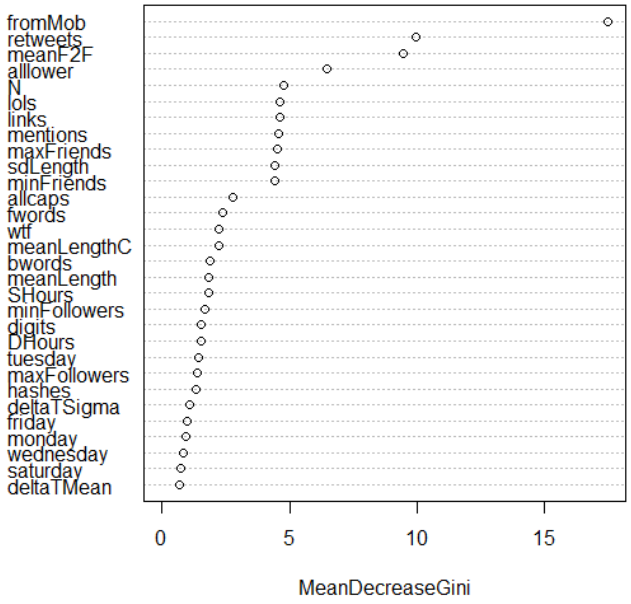

Fig. 4.1 The dot plot shows the *Mean Decrease Gini* of the top 30 of the 40 computed variables in the research. The closer a variable is to zero on this scale, the less of a statistical difference there is between members of either class for the given variable in the training set. This value can provide valuable information during the process of variable reduction.

The predictive importance of the feature *fromMob* at first appears very high. As a result I decided to create two separate models with each classification method: (i) with *fromMob* and (ii) without *fromMob.*

I first removed *fromMob* from the training and test data sets and derived a random forest from the training set using randomForest [Liaw 2002]. I then attempted to classify my test data set using the derived model. What I found was that with *fromMob* removed from the set of features, the model correctly predicted the account's class 209 out of 220 times. I then repeated this process, but left *fromMob* in both the training and test sets. The resulting model correctly predicted the account's class 209 out of 220 times again.

At first the results of the two models were puzzling considering the importance measure randomForest had given *fromMob.* The answer to how this result could happen stems from exactly how *mean decrease gini* measures a features importance. The importance of *fromMob* was determined to be high, not from its impact on final classification results, but instead on its ability to decrease impurity between the two classes at a randomly chosen node in a decision tree. While removing *fromMob* had no effect on the final classification for randomForest, k-Nearest Neighbor and Linear Discriminant Analysis proved more sensitive.

I applied a similar process to the k-Nearest Neighbor (k-NN) method to create two prediction models: (i) with *fromMob* and (ii) without *fromMob.* The method, as implemented in the R package knn [Venables et al. 2002], also has a method for cross validation. I include a third k-NN prediction model using this cross validation technique to test how well the results will generalize to an independent data set.

Before obtaining optimal results using k-NN two important elements had to be addressed: (i) the scale of feature values and (ii) the value of *k*. I first normalized all 40 features into the range zero to one:

$$f(x) = \frac{x[i] - \min(x)}{\max(x) - \min(x)}$$

where *x* is a column vector representing a feature. This is an important step to take when using k-NN, otherwise the feature with the largest scale will dominate the measure. I then had to find the optimal value of *k.* I did this in R by iterating over the knn method on my training set and incrementing the value of *k* by one on each iteration. The values of *k* were paired with the percentage of correctly predicted account classes to observe the optimal value of *k.* As figure 4.2 demonstrates, the optimal value of *k,* in this case, is five.

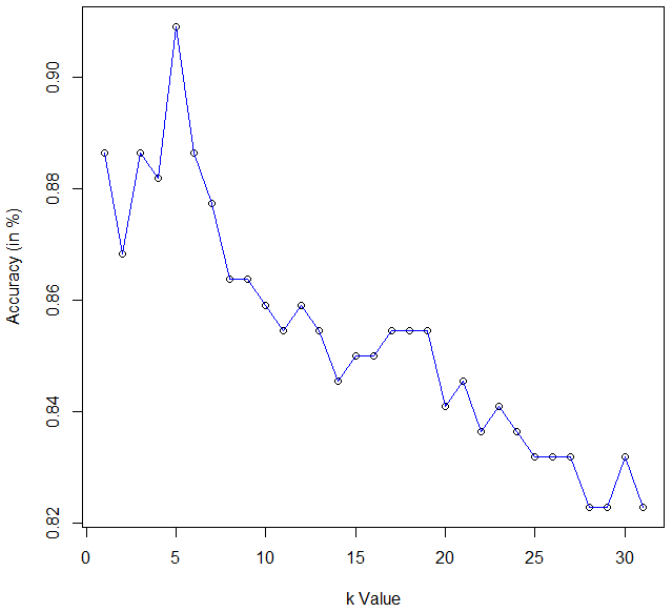

Fig. 4.2 The line plot represents the change in percent accuracy of the k-NN classification model as the value of *k* used in the algorithm changes. The *k* values assessed range from 1 to 31 as represented on the *X*-axis. The *Y*-axis represents the percent accuracy the prediction model exhibited for the given *k* value. This figure was derived from the training set excluding the feature *fromMob.*

The first k-NN model, which excluded *fromMob* from both the training and test sets, correctly predicted the account's class 200 out of 220 times. The second model, which included *fromMob,* correctly predicted the account's class 207 out of 220 times. I then used the knn cross validation method which resulted in a generalized accuracy rating of 0.916 or the equivalent of just over 402 correctly predicted account classes out of 439. The data used as input for cross validation included the feature *fromMob.*

The third and final method used in this research was Linear Discriminant Analysis (LDA). Out of the three methods mentioned LDA was most affected by the feature *fromMob.* In figure 4.3 the difference in linear separability with and without the feature is demonstrated. The result is a wider dispersion of linear discriminant coefficient values causing a greater number of misclassified data points.

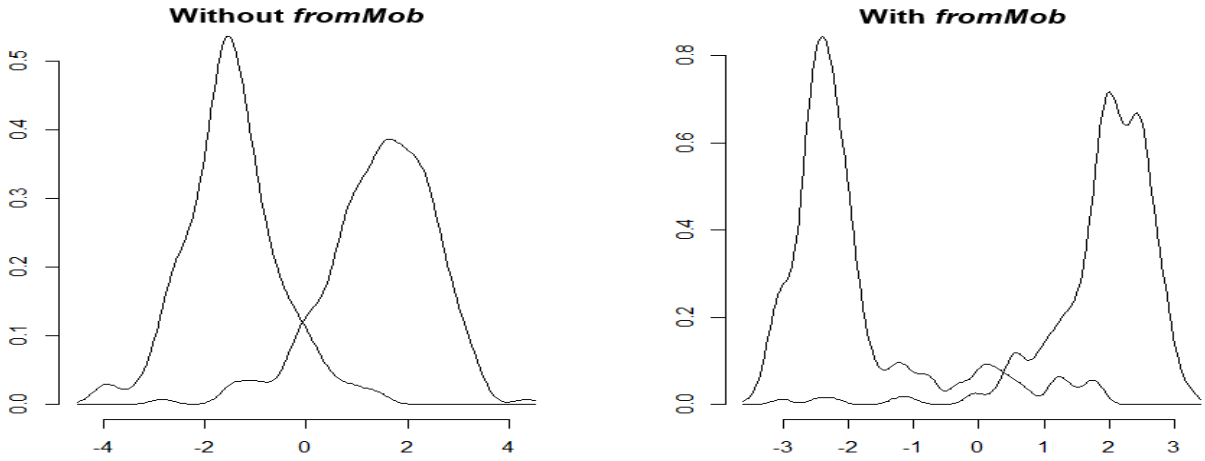

Fig 4.3 The two density plots show the linear separability of the dataset with and without the feature *fromMob.*

With *fromMob* removed from the set of features, the model correctly predicted the account's class only 194 out of 220 times. With *fromMob* included in both the

training and test sets the resulting model correctly predicted the account's class 203 out of 220 times. The results of LDA, and the fact that it performed the worst of the three methods, is primarily due to one important factor: the strict separator between the two classes. This does not allow for a "fuzzy" interpretation by the classifier of data points that fall near this line. As time goes on and more Twitter bots are created which more closely exhibit human features, this issue could very well be exacerbated for a LDA classifier.

5. Conclusions

Twitter bots can be classified with a high degree of accuracy. In this work, up to a 95 percent accuracy rate using randomForest was achieved. The ability to make this statement is of growing importance in today's age. Many business owners use social media outlets such as Twitter to advertise. As these businesses become aware of the bot population disguised among their potential viewers, they want to know that they're reaching their target audience, humans. What they need is analytic data on their audience.

Twitter has responded to this need to an extent with a tool written in R that they uploaded to GitHub called *AnomalyDetection*[3]*.* The tool is primarily focused on human (or spam) behavior analysis. Twitter's hope in releasing this tool open source on GitHub is that other software developers will help evolve the tool to aid in the global fight against hacking and spammers.

In the tool's current state it does very little in the way of preventative measure. The research I have done this semester could certainly be contributed to this tool as a means of tracking the estimated number of bots following an account on Twitter. A feature such as this would allow account holders to detect if they are being targeted by a network of bots and take preventative measure. Moreover, this feature would only be as effective as its accuracy rate. As a result, it would likely only be accepted as a useful feature with a high degree of accuracy. I believe that the 95 percent accuracy achieved in this research would meet this accuracy expectation and prove a valuable source of analytic data for Twitter account holders.

[3] "AnomalyDetection", Accessed April 27, 2015. https://github.com/twitter/AnomalyDetection

References

Boshmaf, Yazan, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. "Design and Analysis of a Social Botnet." *Computer Networks* 57.2 (2013): 556-78. Web.

Breiman, Leo. "Bagging Predictors." *Machine Learning* 24.2 (1996): 123-40. Web.

Breiman, Leo. "Random Forests." *Machine* 45.1 (2001): 5-32. Web.

Briscoe, Erica, Scott Appling, and Heather Hayes. "Cues to Deception in Social Media Communications." *47th Hawaii International Conference on System Sciences (HICSS)* (2014): 1435-443. Web.

Cheng, Alex. "Six Degrees of Separation, Twitter Style." *Twitter Friendship Data*. Sysomos, Apr. 2010. Web. 27 Apr. 2015.

Chu, Zi, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. "Who Is Tweeting on Twitter: Human, Bot, or Cyborg?" *26th Annual Computer Security Applications Conference. ACASC* (2010): 21-30. Web.

Cover, T., and P. Hart. "Nearest Neighbor Pattern Classification." *IEEE Transactions on Information Theory* 13.1 (1967): 21-27. Web.

Ferrara, Emilio, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. "The Rise of Social Bots." *[1407.5225] The Rise of Social Bots*. N.p., 2015. Web. 27 Apr. 2015.

Fisher, R. A. "The Use Of Multiple Measurements In Taxonomic Problems." *Annals of Human Genetics* 7.2 (1936): 179-88. Web.

Gianvecchio, Steven, Mengjun Xie, Zhenyu Wu, and Haining Wang. "Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification." *IEEE/ACM Transactions on Networking* 19.5 (2011): 1557-571. Web.

Gini, Corrado. *Variabilità E Mutabilità" 'Variability and Mutability'* N.p.: n.p., 1912. Print. C. Cuppini, Bologna, 156 pages. Reprinted in Memorie di metodologica statistica (Ed. Pizetti E, Salvemini, T). Rome: Libreria Eredi Virgilio Veschi (1955).

Hwang, Tim, and Ian Pearce. "Socialbots: Voices from the Fronts." *Interactions* 19.2 (2012): 38-45. Web.

Karinthy, Frigyes. *Chain-Links*. N.p.: n.p., 1929. Print. Translated from Hungarian and annotated by Adam Makkai and Enikö Jankó.

Liaw, Andy, and Matthew Wiener. "Classification and Regression by RandomForest." *R News* 2.3 (2002): 18-22. Web.

Pal, Aditya. "User Classification in Online Communities." Diss. U of Minnesota, 2012. Print.

Sebastiani, Fabrizio. "Machine Learning in Automated Text Categorization." *ACM Computing Surveys* 34.1 (2002): 1-47. Web.

Tavares, Gabriela, and Aldo Faisal. "Scaling-Laws of Human Broadcast Communication Enable Distinction between Human, Corporate and Robot Twitter Users." Ed. Yamir Moreno. *PLoS ONE* 8.7 (2013): E65774. Web.

Torralba, Antonio, and Alexei Efros. "Unbiased Look at Dataset Bias." *CVPR* (2011): 1521-528. Web.

Venables, W. N., and Brian D. Ripley. *Modern Applied Statistics with S*. New York: Springer, 2002. Print.

Wagner, Claudia, Silvia Mitter, Christian Korner, and Markus Strohmaier. "When Social Bots Attack: Modeling Susceptibility of Users in Online Social Networks." *21th International Conference on World Wide Web.* (2012): 41-48. Web.

Wald, Randall, Taghi M. Khoshgoftaar, Amri Napolitano, and Chris Sumner. "Predicting Susceptibility to Social Bots on Twitter." *14th International Conference on Information Reuse and Integration (IRI).* (2013): 6-13. Web.

Wang, Gang, Manish Mohanlal, Christo Wilson, Xiao Wang, Miriam Metzger, Haitao Zheng, and Ben Y. Zhao. "Social Turing Tests: Crowdsourcing Sybil Detection." *NDSS. The Internet Society.* (2013): n. pag. Web.

Yardi, Sarita, Daniel Romero, Grant Schoenebeck, and Danah Boyd. "Detecting Spam in a Twitter Network." *First Monday* 15.1 (2010): n. pag. Web.