Predicting Information Spreading in Twitter

Tauhid R. Zaman*
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA
zlisto@mit.edu

Ralf Herbrich
Microsoft Research Ltd.
Cambridge, UK
rherb@microsoft.com

Jurgen van Gael
Microsoft Research Ltd.
Cambridge, UK
jvangael@microsoft.com

David Stern
Microsoft Research Ltd.
Cambridge, UK
dstern@microsoft.com

Abstract

We present a new methodology for predicting the spread of information in a social network. We focus on the Twitter network, where information is in the form of 140 character messages called *tweets*, and information is spread by users forwarding tweets, a practice known as *retweeting*. Using data of who and what was retweeted, we train a probabilistic collaborative filter model to predict future retweets. We find that the most important features for prediction are the identity of the source of the tweet and retweeter. Our methodology is quite flexible and be used as a basis for other prediction models in social networks.

1 Introduction

Determining who is influential in a network or how many people a piece of information reaches is very important in many different fields. For example, online advertisers could use this information for efficient targeted marketing campaigns. Media companies could learn how to effectively generate buzz for new films, shows, or musicians. Political action groups could learning who they should try to influence in order to spread their message as far as possible. With the enormous growth in social networking sites such as Twitter and Facebook, there is now an ample amount of data available for learning how information spreads at a micro-level. One then needs to understand the proper way to utilize this data in order to predict future information spreading patterns.

In this work, we will focus on predicting information spreading in Twitter. Twitter is a microblogging service that allows users to share information in the form of 140 character messages called *tweets*. In Twitter, a user has followers who will receive any tweet posted by the user, and a user can follow other users. The Twitter network is comprised of the follower/following relationships.

Information can spread in Twitter in the form of *retweets*, which are tweets that have been forwarded by a user to his or her followers. A retweet is identified by the string "RT @" followed by the name of the tweet source in the text of the retweet. Retweets allow one to track the flow of information in Twitter because they indicate situations where a user felt a tweet was important enough that he or she shared it with his or her followers. For this reason, to predict information spreading in Twitter, we wish to predict retweets.

*

The Twitter network structure, retweet network structure, and temporal properties of retweets were analyzed in [1]. The influence of users in Twitter across topics and time was analyzed in [2] using different measures including number of retweets and number of followers. In [3], the conversational aspects of retweeting were analyzed. While the Twitter network and retweets have been studied, there has been little work done on predicting retweets at a micro-level in Twitter. However, there has been other work done in using social networks for prediction. For example, box-office revenue for movies have been predicted using chatter from Twitter in [4]. This work focused on aggregate measurements such as the rate at which people tweet about a movie. The detailed network structure was not incorporated into the predictions.

In this work we present a methodology for predicting individual retweets in Twitter. We gather data from Twitter in order to train probabilistic collaborative filtering models to predict future retweets. These models learn retweet patterns using the tweet source (the tweeter), the user who is retweeting (the retweeter), and the tweet content. We find that the most important features for prediction are the tweeter and retweeter.

2 Retweet training data

In this section we will describe how we gathered retweet data from Twitter and used it to generate training data for probabilistic collaborative filtering models.

2.1 Probabilistic collaborative filtering models: Matchbox

We wish to develop models which can predict retweets. Specifically, we wish to develop a model where the input is the tweeter, a retweeter, and the content of the tweet. The output of the model will be a value p which is the probability of a retweet of the tweet by the retweeter. The predictive model we use is a probabilistic collaborative filtering prediction model called Matchbox [5] which was originally developed to predict the movie preferences of users based on meta-data about movies.

Matchbox uses three types of input to learn: user features, item features, and binary feedback. Matchbox models learn correlations between users and items in order to predict user preferences for items. Details on the Matchbox model can be found in [5].

For our application we will have three types of features. First, there are the tweeter's features (name, number of followers, etc.). Second, there are the retweeters features (name, number of followers, etc.). Third, there is the tweet content. There are different ways to divide these features into item and user features, and we will train different models to see which division works best. The binary feedback is 1 if the retweeter retweeted the tweet within a certain time window, and 0 otherwise. For our training data, we used a time window of one hour. This is sufficient because it was found in [1] that half of the retweets occur within an hour of the source tweet.

2.2 Retweet network and negative feedback

By collecting retweets, we can obtain the positive binary feedback required for training Matchbox models. However, we also need negative feedback for the models to be properly trained. For every tweet, the negative feedback would come from the followers who do not retweet the tweet within an hour. However, there may be followers who are never active on Twitter at all, and these would bias the training data. What we want are active users who have retweeted or have been retweeted in the past. Therefore, in order to obtain negative feedback, we need to compute the *retweet network*.

We crawled Twitter from June 20th, 2010 to July 29th, 2010, collecting every retweet in this period. We detected retweets by looking for the string "RT @" in the body of the tweet. There were 102 million retweets found this way. By selecting the unique (tweeter, retweeter) pairs from these retweets, we obtained a network of 50 million edges and 7.3 million distinct users.

2.3 Generating Training Data

The training data for the Matchbox models was generated in the following manner. We collected every tweet from a one hour time window and looked for any retweets of these tweets for up to one hour after the time of the tweet. These retweets were the positive binary feedback. We obtained the

Model	User Features	Item Features
Model 1	Retweeter name, numer of retweet-followers, number of retweet-following	Tweeter name, number of retweet-followers, number of retweet-following, tweet words
Model 2	Retweeter name, numer of retweet-followers, number of retweet-following	Tweeter name, number of retweet-followers, number of retweet-following
Model 3	Retweeter name, numer of retweet-followers, number of retweet-following	Tweeter name, number of retweet-followers, number of retweet-following, normalized tweet words

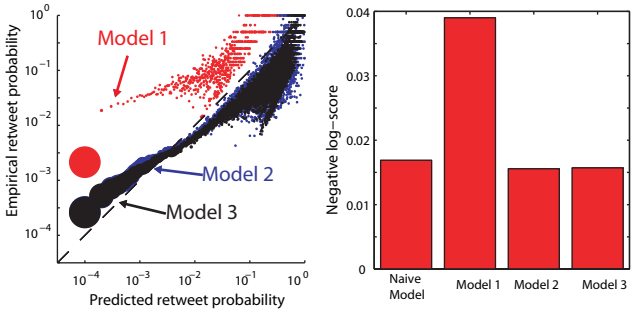

Figure 1: (left) Table indicating the user and item features of the retweet prediction models. (center) Bubble plot and (right) negative log-score of the models' performance.

negative feedback from all followers of the tweeter in the retweet network who did not retweet. This data contained over 99.8 % negative feedback because most tweets are not retweeted.

3 Model Performance

We trained three different Matchbox models with 1 hour of tweets. We then used the models to predict retweets for the subsequent hour. The models differ in their user and item features, and are listed in Figure 1. The features used for tweeters and retweeters were name, number of retweet-followers, and number of retweet-following.

To evaluate the performance of these models on the prediction dataset, we show calibration plots for the different models in Figure 1. Calibration plots are constructed as follows. The prediction values (p's) are grouped into bins of width 0.01%. Then, the empirical retweet probability for nodes within each bin is calculated. The calibration plot has the predicted p on the x-axis and the empirical retweet probability within the bin on the y-axis. The size of the points on the plot indicate the number of data samples in the bin. For models which are well calibrated, the points should lie along the line $y = x$, indicating that the model predicts the right empirical retweet probability for the data in the bins.

Another method of evaluating model performance is the negative log-score. This score is calculated for a set of N data samples with binary labels $\{y_1, y_2, ..., y_N\} \in \{0, 1\}$, and prediction values $\{p_1, p_2, ..., p_N\} \in [0, 1]$ obtained from the model. The negative log-score is defined as

$$\mathrm{NL}(\mathbf{y}, \mathbf{p}) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \,. \tag{1}$$

If the model has very strong predictive power, then $p_i = y_i$ and the negative log-score is zero. If the model is not perfect, then the negative log-score will increase. Therefore, a smaller negative log-score means better model performance. In Figure 1 we show the negative log-score for the models and for a naive model which predicts $p = 0.2\%$ for every (tweeter,retweeter). This is the empirical retweet probability over all the training data. This naive model has very little predictive power, but it is a good benchmark against which to compare the Matchbox models.

For each model, certain stop words (i.e. the, a, and, RT, etc.) are removed from the tweet text. Model 1 has the tweeter and the tweet words as item features. The retweeter is the user feature. Model 2 uses only the tweeter as the item feature and ignores the tweet entirely. As can be seen in the bubble plot and log-score, Model 2's performance is much better than Model 1 and its negative log-score is less than the naive model. Removing the tweet seems to improve performance. This may be due to the words of the tweet being unnormalized. This means that longer tweets have feature vectors with a greater norm than shorter tweets. We normalized the words of the tweet by the number of words in the tweet in Model 3. This normalization improved performance as seen in the bubble plot, but the negative log-score of Model 3 is slightly larger than Model 2, but still less than the naive model's score. This indicates that most of the predictive power comes from the tweeter and retweeter.

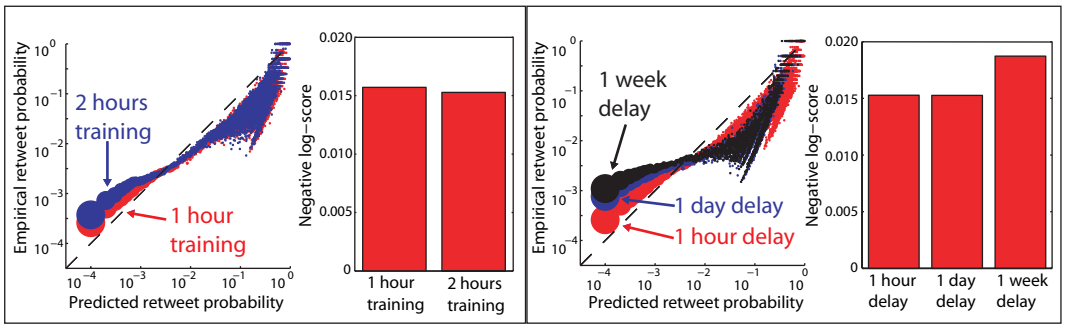

Figure 2: (left) Bubble plot and negative log-score of Model 3 with one and two hours of training data. Predictions are made for tweets from the hour after the training data. (right) Bubble plot and negative log-score of Model 3 with two hours of training data. Predictions are made for tweets 1 hour, 1 day, and 1 week after the training data.

We wished to see if using more training data improved performance. To test this, we trained Model 3 with one hour and with two hours of training data and predicted on the subsequent hour. The bubble plot and negative log-score are shown in Figure 2. As can be seen, when we increase the amount of training data, the performance of the model improves slightly.

We also wished to see for how far into the future the model would be accurate. We took Model 3 with two hours of training and predicted for an hour of tweets from the following hour, following day, and following week. The results are shown in Figure 2. Here we see that the model performance is roughly constant for up to a day later, but the negative log-score begins to increase for predictions on tweets occurring a week later.

4 Conclusion

We have presented here a methodology for predicting retweets in Twitter. We use retweets as positive feedback and lack of retweets by followers in the retweet network as negative feedback. The relevant features for prediction are the tweeter and retweeter. Our methodology is very flexible and allows for improvements on our current models by incorporating information such as rates of tweets on certain topics or correlations in retweets.

References

[1] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a Social Network or a News Media?", *Proceedings of the 19th international conference on World Wide Web*, pp. 591-600, 2010.

[2] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi "Measuring User Influence in Twitter: The Million Follower Fallacy", *Proceedings of international AAAI Conference on Weblogs and Social*, (2010).

[3] D. Boyd, S. Golder, and G. Lotan "Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter", *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, pp. 1-10 , 2010.

[4] S. Asur and B. A. Huberman, "Predicting the Future with Social Media", *preprint arXiv,*:1003.5699, 2010.

[5] D. Stern, R. Herbrich, T. Graepel, "Matchbox: Large Scale Online Bayesian Recommendations", *Proceedings of the 18th international conference on World Wide Web*, pp. 111-120, 2009.