International Journal of Hybrid Information Technology
Vol. 9, No.6 (2016), pp. 327-336
http://dx.doi.org/10.14257/ijhit.2016.9.6.29

Information Diffusion Temporal Dynamic Prediction in Microblog System Based On User Influence Learning

Kechen Zhuang [1]*, Fawang Han[2], Haibo Shen[1], Kun Zhang[1] and Hong Zhang[1]

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China [1]
School of Information Technology, Nanjing Forest Police College, Nanjing, China [2]
*xjzkcgf@126.com**

Abstract

Information diffusion in online social network especially in microblog system, can largely affect the public opinion and even the development of events, so the prediction of the future dynamics of diffusion could be very valuable at many aspects. In this research, a novel graph-based cascades construct algorithm is proposed, with which we build a prediction model for future information diffusion dynamics. By learning user influence features that related with the network topology and users' interactions from a large scale real Weibo dataset, we successfully predicted the short term topic related tweets population dynamics.

Keywords*: Microblog, Information diffusion, Tweets prediction.*

1. Introduction

Online social network now become the most well-known online social media (Such as Twitter, Weibo, *etc.*) where information is generated and consumed by millions users, and updated by others through commenting, replying, reposting, *etc.* Similar to the word of mouth, information in the network can pass from a user to others, which the process is known as information diffusion. Since every user is both communicator and consumer, and the Internet as the intermediary, the information spreading in network is amplified in both speed and range than offline, which cause the outbreak of the news nowadays is easier to happen.

The information diffusion is the study of how the information or innovations propagate in the networks. The diffusion process or the propagation has been studied in areas like epidemiology for decades and many recently information propagation researches also reuse the epidemiology methods to explicate the mechanism of diffusion in other environments. However, with the rapid growth of online social network, the information diffusion in the network become more complex and dynamic, because there are large scale of users, wide diversity in their profiles, wide dynamics in their behaviors, and still not an universal understanding for modeling and capturing the diffusion pattern. Therefore, a better model for the information diffusion approximating and predicting is much needed.

The diffusion process in network usually is affected by structure of the network. In the online social network, with the massive users scale, the intertwined users' relationships and the dynamics temporal changes caused the network structure tangled and elusive. Therefore, to better understand the diffusion process, the mining of complex network structure is not open around the problem. There are studies research on the issues about the interplay between the network structure and the diffusion process [1] [2], trying to figure out the different structural characteristics of the network impact on the information diffusion. Yet it is still an open problem. Most studies only focus on static network structure and single way diffusion, but the reality online social network, especially like the user relation networks in Weibo, are very dynamic. Which means the network evolution and temporal dynamics also need to take full account when modeling the information diffusion.

ISSN: 1738-9968 IJHIT

The users' influence is also an important factor that can greatly affect the information diffusion process. Differ from some online social networks like Facebook, whose users' relation network are relatively static and evaluating slowly, the online social media like Weibo has highly dynamic users' relation network and play the role of real-time media in the Internet. The famous users often have higher and faster influence on their followers. While the individual influence could highly affect the information diffusion process, the information posted by influential user usually could be heard and reposted by more users. The user influence could be quantized in some extent, according to researchers whose researches focus on measuring the user influence based on network structure features, individual diffusion features and content dynamics [3] [4] [5].

By considering the user influence and network structure, many propagation models were proposed based on epidemiology, such traditional model rarely considered the temporal features of diffusion. While the information diffusion often creates sudden bursts in very short period and cause a cascading effect [6], which could cause enormous public opinion influence. The temporal dynamics of the diffusion network should be taken seriously when modeling the diffusion.

In this paper, we study the interplay between users' influence and the information diffusion. By proposed an algorithm for diffusion cascades construction and a diffusion prediction model based on decision tree, we analyzed the Weibo dataset, analyzed the multiple user influence related features, trained the model from the learning dataset, and successfully predicted the temporal dynamics of the topic tweets population.

2. Related Works

Diffusion processes in networks have been studied in a variety of different areas. In epidemiology, modeling and dynamics of infectious diseases is well-studied with a rich literature. A number of those models have been used to model the information propagation in social networks as a viral spread process [7][8]. But in some studies, the infected nodes generated by the epidemic models either majority or minority, which is not comply with the information diffusion observations from microblog networks.

There are several types of diffusion models have been proposed. The Independent Cascades model (IC) [9][10] associates a fixed propagating probability with each edge and each node could attempt infecting its neighbors only once. In the Linear Threshold model (LT) [11], each node associated with a threshold value. If the number of infected neighbors of a node exceeds the threshold then the node itself becomes infected. The propagating parameters in those models are usually fixed or came from the analysis of the users' behaviors evaluation. The simulations with those models are based on graph with small scale node number, which may not reflect the real process of information diffusion in large-scale network.

Researchers have already did some works on diffusion prediction in social network [12][13], but some of them rely solely on network structure or tweets content, and test the proposed model on small-scale dataset. So in this paper, we overcome the incomplete data difficulty, and successfully predict the short term temporal dynamics of tweet population, which could help monitoring of public opinion, especially in the case of outbreak event.

3. Data Processing

Due to restrictions mechanism of Sina Weibo API and user privacy protection, the massive original data is difficult to crawl. In this research we use anonymous data set [14]. The dataset details are summarized in Table 1.

Table 1. Sina Weibo Dataset

Users	Tweets	Links	Topics
58,655,849	369,797,719	265,580,802	203

There are over 58 million anonymous users in this dataset which is a sample of the whole 300 million, but it is a relatively large network, and the diffusion on such a large network would be much closer to the real environment. The follow relationship between users formed a directed follower graph. There are 369,797,719 tweets have been posted (retweets included) by more than 1.3 million users in this dataset. We measured the retweet times and the retweet interval in Figure 1.

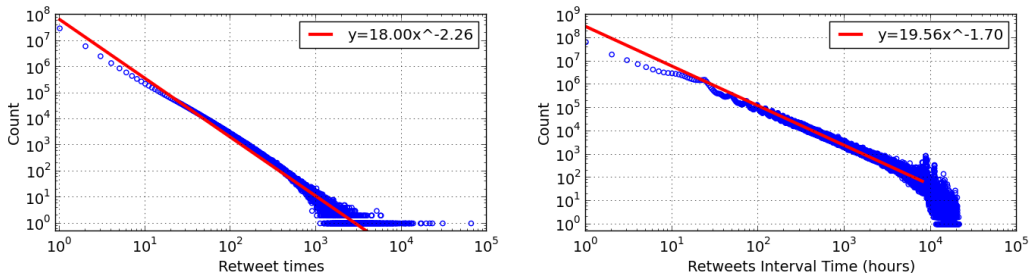

Figure 1. Retweet Count and the Retweet Interval

Our research mainly focus on the information diffusion temporal prediction, so we investigate the topics temporal dynamics and the retweet temporal dynamics, by mining on the data set we have, and plot the result in Figure 2. In the top picture in Figure 2, we plot the population time series for 5 popular topics. For the outbreak event topics, it's clear to find the multi-peak pattern, and the pattern is also fit for the retweets temporal dynamics.

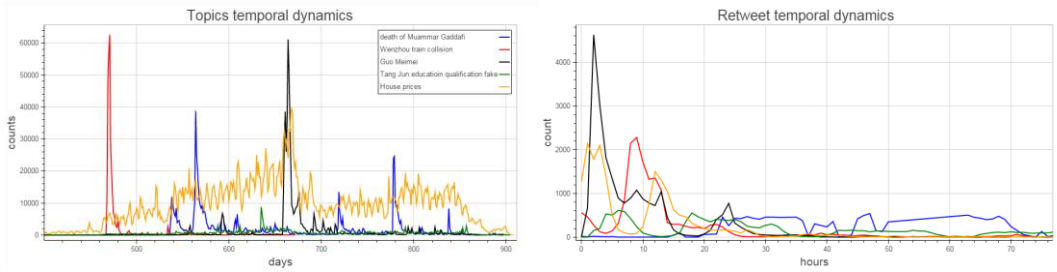

Figure 2. Several Topics and Retweets Temporal Dynamics

4. Diffusion Cascades

In the microblog users follower/followee relationship network, users are the nodes, and the follow relations among them could between the direct links between nodes Let $G=(V,E)$ be a relationship network, in which the V is a set of users, E is a set of links between users, $e(v,u)\in E$, where $v,u\in V$, denotes that a directed link from v to u.

After an user of the Weibo posts a tweet, his or her follower users may have chance to read this tweet and retweet it if they are interested, and then the tweet would spread to their follower users, which causes large scale diffusion. The diffusion process in the networks left the traces in form of hierarchical trees. Specifically, the retweet process in the microblogs formed retweet cascades which presented in the form of tree. A cascade can be represented as $C=(m_o,M,R)$, where m_o is the root node or the tweet source, which means all tweets in this cascade are retweeted from m_o directly or indirectly. M is the set of participated tweets,

$m_i \in M$ is the participated tweet has the basic feature $m_i(v_i, t)$ and could be represented as m_i^t, where v_i is the user who post the tweet and t is the post time. R is the set of links between tweets defined the retweet relations, $e(m_v, m_u) \in R$ is an edge represent the retweet relation between tweets where $m_v, m_u \in M$, and t is the retweet time.

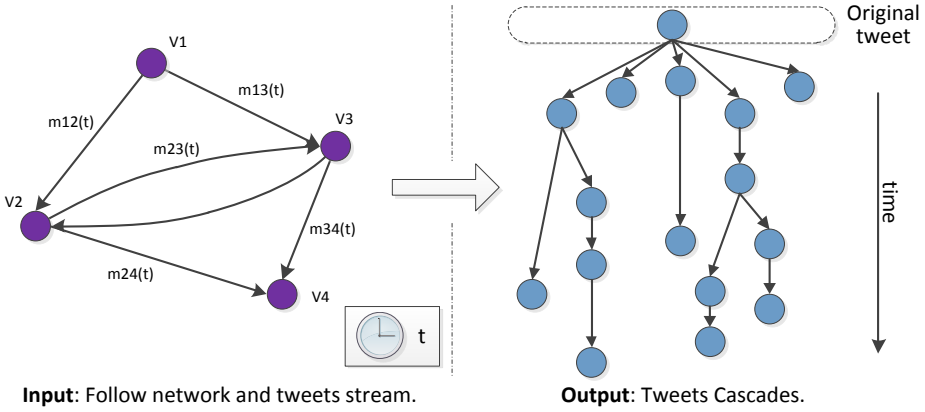

Figure 3. Construct Cascades from Tweets Stream

The structure of retweet cascades is based on the follower/followee relationship between users in the microblog network, which is the direct graph $G = (V, E)$ defined above. In order to analyze the characteristics of cascades to further modeling and prediction, we need to construct the retweet cascades from the tweets stream.

The real tweets stream crawled from the Weibo is incomplete due to the restrictions of privacy and permission issues, so there is needed to fill vacancy when constructing the retweet cascades. For example, in the tweet stream, there is a tweet m_i^t was tweeted by user v_i at time t, then there is another tweet $m_k^{t+\Delta t}$ is retweeted from m_i^t by user v_k at the time $t + \Delta t$, but in the relationship network $G = (V, E)$, the user v_k is not the follower of v_i but the follower of v_j, and the user v_j is the direct follower of v_i. Then based on the relationship network, we can assume that there could be a tweet $m_j^{t+\Delta t'}$ is retweeted from m_i^t by user v_j at the time $t + \Delta t'$, where $\Delta t' < \Delta t$, and for simplicity, we set $\Delta t' = \Delta t / 2$.

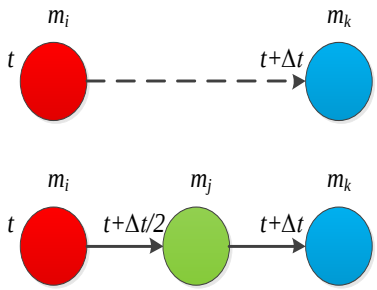

Figure 4. Estimate Latent Retweets

In order to simplify the algorithm, the tweet stream need be classified before the cascade construction. We classify the tweets data into tweet lists with the same source tweet, which means the tweets in same list have the same original tweet they retweeted from.

Algorithm 1. Retweet Cascades construction

Input: *TweetList sort by tweet time, original tweet* $m_o(v_o,t)$, *Relationship network G=(V, E)*
Output: *Retweet cascade* C_i

```
1   sort TweetList by tweet time.
2   Construct a new cascade Ci(mo,M,R), set the mo as the root node.
3   while TweetList not empty do
4     Pop the first tweet mj(vj,tj) from the TweetList.
5     Search in the cascade Ci for the latest node mi (vi,ti)near mj.
6     if vi is directly followed by vj in G then
7       add mj to M, add e(mi,mj) to R.
8     else
9       get the follow NodeList between vi to vj.
10      count = 0; length = number of nodes in NodeList.
11      for each node vk in NodeList do
12        count =count+1.
13        form new tweet mk(vk,tk), where tk = ti + (tj-ti)* count/(length+1).
14        if vk is the last node in NodeList then
15          add mj to M, add e(mk,mj) to R.
16  return Ci.
```

With a tweet list sort by tweet time as the input, we apply the following algorithm to build the cascade based on the relationship network *G*, and the output will be a cascade C_i. If the original tweet is $m_o(v_o,t_o)$, then set the root node of C_i as m_o. When a tweet $m_j(v_j,t_j)$ which is retweeted from m_o was processed, we search in the relationship network G, if the node v_j is the direct follower of node v_o, add the tweet m_j and an edge $e(m_o,m_j)$ into C_i. Otherwise, if there are follow relationship in *G* like $v_o \to v_1 \cdots v_i \cdots v_n \to v_j$, then search in C_i to find latest node near to m_j. If the latest node near to m_j is the node $m_n(v_n,t_n)$ which is directly followed by m_j, we add the tweet m_j and an edge $e(m_n,m_j)$ into C_i. Otherwise if $m_i(v_i,t_i)$ which is not the directly followed by m_j is the latest node, we need to assume users $v_i \cdots v_n$ all retweeted the tweet m_o, for simplicity, we assume retweet time intervals are equal, here we construct the retweets $m_i(v_i,t_i+\frac{t_j-t_i}{n-i+1}),\cdots,m_n(v_n,t_i+(n-i)\frac{t_j-t_i}{n-i+1})$, and add all retweets $m_i,\cdots,m_n$ into C_i, also add the edges $e(m_i,m_{i+1}),\cdots,e(m_{n-1},m_n),e(m_n,m_j)$ into C_i.

5. Modeling

Based on the retweet cascades we generated with the algorithm above, our proposed model mainly focus on learning the user influence related features to train predict model, so as to predict the future retweet dynamics.

5.1. Features Analysis

The diffusion between users depends on different features which can be classified into social features, content features and temporal features.

Social Features.

(1) *Popular degree* (s*P*): When the diffusion happen between user *u* and *v*, generally the more influence *u* has, the large probability of diffusion for *u* to *v*. While the influence of a user on another has been studied in many articles. Here as a single influence, we

simply set the followers number as the popular degree of a user. We set $\delta = 1000$ as a standard, for a user *u*, $sP(u) = \frac{|F(u)|}{\delta}$ if $|F(u)| < \delta$, otherwise $sP(u) = 1$.

(2) *Tweet frequency* (*sT*): The tweet frequency could be a good measure for user's activity. Since a more active user could have larger probability to read and retweet messages. We use the average hourly tweets number of a user to represent its tweet frequency. For user *u*, $sT(u) = \frac{|M(u)|}{\sigma}$ if $|M(u)| < \sigma$, otherwise $sP(u) = 1$, with σ is the total hours in the learning dataset.

(3) *Interaction* (*sI*): The interaction between two specific user *u* and *v* is defined to measure frequency of user *v* retweeted from *u* directly. And it's computed by the ratio of the amount of *v* retweeted from u and amount of *v*'s total tweets. For sender user *u* and receiver user *v*, $sI(u,v) = \frac{|rM(u,v)|}{|M(u)|}$.

(4) *Mutually follow* (*sF*): If user *u* and *v* followed each other, there is a great probability that these two users are friend in real life, which also lead to a great probability that the tweet of user *u* would be retweeted by user *v*. Here the mutually follow is represented as a binary value. $sF(u,v) = 1$ if $v \in F(u)$ and $u \in F(v)$, otherwise $sF(u,v) = 0$.

Content Features.

(5) *Topic* (*cT*): The topic in our dataset means the popular topic words people discussed. If the tweet contains a popular topic, it has larger probability to be retweeted. The topic feature here is also represented in a binary way. $cT(u, m_u) = 1$ if $topic(m_u) \cap T \neq \varnothing$, otherwise $cT(u, m_u) = 0$.

(6) *Urls* (*cU*): If the tweet posted by user *u* contain some urls, which could be links to interesting contents that increase retweet probability. $cU(u, m_u) = 1$ if $urls(m_u) \neq \varnothing$, otherwise $cU(u, m_u) = 0$.

Temporal features.

When user *v* posted a tweet, the Weibo system will immediately send it to *v*'s follower *u*. If *u* just also online during that time period, the tweet has a great probability would been seen and retweet by *u*, otherwise the tweet is easy to sink down in the *u*'s read-list, and hard to been seen and retweeted.

(7) Period score (*tP*): We divide a day into 6 time periods, each has 4 hours, if in the learn data set, the user *v* has posted 40 tweet during 8am-12am, and 60 tweets during 4pm-8pm in total, then we get a temporal frequency list: $L_v = [0, 0, 0.4, 0, 0.6, 0]$. For a user *u* post a tweet at time *t*, we set the time period score for its follower *v*. $tP(v,t) = L_v(\lfloor t/4 \rfloor)$.

(8) Delay from origin (tD): Through statistical analy4sis of retweet dataset, we find that users' interests on a certain tweet would decrease significantly with the passage of time, which means as the time increases, the probability of forwarding will continue to decrease. So the time delay from the original tweet is an important feature to measure the retweet probability. $tD(u, m_u) = \frac{t(m_u) - t(m_o)}{\sigma}$, where $t(m_o)$ is the tweet time of the original tweet and the σ is the total hours in the learning dataset.

After the features space is constructed, the predicted model and the time delay can be learned and estimated. This is performed through machine learning techniques and is detailed in the following sections.

5.2. Diffusion Instance Dynamic Prediction

In order to predict future retweet population, in our research, we use the micro-level diffusion dynamic prediction to estimate the future retweet population. Which is, for a tweet

m_u been posted by user u, we will analyze the related features defined above, then process the prediction on each of u's followers to estimate wither m_u been retweeted or not, which is an instance of diffusion or non-diffusion.

Table 2. Properties of the Experimental Sub-Networks

Sub-Net	Users	Tweets	Links	Time	Topics
1	3,970,720	8,568,778	10,436,686	2011-04-01 to 2011-05-30	72
2	2,943,939	9,536,106	12,349,740	2011-09-01 to 2011-10-30	90

First, we build a subset of data comprising 2 experimental sub-networks. Since the tweets in dataset is relatively sparse compared with real social network, which is due to the microblog crawling limitations, we build the sub-networks in a way that make sure all the users in sub-networks did post or retweet some tweets during the given two months. The detail information of the sub-networks is shown in Table 2. In each network, we capture the main topics during sampling time, and building the information diffusion cascades by using the algorithm we proposed in section 4, to simulate the real diffusion process as much as possible. Then we labeled each diffusion instance or non-diffusion instance with 8 features in the context of mining the learning dataset. The features' mean and standard deviation of learning dataset is shown in Table 3.

Table 3. Means and Standard Deviation of the Features in Dataset

Features	*sP*	*sT*	*sI*	*sF*	*cT*	*cU*	*tP*	*tD*
Mean	0.043	0.056	0.001	0.001	0.016	0.087	0.113	0.035
Stdev	0.106	0.075	0.001	0.001	0.127	0.283	0.104	0.124

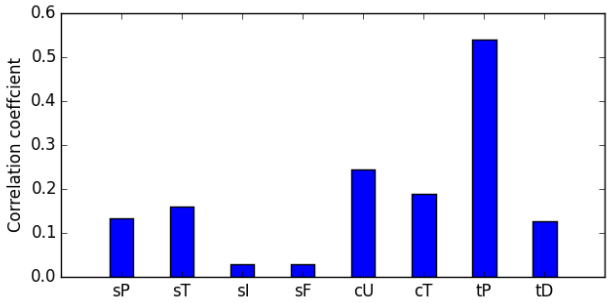

Figure 5. Correlation between Features And Diffusion

The prediction problem then becomes a binary classification problem, which given 8 features to predict the instance is diffusion or non-diffusion. In our research, we choose a decision tree classification algorithm on this machine learning supervised task P ($D|F$), in which D = {diffusion, non-diffusion} and F is the 8-dimensional feature vector. The mean and the standard deviation of the features shown in Table 3 are computed by analyzing the learning dataset. The correlation between features and diffusion instance is shown in Figure 4. Taken together, we can see that *tP* is the most relevant feature which because it combined two users relevance.

The accuracy rate of decision tree binary classification on two sub-networks is shown in Table 4, both are relatively high.

Table 4. The Decision Tree Classification Accuracy Rate on Two Sub-Networks

Sub-Net	Classification accuracy rate
1	91%
2	86%

5.3. Diffusion time prediction

For the diffusion instances in the dataset, we know the time users tweet and retweet, so it is able for us to compute the real time delay for the .The time delay is mainly depends on the *tP* feature between two users, with the following formula: $td(u,v,m_u) = \alpha \cdot tP(v,t_{m_u}) + \beta$.

Then for each instance of diffusion, we can use a least-squares solution for the linear fitting of the real time delay data, and get the coefficients for the formula above. In the experiment, we artificially set ranges for α and β ($\alpha < 0$ and $\beta > \alpha$), which make sure the time delay could fall in (0, β).

6. Experiment

Since the task of analyzing features from the dataset is complex, especially in networks contain millions nodes, we process the whole experiments on distributed computing experiment platform that is based on Spark. In order to investigate the effectiveness of our model and algorithm, we choose the topic tweets population prediction as our main prediction task, which also can be more intuitive to reflect on the information diffusion dynamics in the microblog system.

We choose the most popular topics in two sub-networks respectively, which are "the death of Osama Bin Laden" in sub-net 1 and "death of Muammar Gaddafi" in sub-net 2. Then we choose the day after the topic event happen as the split date, to divide each sub-network to two dataset, respectively as learning dataset (tweets posted before the split date), and test dataset (tweets posted after the split date).

We use the diffusion cascades construction algorithm proposed in section 4 to build diffusion cascades from the learning dataset, then combined with the network topology structure to get all the diffusion and non-diffusion instances. Extracting the features from the instances, then learn the decision tree from all the diffusion and non-diffusion instances. After the decision tree model is created, we apply the model on the prediction of a specific topic population.

For example, In sub-network 1, we selected all the tweets with the topic "the death of Osama Bin Laden", the split-date is "2011-05-02", then we use the tweets posted before "2011-05-02 23:59:59", and build all possible diffusion instances, and use the decision tree model to predict the instances is diffusion or non-diffusion, then use the time delay formula to estimate the retweet delay, and we can get predicted retweets population in the next day (2011-05-03). Then we added the real topic tweets posted in "2011-05-03" into the selected tweets dataset, and predict retweets population in next day (2011-05-04), and so on. Finally get the retweets population in each day within the test dataset.

We plot the comparison of real and prediction time series for two topics in two sub-networks in Figure 6 and Figure 7. The real tweets population is the blue line and the predicted population is the red dash line, as we can see our prediction captured the variations well while slightly underestimated the tweets population. This is mainly due to the presence of the external influence [15], since there are many different media exist in real world, the microblog users could be influenced by external news and post more tweets about the same topics. Both results shown the real and predicted population have two or more peaks in the time series, which meet the two step of the information diffusion, that the topic information spread from other media to few popular users in microblog system, then it diffusion to mass

users (this may be the first peak of the tweets population), and then it may spread and discussed in some communities of mass users (this may be the two or more peaks). The two or more large peaks may also cause by new developments of the topic events.

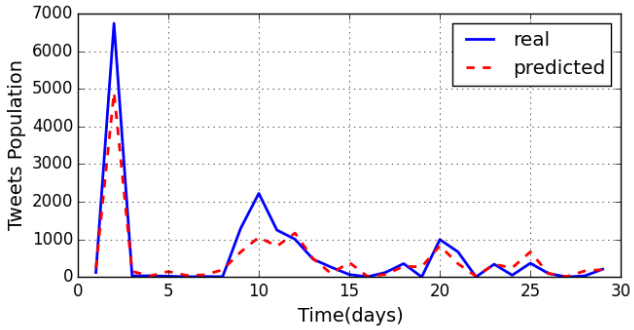

Figure 6. Comparison of Real and Predicted Time Series for the Topic "The Death of Osama Bin Laden"

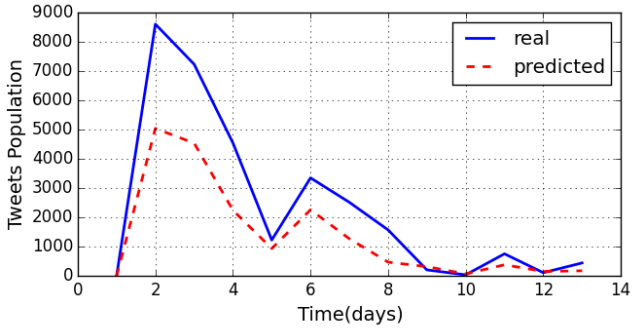

Figure 7. Comparison of Real and Predicted Time Series for the Topic "Death of Muammar Gaddafi"

7. Conclusion

In this paper, by analyzing the large scale microblog data set, we proposed a cascades construction algorithm and successfully build information diffusion cascades, from which we extract the real and latent diffusion or non-diffusion instances. By analyzing mutilple features in learning dataset, we trained decision tree model to do the classification task, the prediction accuracy rate is relatively high, and finally, the predicted retweets population fit the real population in variations time series well too. The results shown that, in the condition of exclude the external influence, our algorithm and model can predict the short term retweets population well, especially on the prediction of diffusion variations and the peaks, which could be used as the basis for further event outbreak detection.

Acknowledgements

The work was sponsored by "The Fundamental Research Funds for the Central Universities" (No.LGYB201505) of the Nanjing Forest Police College in 2015, and sponsored by the National Nature Science Foundation of China (No.61300053).

References

[1] Ugander, Johan, L. Backstrom, C. Marlow and J. Kleinberg, "Structural diversity in social contagion", Proceedings of the National Academy of Sciences, vol.109, no. 16, **(2012)**, pp. 5962-5966.
[2] Rodriguez, M. Gomez, J. Leskovec, D. Balduzzi, and B. Schölkopf, "Uncovering the structure and temporal dynamics of information propagation", Network Science, vol. 2, no. 01, **(2014)**, pp. 26-65.
[3] Brown, E. Phil and J. Feng, "Measuring User Influence on Twitter Using Modified K-Shell Decomposition', In Fifth International AAAI Conference on Weblogs and Social Media. **(2011)**.
[4] Li, Xiang, S. Cheng, W. Chen and F. Jiang, "Novel user influence measurement based on user interaction in microblog", In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ACM, **(2013)**, pp. 615-619.
[5] V. Steeg, Greg, and A. Galstyan, "Information-theoretic measures of influence based on content dynamics" In Proceedings of the sixth ACM international conference on Web search and data mining, ACM, **(2013)**, pp. 3-12.
[6] Myers, A. Seth and J. Leskovec, "The bursty dynamics of the twitter information network", In Proceedings of the 23rd international conference on World Wide Web, ACM, **(2014)**, pp. 913-924.
[7] Hao, Wang, Y. Li, Z. Feng, and L. Feng, "ReTweeting analysis and prediction in microblogs: An epidemic inspired approach", Communications, China, vol. 10, no. 3, **(2013)**, pp. 13-24.
[8] Jin, Fang, E. Dougherty, P.Saraf, Y. Cao and N. Ramakrishnan, "Epidemiological modeling of news and rumors on twitter", In Proceedings of the 7th Workshop on Social Network Mining and Analysis, ACM, **(2013)**, pp. 8.
[9] Kawamoto and Tatsuro, "A stochastic model of tweet diffusion on the Twitter network", Physica A: Statistical Mechanics and its Applications, vol. 392, no. 16, **(2013)**, pp. 3470-3475.
[10] Wang, Dong, H. Park, G. Xie, S .Moon and M. -Ali Kaafar, "A genealogy of information spreading on microblogs: A Galton-Watson-based explicative model", In INFOCOM, 2013 Proceedings IEEE, IEEE, **(2013)**, pp. 2391-2399.
[11] Kimura, Masahiro, K. Saito, R. Nakano and H. Motoda, "Extracting influential nodes on a social network for information diffusion", Data Mining and Knowledge Discovery, vol. 20, no. 1 **(2010)**, pp. 70-97.
[12] Galuba, Wojciech, K. Aberer, D. Chakraborty, Z. Despotovic and W. Kellerer, "Outtweeting the twitterers-predicting information cascades in microblogs", In Proceedings of the 3rd conference on Online social networks, vol. 39, no. 12, **(2010)**, pp. 3âAS3.
[13] Kupavskii, Andrey, L. Ostroumova, A. Umnov, S. Usachev, P, Serdyukov, G, Gusev and A. Kustarev, "Prediction of retweet cascade size over time", In Proceedings of the 21st ACM international conference on Information and knowledge management,. ACM, **(2012)**, pp. 2335-2338.
[14] WISE 2012 Challenge. http://www.wise2012.cs.ucy.ac.cy/challenge.html, **(2012)**.
[15] Myers, A. Seth, C. Zhu and J. Leskovec, "Information diffusion and external influence in networks", In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, **(2012)**, pp. 33-41.

