Using Twitter to Predict Voting Behavior

Mike Chrzanowski
mc2711@stanford.edu

Daniel Levick
dlevick@stanford.edu

December 14, 2012

Background:

An increasing amount of research has emerged in the past few years using social media to either predict the outcomes of elections or, in the case of the United States, to classify users as Democrat or Republican. While predictions based on social media are not representative of the voting population, they have been shown to compete with surveys in accuracy and can be done in real time to provide instantaneous feedback to political events (Tweetminster, 2010).

Twitter provides an excellent platform for solving political classification problems. The real-time availability of large amounts of data as well as built-in content sorting mechanisms (e.g., hashtags, retweets, followers) are major advantages over using other forms of social media. If these advantages can be leveraged to predict how Twitter users will vote, this information could be used to predict an election outcome or predict the likely impact of real-time events on voting patterns.

Problem Statement:

Our goal is to train an algorithm that could have predicted an arbitrary Twitter user's vote in the 2012 US Presidential Election, using methods that could be applied to future elections. Recent studies have shown the ability to accurately classify users as Democrat or Republican (e.g. Pennacchiotti, 2011; Boutet, 2011). With less consistent accuracy, other studies have attempted to predict the outcome of major elections based on aggregate data such as the volume of candidate mentions (Tumasjan, 2011). No studies we are aware of have attempted to directly predict an individual user's vote in an upcoming election outside of the framework of political parties. By eliminating this framework, we hope to better predict the voting patterns of users, especially those that do not identify strongly with either party.

Data Collection:

We collected two groups of tweet data. The first group is composed of tweets from the candidates and people that have publicly endorsed them. These are the sources that most likely originate much of the retweeted content and hashtags that might be used by supporters. The second group exploits the revealing of voting preferences by average Twitter users on Election Day. For example, some users explicitly stated whom they voted for after leaving the voting booth, while others retweeted tweets of the form “retweet this if you voted for candidate X.” This data set is closer to representative of the average Twitter user and provides an observable ground-truth by which to train and test our algorithm. This is still, admittedly, not a representative sample of all Twitter users.

The use of reveal tweets has three advantages over data collection techniques we have encountered in prior research. First, it is fast and minimally subjective; we did not have to manually classify users based on their past tweets. Second, it does not rely on user self-classification through websites like WeFollow or Twellow and therefore may capture users that either do not use these services or do not have strong allegiance to a particular party. Third, it is a direct measure of the variable we are trying to classify: voting behavior. This is especially relevant in the U.S. where only 60% of the population identifies with a political party (Jones, 2012) and typically less than 50% of the population actually votes in presidential elections.

Our data corpus contains over 7.5 million tweets created before Election Day. 50% came from Obama-voting authors and the other half came from Romney-voting authors. The tweets were generated from 4,835 accounts, of which 55% voted for Obama and 45% for Romney.

Reference Model Methods and Results:

Our primary algorithm uses all available data to train and cross-validate an SVM. We first harvest tweets via a Python implementation of the Twitter API and push them into a MySQL database. Each tweet, paired with its author's classification, is a training example. For preprocessing, we use NLTK's Lancaster algorithm for stemming and we substitute all numbers with "numbr". This data is then stored. When beginning a model run, this data is pulled and randomly divided into training and testing sets according to a cross-validation ratio. We then transform the training and test sets into SciPy sparse matrices by constructing frequency vectors of tokens that appear at least twice in the training set corpus using scikit-learn's CountVectorizer class. We then use the training set matrix to create inverse document frequency weights that are used to do TF-IDF transformation on both matrices using the library's TfidfTransformer class. We train an instance of the library's LinearSVC SVM on the training matrix and test it on the test matrix to classify each tweet as +1 (user voted for Romney) or -1 (user voted for Obama). At the end, users are classified by the sign of the sum of their tweets' predicted classifications.

With an 80/20 split of training versus test samples, the user classification model achieves 91% testing accuracy and 94% training accuracy. This is higher than accuracies achieved using tweet text to predict political alignment in recent literature (Conover, 2011). Our model's runtime is just over ten minutes when the preprocessing and data gathering steps are ignored.

Algorithm Exploration:

Alarmed by our high scores, one of our priorities was determining whether any assumptions we made in designing the reference model yielded inflated accuracy ratings. We point out that our disregard of which account a certain tweet is authored by when partitioning our data corpus allows for tweets from the same author to enter into both the training and test sets. To test the effects of this, we ran the reference model but placed all of each account's tweets in either the training or testing set. Testing accuracy was 82% with a 50% cross-validation ratio. We found that we did not have enough data, in terms of the total number of accounts available, to produce consistent results with other cross-validation ratios and were therefore unable to explore this property further. That is, accuracy rates had a very high variance depending on whether the tweets of certain accounts were placed in the training or testing sets, and the effect was exacerbated by uneven cross-validation ratios. However, the experiment shows that while the assumption has a noticeable effect on performance, accuracy rates remain very high.

We also began exploring the effects of certain characteristics of the training and test data upon our accuracy rates. We ran the reference model while precisely controlling the exact number of tweets, their age, the number of users from which the tweets originated, and the maximum size of the token corpus used to create our sparse matrices.

We were primarily interested in seeing how well we could predict the classifications of new tweets based on older tweets. Thus, most of our experiments followed the format of cleaving the tweet data into two sets based on whether they were created before or after a certain point in time. Tweets that were authored before that temporal pivot constituted the training set, and those created afterwards composed the testing set.

As it turns out, for any pivot point we used between 1/1/2011 and 11/1/2012, increasing the number of tweets within the training set while holding constant the number of tokens (i.e., restricting CountVectorizer objects to only produce matrices with a certain number of columns) and originating authors is almost never useful: the prime drivers in increasing accuracy come from increasing the number of accounts or increasing the number of tokens. In addition, we found that if all the other described variables are held constant, tweets generated later in time have more predictive power of future tweets than tweets authored earlier. Though we did not specifically test for statistical significance, and thus are not confident in the magnitudes of our observed accuracy rate increases, our outlined results held despite repeated runs of the model with the same parameters and using millions of randomly selected tweets.

To analyze whether increasing the number of originating accounts or tokens is more helpful, we decided to create a token corpus independent of our data by mapping every unstemmed token to a token within a predefined dictionary (i.e., the SCOWL word list) as determined by PyEnchant's spell checker functionality, which traverses a trie to uncover tokens with the longest common substrings. We then stemmed and subbed this text just as we would the normal, raw tweet text and ran the reference model on it using temporal pivots while controlling for the same variables as described above. Using this new corpus, we again found that increasing the number of tweets in the training set while holding the number of originating users and tokens constant resulted in no increase in accuracy, and we again found that accuracy rates increase based on how close to the election tweets were created. We attributed this to the fact that the use of dictionary words still changes with time based on how new raw tokens are being mapped to the dictionary token list. Unfortunately, we saw the same increases in accuracy rates when we varied the number of accounts or the number of tokens as we did when dealing with the non-spell-checked corpus. We thus come away with the qualitative conclusion that working with as diverse and fresh a data corpus as possible is the prime driver of prediction accuracy of future tweets.

Other Algorithm Exploration (Failed Experiments):

One avenue for exploration that we were interested in pursuing was better accuracy rates using only hashtag data. When we used the reference model on hashtag data alone, which consisted of 1.7m tweets composed of 293,000 unique hashtags, we achieved a tweet-level classification testing accuracy of 72% and training accuracy of 74%. We then tried to perform unsupervised clustering of the hashtag data using K-Means. Our pipeline remained unchanged from that used in our reference model except that instead of classification, we performed clustering based on a variable number of centroids and, after the algorithm converged, used majority voting of the assigned tweets within each cluster to determine a label for that whole cluster. We then calculated the prediction accuracy on the testing set by assigning each testing tweet to the closest group and determined whether that cluster's label matched that of the test sample's label. As it turns out, even after grouping the data into 12,000 clusters (and being able to go no further because of memory limitations), we could only achieve tweet-level testing accuracy rates of 66%. This was not a competitive tweet-level classification accuracy given the algorithm's extremely long runtime, and since account-level classification accuracy seems very closely tied to the success of tweet-level classification, we didn't spend more time on this.

Another attempt to improve on the accuracy rates relied on adding features that signified whether a certain tweet's author was following the Twitter accounts of the presidential or vice presidential candidates. As the hashtag corpus is far smaller than the stemmed token corpus used in the reference model, adding 4 new features would show a more noticeable accuracy change

and so would allow us to determine whether to spend the engineering effort to also add them to the reference model. As it turned out, after running the experiment many times, tweet-level accuracy rates were completely unchanged, account-level testing accuracy went up by .4% to 76.5% and account-level training accuracy fell by 0.4% to 82.1%. We felt that while these results are impressive given that we only added 4 additional columns to each training and test vector, it would not be possible to add these features to every vector in the reference model and maintain the same runtime of the reference model (i.e., there are a lot of expensive intermediate steps required to create sparse matrices to accommodate these new columns) without first drastically shrinking the size of the data we were working with. As we were more concerned with keeping the model runtime low to be able to run many experiments quickly, we decided to only leave this feature in place when processing hashtag data.

Attempting to plot a subset of the tweet data as well as the SVM's decision boundary by applying PCA resulted in an enormous loss of information; the end result was garbage data.

Temporal Performance Experiment:

One potential application of our algorithm is vote prediction in advance of an election. To model this situation, we trained and tested the algorithm only on tweets created before a target date. In the real world, data labels would be obtained via surveys of voting intention. Interestingly, we note that the breakdown of tweets and accounts by candidate supported and cut-off dates used didn't change very much at all from the corpus-level breakdowns. The account-level testing accuracy results are presented in Figure 1.

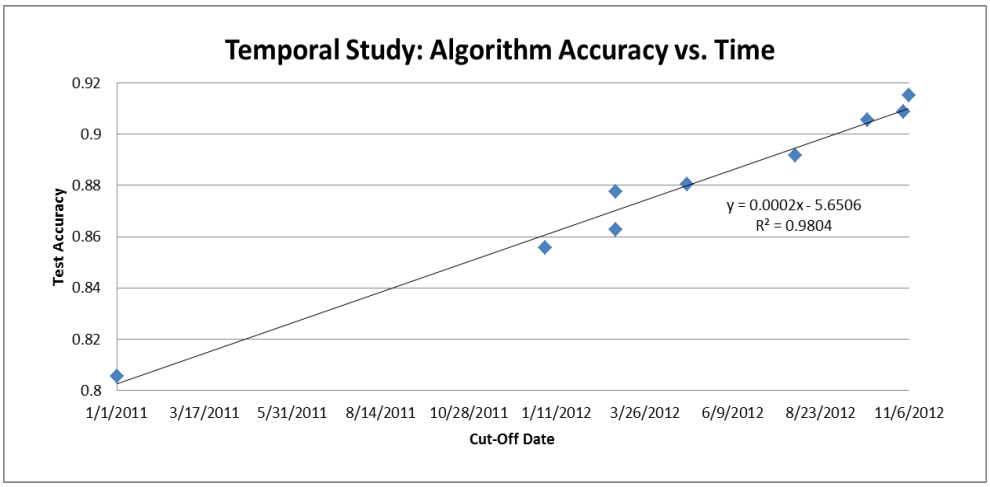

Figure 1: Temporal Performance Experiment Results

The algorithm achieves greater than 80% testing accuracy when using only data from before January 1, 2011. This suggests that our model could provide accurate voting predictions up to two years before an election.

Demographic Analysis Application:

Classification of over 4000 Twitter accounts allowed us to perform interesting comparisons of the two voter groups and which may be a starting point for future research in the area. This represents another potential application of our algorithm: post-election demographic analysis. For instance, we found that Obama voters use curse words between 1.5-2x more often than Romney voters. Tweets from Romney voters were 50% more likely to mention religious terms.

Romney voters also used hashtags far more often and for more political speech than Obama voters. Figure 2 shows a list of the top ten most frequent hashtags for each voting class.

Figure 2: Top Ten Hashtags for Each Voting Class

	Obama Voters (instances)	Romney Voters (instances)
1	teamFollowBack (5170)	tcot (41793)
2	FF (5142)	RomneyRyan2012 (13904)
3	Sandy (4469)	Benghazi (13150)
4	oomf (4443)	TeaParty (7763)
5	p2 (3866)	Obama (7687)
6	Obama2012 (3499)	FF (6076)
7	Capricorn (3423)	debates (5784)
8	Np (2763)	Debate (5611)
9	RT (2558)	Sandy (5531)
10	Sagittarius (2337)	Gop (5258)

Conclusions:

Through the construction and optimization of our algorithms, we accomplished our goal of accurately predicting Twitter user voting behavior. We feel that analysis of our algorithms provides useful recommendations for future social media machine learning design efforts. Classifying Twitter users by voting patterns also resulted in several interesting demographic findings, and a study of performance on older tweets showed some promise for applications in pre-election vote prediction. Future areas of research include closer inspection of the demographic information of users and their tweets as well as establishing quantifiable, statistically significant measurements of how useful adding new accounts or tokens would be in raising accuracy rates.

Acknowledgements:

We'd like to thank the CS 229 teaching staff for helping us formulate a research topic to pursue and for suggesting interesting questions to spend time exploring.

References:

Atkinson, K. (2011, January 6). Kevin's Word List Page. Retrieved December 13, 2012, from Kevin's Word List Page website: http://wordlist.sourceforge.net/

Bird, Steven, Edward Loper and Ewan Klein (2009). Natural Language Processing with Python. O'Reilly Media Inc.

Boutet, A., & Yoneki, E. (2011). Member Classification and Party Characteristics in Twitter during UK Election. Collocated with OPODIS 2011/Toulouse, France Editors: Lélia Blin and Yann Busnel, 18.

Conover, M. D., Gonçalves, B., Ratkiewicz, J., Flammini, A., & Menczer, F. (2011, October). Predicting the political alignment of twitter users. In Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom) (pp. 192-199). IEEE.

Jones, J. (2012). Record-High 40% of Americans Identify as Independents in '11. Retreived November 14, 2012, from http://www.gallup.com/poll/151943/record-high-americans-identify-independents.aspx.

Pennacchiotti, M., & Popescu, A. M. (2011). Democrats, republicans and starbucks afficionados: user classification in twitter. In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 430-438).

Tumasjan, A.; Sprenger; T., Sandner; P., Welpe, I. "Predicting elections with Twitter: What 140 characters reveal about political sentiment," in Proc. of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM), 2010.

Tweetminster (2010). Is word-of-mouth correlated to General Election results? The results are in. Retrieved November 14, 2012, from http://www.scribd.com/doc/31208748/Tweetminster-Predicts-Findings.