International Journal of Advanced Intelligence
Volume 0, Number 0, pp.XXX-YYY, November, 20XX.

AIA International Advanced Information Institute

Predicting Popularity of Messages in Twitter using a Feature-weighted Model

Yang Zhang, Zhiheng Xu, Qing Yang
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
Beijing, China, 100190
yagzhg, xuzhiheng, qyang@nlpr.ia.ac.cn

Twitter is the hottest microblogging service nowadays and studying the characteristics of popular messages in Twitter is important in many different fields such as viral marketing, personalized messages recommendation, breaking news detection, etc. This paper investigates the problem of predicting the popularity of messages as measured by the number of future retweets. We formulate the task into a multi-class classification problem and specially bring into the extra feature-weighted mechanism to account for the fact that different features in Twitter have extremely different impact on retweeting. Experimental results show that the feature-weighted model has a quite good performance in the prediction task and outperforms other non-weighted models.

Keywords: Twitter; Tweet Popularity; Feature-weighted; Multi-class Classification;

1. Introduction

Twitter has recently emerged as a popular social microblogging service where users share and discuss about everything, including news, anecdotes, and even their own life stories. Since its creation in 2006, Twitter has experienced an exponential explosion in its tweet amount, generating approximately 200 million tweets per day as of 2011 [1]. Due to the high volume of messages, users in Twitter will probably experience the problem of information overload, and it is hard for them to find useful information. On the other hand, users might miss some important or interesting messages due to the limitation of information diffusion trees. To address these two problems, it is necessary to determine the popularity (i.e. importance or interestingness) of messages in Twitter, which is the focus of this paper.

As messages in Twitter get popular through the mechanism of retweeting, we naturally use the number of retweets as a measure of popularity. While much work has been conducted for predicting whether a tweet will be retweeted,[2] [3] [4] [5] [6] few publications study the actual number or the volume range of future retweets for a new message.[2] Suh et al. studied a variety of factors that might influence retweets and found that different features had extremely different impact on retweeting, e.g., whether a tweet contains URLs apparently plays a much more important role in retweeting than whether this tweet contains some meaningless word like "hi".[4] Although this phenomenon is quite ordinary, it is much more obvious in the activity of retweeting in Twitter. Based on these findings, We emphasize the differences of impact of different features specially in our prediction task and propose a feature-weighted model, in which the important features will play a more important role

compared with in those non-weighted models.[2] The fundamental questions in this paper are: (1) Can we accurately predict the popularity of messages in the form of volume range of retweets using a set of tweet and author features? (2) Is our feature-weighted mechanism effective as expected in actual prediction task?

2. Preliminary

In this section, we outline the actual research questions and describe our data set briefly.

With the purpose of predicting popularity of messages, We cast the task into a multi-class classification problem which predicts the volume of retweets a particular message will receive in the near feature, rather than the exact number of retweets. We divide the messages into four different retweet volume classes: 0: not retweeted, 1: retweeted less than 10, 2: retweeted less than 100, 3: retweeted more than 100.

We used Twitter's streaming API[a] to collect a random sample of the public tweets from April 11, 2011 to April 20, 2011, yielding 14,315,528 tweets (approximately 1.5M tweets per day). Twitter has almost 200M tweets daily, so we estimate that the data set represents about 0.8% of all tweets. After getting rid of the non-English tweets, we finally get 9,772,457 tweets, including 8,636,643 class 0 tweets, 621,154 class 1 tweets, 373,725 class 2 tweets and 140,935 class 3 tweets. The tweets from April 11 to April 18 are taken as train set, and the remainder is test set.

3. Predicting Popularity of Tweets

In this section we elaborate how to automatically predict the popularity of tweets in detail. Given that different features have extremely different impact on retweeting, we consider the influence of features specially and propose the feature-weighted model. Just as implied by the name, the feature-weighted model consists of three fundamental elements: features, the basic non-weighted model, and the weight mechanism, which will be discussed respectively in the following sections.

3.1. *Features*

As humans are indeed capable of estimating popularity of tweets just based on the tweet itself and author of the tweet,[3] we divide the Twitter features into two distinct sets: **user features** (features related to author of a tweet), and **tweet features** (which encompass various statistics of the tweet itself).

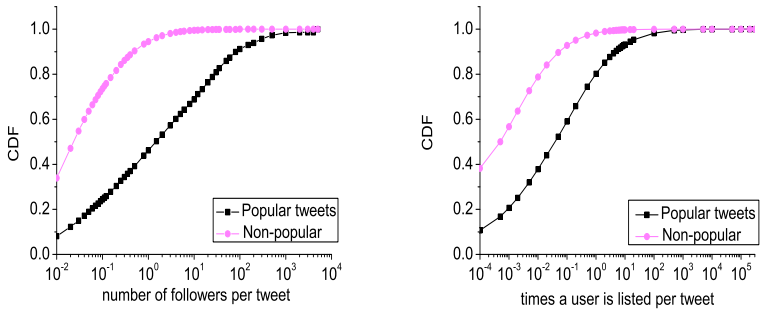
We select the following 11 features referred to the author of the tweet: **number of followers, friends, past tweets, favorites, number of times the user was listed, account days, user activity, length of user's screen name, and whether the user is verified, average number of followers accumulated by a tweet, average times a user was listed through a tweet.** Most of these features have been mentioned in prior work, except the last two features, which probably have a quite strong positive correlation with the popularity

[a] http://apiwiki.twitter.com/Streaming-API-Documentation

of the user's tweets. As we know, once a user publishes a popular tweet, either interesting or important, then he will get many new followers and be listed more. If a user has many followers or been listed many times only with a small number of tweets, this should mean that the user's tweets generally have quite high quality and popularity. The cumulative distribution function (CDF) for these two features are shown in Figure 1.

We use the following 8 features related to the tweet itself: **number of URLs, hashtags, mentions, words, characters, whether the tweet is a reply, whether the tweet was retweeted before, time the tweet was published.** Notably, we don't consider any content features such as topics or sentiments in tweets. URLs, hashtags, mentions and reply were already shown in previous work to have a high correlation with retweetability. We note that, popular tweets usually carry more information compared with non-popular tweets, with more words and characters in them. In addition, as the number of active users on Twitter varies with time, the time a tweet was published has a great impact on the final popularity of this tweet. The feature whether the tweet was retweeted before is correlated with the novelty of the tweet and reflects the quality of the tweets from a side view.

All these features mentioned above will be scaled and some of them might need further transformation (e.g., log-transformation, to account for the magnitude or skew).



(a) (b)

Fig. 1. Cumulative distribution functions of features

3.2. *Non-weighted Model*

Hong et al. predicted the popularity of tweets using Logistic Regression classifier. [2] However in this paper, to address the multi-class classification problem, we use the Support Vector Machine (SVM) classifier as our non-weighted model, which is a state-of-the-art method in classification and will obtain better results shown in later experiments. The goal of a SVM is to find the hyperplane that optimally separates the training data into two portions of an N-dimensional space with a maximum margin. We use a non-linear SVM with the Radial Basis Function (RBF) kernel that allows SVM models to perform separations

with very complex boundaries. The implementation of SVM is provided with LibSVM, [7] an open source SVM package that provides a series of optimizations.

3.3. *Feature-weighted Mechanism*

To emphasize the fact that different features have extremely different impact on popularity of messages, the features will be assigned corresponding weights based on their importance in impacting retweeting. Then the weights will be brought into the basic non-weighted model as extra adjustment factors. Information Gain (IG) is applied to determine the influence that different features have on retweeting, and intuitively the weights for different features should be decided according to their IG scores. We first compute the average IG score IG_{mean} for all these 19 features, and then the weight for feature f can be calculated as follows:

$$weight(f) = \sqrt[n]{IG(f)/IG_{mean}} \tag{1}$$

$IG(f)$ is the IG score of feature f, and n is used to adjust the intensity of this weight mechanism. The smaller n is, the greater the influence of this mechanism will be. We can note that if a feature has greater impact on retweeting compared with average features, the weight of this feature will be bigger than 1, meaning that the feature will get strengthened further in this feature-weighted mechanism compared with in non-weighted models.

4. Experiments

The experiments are conducted on the Twitter dataset mentioned in Section 2. In this section, we first give the ranking that different features impact retweeeting. Then we investigate the influence of parameter n in the feature-weighted model. Finally the prediction results of our method are presented as well as other models.

4.1. *IG Ranking of Features*

Information Gain can be applied to measure the impact that different features have on retweeting. A high IG score indicates that the feature has a greater impact on retweeting. Table 1 presents the ranking that features impact on retweeting based on their IG scores.

Table 1 shows that different features indeed have very different impact on retweeting, e.g., the IG score of feature at position 1 is two orders of magnitudes bigger than that of feature at position 19. In addition, we note that the top two of the table are features that reflecting users' influence, followers and times that a user was listed. Following them are the two features that we propose first, average number of followers per tweet and average times a user was listed per tweet, verifying that the two features indeed have very great impact on retweeting. Other features that we use first, i.e., number of words, characters, account days, time the tweet was published and whether the tweet was retweeted before, are all distributed in the middle of the table (position 12, 13, 9, 11, 10). Surprising, User's activity has little impact on the popularity of his tweets as well as number of URLs (position 18, 19). The ranking provides us a general picture about the importance of different features and the IG scores will be used to calculate the weights in our model.

Table 1. Ranking of all attributes

Position	Features and their IG scores
1	Number of followers (0.1677)
2	Number of times the user is listed (0.1583)
3	Average number of followers per tweet (0.1392)
4	Average times user was listed per tweet (0.1239)
5	Number of mentions in a tweet (0.0521)
6	Whether a tweet is a reply (0.0431)
7	Whether a user is verified (0.0363)
8	Number of hashtags in a tweet (0.0224)
9	Account days (0.0206)
10	A tweet was retweeted before or not(0.0206)
11	Time a tweet is posted (0.0204)
12	Number of words in a tweet (0.0204)
13	Length of a tweet (0.0189)
14	Number of friends (0.0181)
15	Number of favorites (0.0072)
16	Number of past tweets (0.0052)
17	Length of user's screen name (0.003)
18	Number of URLs in a tweet (0.0016)
19	Number of tweets per day (0.001)

4.2. *Selection of Parameter n*

As the intensity of feature-weighted mechanism varies with the value of n in equation (1), how to select appropriate parameter n so that our feature-weighted model can have the best performance is the next challenge. Figure 2 shows the prediction results when we vary the parameter n.

Figure 2 shows that feature-weighted model gives a best performance when n is around 2. We note that when n is bigger than 10, the prediction results nearly remain constant. The reason may be that the weights of each feature are very close to 1 when n is very big (e.g., bigger than 10), and in this case, the feature-weighted model almost relaxes to a non-weighted model. Additionally, we also notice that the performance drops dramatically with the decrease of n when n is smaller than 1. The reason probably is that the weights of some most important features are much heavier than those of less important features in this case, and finally the feature-weighted model will relax to a only-one-feature model. n is selected as 2 by default in the following section.

4.3. *Results of Popularity Prediction*

To evaluate the performance of our feature-weighted model, we use the standard information retrieval metrics such as $precision$, $recall$ and $Macro\text{-}F_1$. Besides our feature-weighted model, two other methods are applied in the prediction task as the base line:

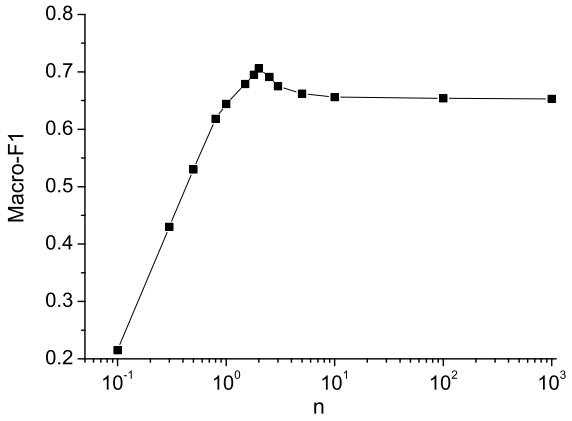


Fig. 2. Prediction results of different parameter n

Logistic Regress Model and SVM classifier. The first one was widely used in previous retweeting prediction problems [2 5 6] and the second is the non-weighted model used in our feature-weighted model. The prediction results are shown in Table 2.

Table 2. Results of prediction

Methods	**Precision**	**Recall**	**Macro-F_1**
Logistic Regression	0.6016	0.5237	0.5599
SVM classifier	0.7081	0.6026	0.6511
Feature-weighted model	**0.7669**	**0.6534**	**0.7056**

From Table 2 we can note that SVM outperforms Logistic Regression method, this is why we select SVM as the non-weighted model in our method instead of Logistic Regression. In addition, we also note that our feature-weighted model performs best in the prediction task. The results of Table 2 successfully answer the two questions proposed at the front of this paper: the popularity of messages in Twitter can be accurately predicted and the performance will get further improved if the feature-weighted mechanism is specially applied.

Table 3 further shows the prediction results for each class using our feature-weighted model. We note that our method performs very well in class 0 and 3, but badly in class 1 and 2. This huge discrepancy among classes probably results from the following fact. Class 0 and 3, which represent non-popular and popular tweets respectively, have a more widely range in popularity than class 1 and 2. If class 1 and 2 are merged into one, it is expected that the accuracy will get improved, however at the cost of obscuring the degree of popularity. Actually, accurately distinguishing popular tweets from those non-popular

Table 3. Prediction results for each single class

Class	**Precision**	**Recall**	**Macro-F_1**
0	0.8975	0.9810	0.9374
1	0.8302	0.4841	0.6116
2	0.5012	0.2167	0.3026
3	0.8387	0.9716	0.9003

ones is usually much more meaningful and important than separating class 1 and 2.

5. Conclusion

A fundamental task in Twitter is predicting the popularity of tweets, which is the focus of this paper. Given that different features in Twitter have extremely different impact on retweeting, we brought into the feature-weighted mechanism, which is a reward mechanism in actual, namely, making the important features play a more important role in prediction. Experimental results showed that our feature-weighted model had the best performance in the prediction task.

Nevertheless, as an early attempt to predict the popularity of tweets in Twitter, our work still has space for improvement. We envision two directions towards which our work can evolve. First, we intend to bring into some content features such as the topics and sentiments in tweets; second, given that different features don't have independent impact on retweeting, we plan to investigate the correlation among features.

References

1. "Your world, more connected." http://blog.twitter.com /2011/08/your-world-more-connected.html.
2. L. Hong, O. Dan, and B. D. Davison, "Predicting popular message in twitter," in *WWW'11*, 2011.
3. S. Petrovic, M. Osborne, and V. Lavrenko, "Rt to win! predicting message propagation in twitter," in *AAAI'11*, 2011.
4. B. Suh, L. Hong, P. Pirolli, and E. H. Chi, "Want to be retweeted? large scale analysis on factors impacting retweet in twitter network," in *IEEE Second International Conference on Social Computing*, 2010.
5. N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi, "Bad news travel fast:a content-based analysis of interestingness on twitter," in *ACM 3rd International Conference on Web Science*, 2011.
6. L. K. Hansen, A. Arvidsson, F. A. Nielsen, E. Colleoni, and M. Etter, "Good friends, bad news affect and virality in twitter," in *Internatonal Workshop on Social Computing, Network and Services*, 2011.
7. R. Fan, P. Chen, and C. Lin, "Working set selection using the second order information for training svm," *JMLR*, 2005.