

Predicting Donald Trump’s Tweet Frequency with Monte Carlo Simulation

Adam Blakney

Abstract

In this paper we apply several statistical learning, modeling, and forecasting techniques to the problem of predicting the number of tweets Donald Trump will post in a certain amount of time. We tailor our approach to the constraints of PredictIt’s Tweet markets.

1 Introduction

Prediction markets are markets that allow users to bet on the occurrence of events in the future [3]. In particular, prediction markets may offer the opportunity to speculate about the outcomes of sports games, election outcomes, etc. PredictIt is one such prediction market that focuses on political events [4]. In particular, PredictIt offers markets that allow users to speculate on the number of tweets that certain political figures will post in a pre-determined amount of time. Many different factors may be considered in the task of predicting the Twitter behavior of a political figure; past Twitter behavior, current political climate, and other current events may be relevant. In this paper, we focus on the use of past Twitter behavior to predict future Twitter behavior. In particular, using the past Twitter behavior of the user @realDonaldTrump (see [1]) we aim to predict how many tweets @realDonaldTrump will post in time spans ranging from twelve hours to one week.

2 PredictIt Tweet Markets

In this section we give an brief overview of PredictIt’s twitter markets, as well as a basic review of prediction markets. [3] and [2] should be consulted for further information on prediction markets and trading on PredictIt.

2.1 Markets, Brackets and Shares

A market is composed of brackets, which specify possible market outcomes. Users interact with the market by buying “Yes” or “No” shares in each bracket, each of which is priced between \$.01 and \$.99. When the market resolves,

exactly one bracket will resolve as the winning bracket while all others resolve as losing brackets. At market closure, a winning bracket's "Yes" shares will be worth \$1; similarly a losing bracket's "No" shares will be worth \$1. Thus, generally speaking, a user is incentivized to buy "Yes" shares in brackets they believe will win and "No" shares in brackets they believe will lose.

2.2 Maximizing Profit: Probabilistic Interpretation

While the specifics of trading shares are rather involved (see [2]), here we discuss some basic facts about prediction markets, and explore how one might profit off of miscalibrated markets. Specifically, in this section, we formalize the notion of expected profit and provide a framework for interpreting prediction markets probabilistically. To formalize these concepts we first introduce the following definitions.

Definition 1. *A market M is a collection of n brackets, denoted b_1, \dots, b_n , one of which will resolve as the winning bracket, and the rest of which will resolve as losing brackets. The index of the winning bracket is unknown until the market closure.*

Definition 2. *Associated with each bracket b_i is a range of tweets R_i such that, if the tweet count x falls in the range R_i at the market closure time, bracket b_i resolves as the winning bracket. Note that the sets R_1, \dots, R_n must be disjoint, as exactly one bracket will resolve as the winning bracket at market closure. Further, denote the "Yes" price of bracket b_i by s_i , and denote the probability of it resolving as the winning bracket p_i . (Consequently, the probability of it resolving as a losing bracket is $1 - p_i$).*

Clearly, the probabilities p_1, \dots, p_n are unknown: otherwise, assuming a rational market, the share prices s_1, \dots, s_n would converge to them and the market could not be exploited. We now consider the impact of buying "Yes" or "No" shares in a market, and note the following.

Remark. *The expected profits from buying one "Yes" or "No" shares in bracket i are*

$$\mathbb{E}(\text{profit}) = \begin{cases} .9p_i - s_i, & \text{when buying a "Yes" share} \\ .9(1 - p_i) - (1 - s_i), & \text{when buying a "No" share} \end{cases},$$

where the factor of .9 arises because PredictIt takes 10% of profits. Thus we note the following:

Remark. *It is profitable to buy a "Yes" or "No" share for bracket b_i when,*

$$.9p_i > s_i,$$

or,

$$.9(1 - p_i) > 1 - s_i,$$

respectively.

Thus, our goal is to estimate the probabilities p_1, \dots, p_n reliably so that we may determine when a buy is profitable.

2.3 @realDonaldTrump Market Example

For each selected Twitter user, a new market is opened each week, and the market brackets—which users may place bets on—are associated with the number of tweets to be posted by the account over that week. Fig.1 shows a snapshot of a @realDonaldTrump Twitter market. The left-hand side gives the bracket outcomes, which specify the number of tweets to be posted by the user. Also visible in the left-hand side is the users' stake in each bracket. Numbers boxed in green signify stake in a "yes" contract while those in red signify stake in a "no" contract. For example, we see that in Fig. 1, under the bracket title "220 - 229" is the number 100 boxed in green, indicating that the user has stake in 100 "yes" contracts for the aforementioned bracket. The remaining columns give market prices. Note that we may interpret these market prices probabilistically. For example, consider










Contract	Latest Yes Price	Best Offer	Best Offer
 189 or fewer	1¢ NC	1¢	<input type="button" value="Buy Yes"/> <input type="button" value="Buy No"/> N/A
 190 - 199	1¢ 2¢↓	1¢	<input type="button" value="Buy Yes"/> <input type="button" value="Buy No"/> N/A
 200 - 209	1¢ 7¢↓	1¢	<input type="button" value="Buy Yes"/> <input type="button" value="Buy No"/> N/A
 210 - 219	3¢ 8¢↓	3¢	<input type="button" value="Buy Yes"/> <input type="button" value="Buy No"/> 98¢
 220 - 229 100 avg. paid 7¢ 1¢↓	6¢ 12¢↓	6¢	<input type="button" value="Buy Yes"/> <input type="button" value="Sell Yes"/> 5¢
 230 - 239 50 avg. paid 14¢ 1¢↓	13¢ 3¢↓	13¢	<input type="button" value="Buy Yes"/> <input type="button" value="Sell Yes"/> 12¢
 240 - 249	17¢ 3¢↑	18¢	<input type="button" value="Buy Yes"/> <input type="button" value="Buy No"/> 83¢
 250 - 259	19¢ 4¢↑	20¢	<input type="button" value="Buy Yes"/> <input type="button" value="Buy No"/> 81¢
 260 or more 15 avg. paid 61¢ NC	39¢ 15¢↑	61¢	<input type="button" value="Buy No"/> <input type="button" value="Sell No"/> 60¢

Figure 1: A snapshot of the @realDonaldTrump Twitter market with a users' stakes in various brackets.

3 Problem Formulation

Recall that our goal is to estimate the probabilities p_1, \dots, p_n of each bracket resolving as the winning bracket. Because each bracket gives a range of tweets to be posted by the end of the week, this entails predicting the probability distribution of the number of tweets posted through the end of one week. To do so, we model the history of tweet counts as outcomes of a sequence random variables, and use the observed values to estimate the next random variable in the sequence.

Consider a continuous time interval of length L partitioned into N sub-intervals τ_1, \dots, τ_N , each of length l (in our case, we use $L = \text{one week}$ and $l = 12 \text{ hours}$). Then we model the number of tweets during each of these intervals as a random variable $X \in \mathbb{R}^N$, with,

$$\begin{pmatrix} X_1 \\ \vdots \\ X_N \end{pmatrix} = \begin{pmatrix} \text{number of tweets during } \tau_1 \\ \vdots \\ \text{number of tweets during } \tau_N \end{pmatrix}.$$

And so $\sum_{i=1}^N X_i$ gives the tweet count for the week associated with X .

Therefore, given a collection of random variables $X^{(1)}, \dots, X^{(W)}$ representing tweet counts through W weeks, our goal is to estimate the distribution of $\sum_{i=1}^n X_i^{(W+1)}$ which gives the number of tweets during the next week. Then we obtain estimates for the bracket probabilities $\hat{p}_1, \dots, \hat{p}_n$ with $p_i = \mathbb{P}_{\hat{F}}(X_i^{(W+1)} \in R_i)$, where \hat{F} is the estimated distribution of $X^{(W+1)}$.

4 Monte Carlo Simulation

4.1 Model Description

In this section we attempt to estimate the probabilities p_1, \dots, p_n via Monte Carlo simulation. We assume that the account's behavior is similar during similar times of the week. Thus, given W weeks of data $X^{(1)}, \dots, X^{(W)}$, where $X^{(i)} \in \mathbb{R}^N$, we assume the $X^{(i)}$'s are independently and identically distributed according to some distribution F . We then consider two strategies for estimating F :

1. Given $X^{(1)}, \dots, X^{(W)}$, estimate the distribution of each component of $X^{(W+1)}$ independently with $\hat{F}_{X_i^{(W+1)}}(x) = \frac{1}{W} \sum_{j=1}^W \mathbb{1}_{\{X_i^{(j)} \leq x\}}$. That is, we take our estimate of the distribution of the i -th component of X to be the empirical distribution of $X_i^{(1)}, \dots, X_i^{(W)}$.
2. Given $X^{(1)}, \dots, X^{(W)}$, estimate the distribution of each component of $X^{(W+1)}$ independently with $\hat{F}_{X_i^{(W+1)}}(x) = w_i \sum_{j=1}^W \mathbb{1}_{\{X_i^{(j)} \leq x\}}$, where $w_i = ce^{-\alpha(N-i)}$ for some constant c . That is, we take our estimate of the distribution of the i -th component of X to be a weighted empirical distribution of

$X_i^{(1)}, \dots, X_i^{(W)}$, where the weights decrease exponentially as we consider older data.

Note that both strategies estimate the distribution of each component independently. The first strategy is a standard bootstrap approach; all past observed values are weighted equally to form the prediction of each component $\hat{X}_i^{(W+1)}$. The second strategy, however, weights data according to an exponentially decreasing function. This is motivated by the idea that more current data should be weighted more heavily, so as to account for long-term trends in the data.

In both strategies we sample many $\tilde{X} \sim \hat{F}$ independently from the empirical distribution as stated. Let $\tilde{X}^{(1)}, \dots, \tilde{X}^{(M)}$ denote M random variables sampled independently from \hat{F} . Then we estimate the bracket probabilities with,

$$\hat{p}_i = \frac{1}{M} \sum_{i=1}^M \mathbb{1}_{\{\tilde{X}^{(i)} \in R_i\}}.$$

4.2 Using the Model Throughout the Week

Given W weeks of data $X^{(1)}, \dots, X^{(W)}$, we have given two strategies to estimate the distribution of the number of tweets during the next week, $\sum_{i=1}^N \hat{X}_i$. However, we might consider updating our estimate throughout the week. As we collect tweet data during the week, the first components of our vector $X^{(W+1)}$ begin to be known. Once we have observed through τ_s during the given week, we may take $\hat{X}_i^{(W+1)}$ to be the number of tweets during τ_i for $i \leq s$, and estimate the rest as described above.

4.3 Model Evaluation

When evaluating our model, we are interested in how accurate we estimate the probabilities $\hat{p}_1, \dots, \hat{p}_n$. Further, we wish to evaluate our model throughout the week, namely after each interval τ_i . To measure model accuracy, we use cross-entropy loss. Given W weeks of data $X^{(1)}, \dots, X^{(W)}$ and an additional s data points during the week $X_1^{(W+1)}, \dots, X_s^{(W+1)}$ representing tweet counts during τ_1 through τ_s , we may measure the

$$H()$$

References

- [1] Donald j. trump. abc <https://twitter.com/realDonaldTrump>.
- [2] Donald j. trump. <https://www.predictit.org/support/how-to-trade-on-predictit>.
- [3] Prediction market. a <https://www.investopedia.com/terms/p/prediction-market.asp>.

[4] Predictit. av predictit.org.