BEEP: a Bayesian perspective Early stage Event Prediction model for online social networks

Xiao Ma, Xiaofeng Gao*, and Guihai Chen
Shanghai Jiao Tong University
yusufma555@sjtu.edu.cn, {gao-xf, gchen}@cs.sjtu.edu.cn

Abstract—In recent years, predicting future hot events in online social networks is becoming increasingly meaningful in marketing, advertisement, and recommendation systems to support companies' strategy making. Currently, most prediction models require long-term observations over the event or depend a lot on other features which are expensive to extract. However, at the early stage of an event, the temporal features of hot events and non-hot events are not distinctive yet. Besides, given the small amount of available data, high noise and complex network structure, those state-of-art models are unable to give an accurate prediction at the very early stage of an event. Hence, we propose two Bayesian perspective models to handle this dilemma. We first mathematically define the hot event prediction problem and introduce the general early stage event prediction framework, then model the five selected features into several continuous distributions, and present two Semi-Naive Bayes Classifier based prediction models, BEEP and SimBEEP, which is the simplified version of BEEP. Extensive experiments on real dataset have demonstrated that our model significantly outperforms the baseline methods.

I. Introduction

Through the years, the event prediction problem, which refers to predicting whether an observed event will become hot in the future, has continuously drawn scholars' attention. Popular events are meaningful to many services and applications, such as marketing, advertisement and recommendation systems, which can assist companies' strategy making and boost their profit.

Although scholars have proposed many powerful models for event prediction problem in online social networks, all of these models have the same limitation: unable to predict the popularity of an event at the very early stage. It is an intuitive idea to make the prediction as early as possible. This is very reasonable since the earlier the possible hot information is recommended to the users, the higher profit the company may gain. However, this is also a challenging task due to the following reasons. First, the frequency at which the events discussed by users always changes irregularly. Many classic forecasting models based on time series analysis need a complete observation over a full evolution period of the time series, and they will not be able to handle such data with a high noise. Second, the structure of the online social network is complex and dynamic. Extracting features in such complex networks may pose high time complexity. However, the prediction problem requires high timeliness, which can possibly be affected by the feature extraction process. Third and the most important, at the early stage when an event just emerges, the amount of the available information is highly limited and the temporal features, such as the sum of the discussion frequencies is indistinctive between hot events and non-hot events. Temporal features alone will no longer be able to support the prediction problem at the very early stage.

In order to overcome the above-mentioned challenges, the proposed model should have the following properties:

- Effectiveness: The prediction model should be able to handle the prediction problem accurately under the high noise with the limited amount of the observed data at the very early stage.
- Timeliness: The computational complexity of the prediction model should be controlled within an acceptable threshold to adapt to the requirement of huge and complex networks.

In this paper, we propose an efficient and effective early stage event prediction model for online social networks to satisfy the above two properties. We formulate the event prediction problem as a classification problem and build a Bayesian perspective Early stage Event Prediction model, in abbreviation, BEEP. BEEP is a Semi-Naive Bayes Classifier based model, which integrates 3 temporal features and 2 structural features and models their dependencies. Semi-Naive Bayes Classifier is intrinsically suitable to handle the prediction problem in online social networks. By modeling the features into random variables, we can successfully handle the high noise in social networks. Besides, as Semi-Naive Bayes Classifier is a generative model with simple network structure, its training and prediction process can be relatively fast. Most importantly, having combined the above 5 features, our model can successfully output the prediction result effectively and timely.

Our main contribution is to establish a powerful event prediction model for online social networks to handle the high noise and the data deficiency problem at the early stage of an event. It shows how to handle the feature selection, modeling and prediction even when the data is

This work has been supported in part by the China 973 Project (2014CB340303), Program of International S&T Cooperation (2016YFE0100300), China NSF Projects (Nos. 61672353, 61472252), and CCF-Tencent Open Research Fund.

*Xiaofeng Gao is the corresponding author

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.124

extremely deficient at the very early stage. We conducted extensive experiments to demonstrate the effectiveness of our model.

II. Related Works

Through the years, social networks have always attracted the scholars' attention, especially the prediction of the social network contents, including predicting the popularity of a single tweet, of an event and of a topic.

One of the pioneers is Suh et al [5]. They jointly considered the content and user profile of a microblog, then designed a model based on PCA and Generalized Linear Model (GLM) to find what may influence the repost of a microblog. Later scholars also adopted the method of combining feature and general machine learning techniques to handle the problem of prediction in social networks.

Later, various feature-based methods have been developed. Researchers have shown that temporal feature based methods actually have the best performance [16], [17], [13]. Szabo et al. [18] also suggested that temporal features alone can reliably predict the future popularity. [19] used the time between the ith repost and the $(i-1)$th repost to support prediction; sum of the frequencies of the first 9 time window(8 hours for each) a topic is discussed by users, average rate of change and standard deviation were used in [20] to predict whether and when a topic will become hot in social networks.

Another important factor for information dissemination in social networks is the structure of the network, in other words, the structure of the friendship network. Meeyoung Cha et al. [11] indicated that whether a user will be influenced depends more on the diversity of his friendship network. More specifically, the higher number of the connected components of a user's friendship network will increase the possibility a user will be influenced. Although evidences were given [12], [13] that structural features are not as effective as temporal features, structural features are still widely used in prediction tasks. [14] made use of the reciprocity of the network, number of the connected components, maximum size of the connected component, average authority of authors and link density in the network to establish their prediction model for tweet popularity. [15] used the number of edges from early adopters to the entire graph as a strong indicator of the dissemination possibility at the early stage.

The majority of the previous works aim to predict the popularity from a fixed corpus during a long time period, but they are unable to handle the timeliness of predicting the popularity of the emerging events. Qi Dang et al. [21] proposed a Dynamic Bayesian Network based model to detect events at the very early stage in microblogging networks. In their model, structural features and user features are taken into consideration. They modeled them into a Dynamic Bayesian Network to predict when the event will emerge. However, since they discretize the variables and ignored the temporal features, which could be the most critical feature in the prediction problem, the performance of their model is doubtable.

In order to predict the whether an observed event will become popular at the very early stage, we jointly consider the diverse features and the timeliness requirement of the task and proposed our prediction model, which will be introduced in the following parts of this paper.

III. BEEP: Bayesian perspective Early statge Event Prediction model

In this section, we present our Bayesian Network based event prediction model.

A. BEEP: Network Structures

We first present the network structure of our prediction model, BEEP (Bayesian network based Early stage Event Prediction model).

BEEP is formulated as a Semi-Naive Bayes Classifier [22], which is a kind of Bayesian Network Classifier as well as an improved Naive Bayes Classifier with a looser constriant. In Semi-Naive Bayes Classifier, the conditionally independent assumption of the variables is loosened: there can be edges, in other words, the dependencies, among some of the variables.

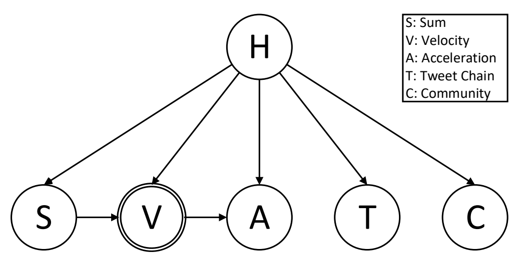


Fig. 1: Network Structure of BEEP

Fig. 1 illustrated the network structure of BEEP. In our model, five features are used.

- Sum: the sum of the discuss frequencies of the event in the observed 8 time windows. We can write it down as $S_i = \sum_{j=1}^{8} f_i^{(j)}$, where $f_i^{(j)}$ is the frequency on which event E_i was being discussed in time window j.
- Velocity: the average rate of change of the sum. Here the velocity have two attributes: the first one is the velocity of the first 4 time windows, and the second one is the velocity of the following 4 time windows, i.e. $V_i = \left(\frac{\sum_{j=1}^{4} f_i^{(j)}}{4}, \frac{\sum_{j=5}^{8} f_i^{(j)}}{4} \right)$.
- Acceleration: the average rage of change of the velocity, namely $A_i = \frac{V_i^{(2)} - V_i^{(1)}}{4}$, where $V_i^{(j)}$ denote the j^{th} attibute of V_i.
- Tweet Chain: this feature is the maximum length of the retweet chain of all the tweets related with this

event. It can approximately represent the existence of the influential user which can significantly enhance the dissemination of the information. We can extract this feature by monitoring the retweet status of the network.

- Community: the number of the communities in the friendship network. Friendship network is built from the follow relationships among the users, and in the friendship network of an emerging event, the number of the communities will greatly enrich the way the information spread. Here, we refer to Louvain Method [23] to extract this feature.

Here, in Fig. 1, node S is for the Sum feature, and similarly, V, A, T and C are for the Velocity, Acceleration, Tweet Chain and Community features. S, T and C is conditionally independent with each other, while A is also dependent on V. This is an intuitive dependency since the acceleration is determined by the velocity, and this is what makes our model the Semi-Naive Bayes Classifier. The reason for introducing this dependency is that even with the same acceleration, the difference of the velocity could lead to different possibility. For example, two event may have the same acceleration, but the one with a higher velocity may have a higher possibility to become a hot event in the future.

After observing the distribution of the above five features, we choose Beta Distribution, Gamma Distribution, Gaussian Distribution, Gamma Distribution and Gamma Distribution to model them accordingly.

Here, the double lined circle, V is called the composite node. The composite node is the composition of a set of conventional nodes with similar distributions. The purpose of introducing composite nodes is to reduce the learning time and spatial complexity of the model [24], [25]. Hence, we actually have $V = [V_1, V_2]$.

The network is very similar to conventional Semi-Naive Bayes Classifier, except that BEEP network contains one composite node. The joint distribution of the network is given by

$$\begin{aligned}&P(H, S, V, A, T, C)\\&= P(H)P(S|H)P(V|H, S)P(A|H, V)P(T|H)P(C|H)\end{aligned} \tag{1}$$

Besides, although temporal features, especially the Sum feature is a strong evidence for the hot event prediction problem, it can be still useless sometimes since in the very early stage of an event, the Sum feature of the hot event and the non-hot event is not that distinguishing. We also developed another simplified version of BEEP, namely SimBEEP that discarts the Sum feature. In the experiment part, we will show that under some special circumstances, SimBEEP can also produce a satisfiable result.

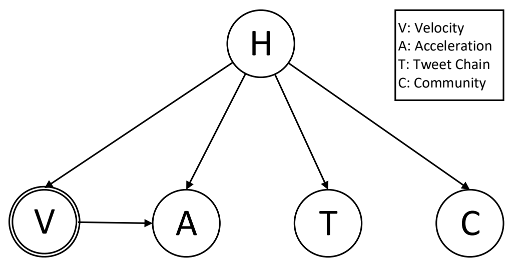


Fig. 2: Network Structure of SimBEEP

B. BEEP: Model Learning

We then go into the details of the learning algorithm for BEEP. As a full generative model, the learning process of BEEP aims to maximize the likelihood of the joint distribution of the network. More specifically, we aim to maximize Equation 1. To do this, given the observations of the past result, including the temporal and feature evolution of the observed events, we aim to maximize the likelihood of their joint distribution.

First, we set the parameters for the network. As mentioned in the previous section, we assume the features S, V_1, V_2, A, T, C follow Beta Distribution, Gamma Distribution, Gamma Distribution, Gaussian Distribution, Gamma Distribution and Gamma Distribution accordingly. Next we assume that H_1 denotes the event is hot, and H_0 denotes the event is not hot, and we can further have the following distributions: $p(S|H_1) = Beta(a_s^1, b_s^1)$, $p(V_1|H_1) = Gamma(c_{v_1}^1, d_{v_1}^1)$, $p(V_2|H_1) = Gamma(c_{v_2}^1, d_{v_2}^1)$, $p(A|H_1) = \mathcal{N}(\mu_a^1, \sigma_a^1)$, $p(T|H_1) = Gamma(c_t^1, d_t^1)$ and $p(C|H_1) = Gamma(c_c^1, d_c^1)$ for H_1; $p(S|H_0) = Beta(a_s^0, b_s^0)$, $p(V_1|H_0) = Gamma(c_{v_1}^0, d_{v_1}^0)$, $p(V_2|H_0) = Gamma(c_{v_2}^0, d_{v_2}^0)$, $p(A|H_0) = \mathcal{N}(\mu_a^0, \sigma_a^0)$, $p(T|H_0) = Gamma(c_t^0, d_t^0)$ and $p(C|H_0) = Gamma(c_c^0, d_c^0)$ for H_0.

In addition, to calculate those conditional probabilities, we apply the Bayesian Rules. Take Take $p(A|V, H_1)$ as an example.

$$p(A|V, H_1) = \frac{p(V_1, V_2, A|H)}{p(V_1, V_2|H_1)} \tag{2}$$

Since V_1 and V_2 are conditionally independent with each other, we further have

$$\begin{aligned}&p(A|V, H_1)\\&\quad= \frac{p(V_1, V_2, A|H)}{p(V_1, V_2|H_1)}\\&\quad= \frac{p(V_1, V_2, A|H_1)}{p(V_1|H_1)p(V_2|H_1)}\end{aligned} \tag{3}$$

To calculate $p(V|S, H_1)$ and $p(A|V, H_1)$, we have to get $p(S, V_1, V_2|H_1)$ and $p(V_1, V_2, A|H_1)$. These multivariate distributions captures the inner relationships among variables. To do this, we make following assumptions.

For $p(S, V_1, V_2|H_1)$, we generalize V_1 and V_2 and model it using a Dirichlet Distribution. More specifically, let $V_1' = \frac{T}{T+1}V_1$, $V_2' = \frac{T}{T+1}V_2$ and $S' = \frac{T}{T+1}S$, where T is the observation time window. By doing so, we may have $S' + V_1' + V_2' \in [0,1]$ which satisfies the constriant of the Dirichlet Distribution. Let $\Delta = [S', V_1', V_2']$, we have $\Delta \sim Dir(\delta)$.

For $p(V_1, V_2, A|H_1)$, we assume that $p(V_1, V_2, A|H_1)$ will follow a Multivariate Gaussian Distribution. It is actually an approximation to the combinations of Gamma Distributions and Gaussian Distributions. More Specifically, Let $Z = [V_1, V_2, A]$, we have $Z \sim \mathcal{N}(\mu_z^1, \Sigma_z^1)$, where μ is the mean vector and $\mathbf{\Sigma}$ is the covariance matrix of this Multivariate Gaussian Distribution Z.

Now, for a feature vector $X_i = \langle S^i, V^i, A^i, T^i, C^i \rangle$ from class H_1, we have

$$\begin{aligned}
&p(X_i, H_1) \\
&= p(H_1)p(S^i, V^i|H_1)\frac{p(A^i, V^i|H_1)}{p(V^i|H_1)}p(T^i|H_1)p(C^i|H_1) \\
&= \beta Dir(S^i, V_1^i, V_2^i|\delta^1)\mathcal{N}(A^i, V_1^i, V_2^i|\mu_z^1, \Sigma_z^1) \\
&\quad /(Gamma(V_1^i|c_{v_1}^1, d_{v_1}^1)Gamma(V_2^i|c_{v_2}^1, d_{v_1}^1)) \\
&\quad \cdot Gamma(T^i|c_t^1, d_t^1)Gamma(C^i|c_c^1, d_c^1) \qquad (4)
\end{aligned}$$

For class H_0, it is similar, so we will not list it in detail.

To keep the notation clean, let $\omega_1 = \langle \delta^1, \mu_z^1, \Sigma_z^1, c_{v_1}^1, d_{v_1}^1, c_{v_2}^1, d_{v_2}^1, c_t^1, d_t^1, c_c^1, d_c^1 \rangle$ and $\omega_0 = \langle \delta^0, \mu_z^0, \Sigma_z^0, c_{v_1}^0, d_{v_1}^0, c_{v_2}^0, d_{v_2}^0, c_t^0, d_t^0, c_c^0, d_c^0 \rangle$ and

$$t_i = \begin{cases} 1, & \text{if } E_i \in \mathbf{E}_h \\ 0, & \text{if } E_i \in \mathbf{E}_n \end{cases} \qquad (5)$$

where $\mathbf{E}_h$ denotes the hot event set and $\mathbf{E}_n$ denote the non-hot event set. Now we can train the parameters on training set D where $|D| = k$ with the likelihood function

$$\begin{aligned}
&p(D|\beta, \omega_0, \omega_1) \\
&= \prod_{i=1}^{k}[\beta p(X_i, H_1)]^{t_i}[(1-\beta)p(X_i, H_0)]^{(1-t_i)} \qquad (6)
\end{aligned}$$

The goal of the parameter training is to get the parameters $\langle \beta, \omega_0, \omega_1 \rangle$ by solving the following problem

$$\max \log p(D|\beta, \omega_0, \omega_1) \qquad (7)$$

More specifically, since the parameters of the log-likelihood of Tweet Chain and Community features are obviourly independent with others, we can simply maximize them to maximize our objective function Equation 7. In other words, we simply fit them into random variables by solving the following two problems:

$$\max \log \prod_{i=1}^{k} Gamma(T^i|c_t^1, d_t^1)^{t_i} Gamma(T^i|c_t^0, d_t^0)^{1-t_i} \qquad (8)$$

$$\max \log \prod_{i=1}^{k} Gamma(C^i|c_c^1, d_c^1)^{t_i} Gamma(C^i|c_c^0, d_c^0)^{1-t_i} \qquad (9)$$

As for the rest part of the objective function, we observe that they all involve variable V_1 and V_2. In previous sections, we assume $p(V_1|H_1) \sim Gamma(c_{v_1}^1, d_{v_1}^1)$ and $p(V_1|H_0) \sim Gamma(c_{v_1}^0, d_{v_1}^0)$. Let $e_{v_1}^1 = \mathbb{E}(V_1|H_1)$ and $e_{v_1}^0 = \mathbb{E}(V_1|H_0)$, therefore we can further have $e_{v_1}^1 = \frac{c_{v_1}}{d_{v_1}}$ and $e_{v_1}^0 = \frac{c_{v_0}}{d_{v_0}}$ according to the property of Gamma Distribution. Similarly, we can get $e_{v_2}^1$ and $e_{v_2}^0$. Here, we have assumed $\Delta \sim Dir(\delta) = Dir(\delta_s, \delta_{v_1}, \delta_{v_2})$, then we have

$$\begin{cases} (e_{v_1}^1 - 1)\delta_{v_1} + e_{v_1}^1\delta_{v_2} + e_{v_1}^1\delta_s^1 = 0 \\ (e_{v_2}^1 - 1)\delta_{v_2} + e_{v_2}^1\delta_{v_1} + e_{v_2}^1\delta_s^1 = 0 \end{cases} \qquad (10)$$

As for the Multivariate Gaussian Distribution, denoting $\mu_z^1 = [\mu_{v_1}^1, \mu_{v_2}^1, \mu_a^1]$, we can easily have

$$\mu_{v_1}^1 = e_{v_1}^1 \text{ and } \mu_{v_2}^1 = e_{v_2}^1 \qquad (11)$$

To keep the notation clean, let $\omega_1' = \langle \delta^1, \mu_z^1, \Sigma_z^1, c_{v_1}^1, d_{v_1}^1, c_{v_2}^1, d_{v_2}^1 \rangle$ and $\omega_0' = \langle \delta^0, \mu_z^0, \Sigma_z^0, c_{v_1}^0, d_{v_1}^0, c_{v_2}^0, d_{v_2}^0 \rangle$. Given the relationships shown by Equation 10 and Equation 11, the rest of the parameter learning problem can be shown as Equation 12.

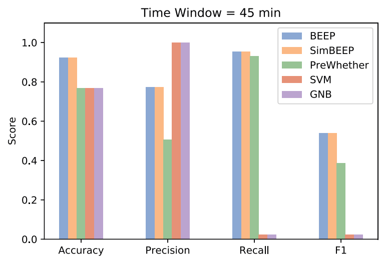
$$\begin{aligned}
&Q(\omega_0', \omega_1') \\
&= \arg\max \log(\beta \cdot (1-\beta)) \\
&+ \sum_{i=1}^{k} t_i[\log Dir(S^i, V_1^i, V_2^i|\delta^1) + \log\mathcal{N}(A^i, V_1^i, V_2^i|\mu_z^1, \Sigma_z^1) \\
&- \log Gamma(V_1^i|c_{v_1}^1, d_{v_1}^1) - \log Gamma(V_2^i|c_{v_2}^1, d_{v_1}^1)] \\
&+ \sum_{i=1}^{k}(1-t_i)[\log Dir(S^i, V_1^i, V_2^i|\delta^0) \\
&+ \log\mathcal{N}(A^i, V_1^i, V_2^i|\mu_z^0, \Sigma_z^0) - \log Gamma(V_1^i|c_{v_1}^0, d_{v_1}^0) \\
&- \log Gamma(V_2^i|c_{v_2}^0, d_{v_1}^0)] \qquad (12)
\end{aligned}$$

At last, we use the Coordinate Ascent Algorithm to learn the parameters.

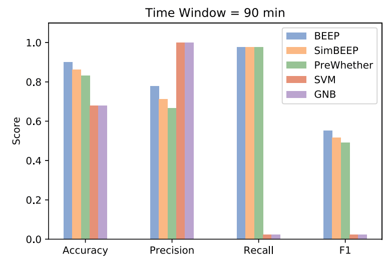
In addition, β in the model can be calculated by $\beta = \frac{|\mathbf{E}_h^D|}{|\mathbf{E}_h^D| + |\mathbf{E}_n^D|}$. Besides, for SimBEEP, the learning process is very similar, so we will not go into the details of it.

IV. Experiments

In this section, we will present our extensive experiment results over the performance of our model, BEEP and SimBEEP, against some conventional prediction models and some state-of-art models.



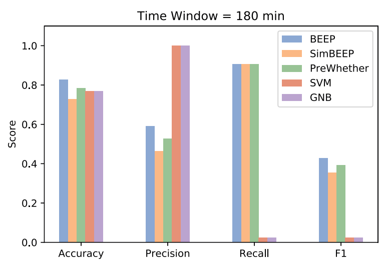
(a) Time Window = 45 min



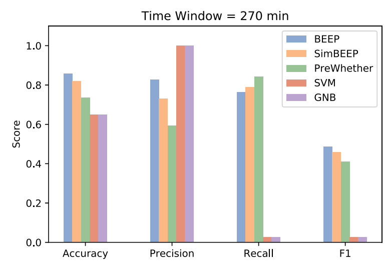
(b) Time Window = 90 min

(c) Time Window = 135 min

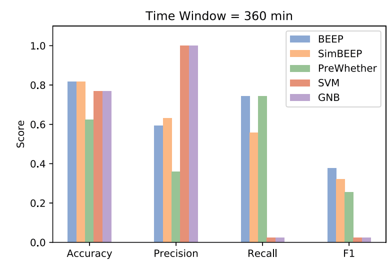
Fig. 3: Experiments Result with Different Time Windows Part 1



(a) Time Window = 180 min



(b) Time Window = 270 min



(c) Time Window = 360 min

Fig. 4: Experiments Result with Different Time Windows Part 2

A. Dataset

The experiment dataset we use is a tweet dataset we crawled using Tweepy from 23rd, July, 2015 to 2nd, Dec, 2016. The dataset consists of 455253 tweets. Our dataset consists of two subset, status and retweet. status recorded the information of the tweet alone, and retweet recorded the retweet relationships of the tweets. For every item in the status dataset, there are following 5 attributes:

- status_id: The identifier of the tweet
- created_at: The post time of the tweet
- text: The content of the tweet
- catch_time: The time our crawler caught the tweet
- user_id: The ID of the user who first posted the tweet

As for the retweet dataset, there are following attributes:

- original_id: The ID of the original tweet in the dataset
- retweet_id: The ID of the retweet status
- retweet_user_id: The ID of the user who posted this retweet

For simplicity, we extract events based on hash-tags, and classify the tweets into different events. We then set threshold for the events, regarding events with a tweet count larger than 800 as hot events, and filtered the events with a tweet count smaller than 100. After these preprocessing to the data, we can get our refined dataset of 107 hot events and 608 non-hot events.

B. Baseline Methods

Many existing methods in time series modeling, recommendation systems and classification can be used to handle the problem of event prediction in online social networks. We compare our method with the following models of predicting whether an event will become hot at the very early stage.

- Gaussian Naive Bayes: Gaussian Naive Bayes Classifier is a well-known classifier which is a special kind of Naive Bayes. It assumes the variables is conditionally independent with each other and all follow the Gaussian Distribution. Gaussian Naive Bayes has many applications, the most famous one of which is the Naive Bayes Spam Filtering.
- Support Vector Machine: SVM is a famous supervised learning model invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963. Through the years it has been well developed and has a large number of applications, such as text and hypertext categorization and image classification.
- PreWhether: PreWhether [20] is a model designed by Weiwei Liu et al. in 2015. It is also an probabilistic model to predict whether a topic would become hot in the future. It uses three features within 9 time windows(each equals to 8 hours) of a topic: sum, average speed of change and standard error, and model them using Beta Distribution, Gaussian Distribution and Gamma Distribution. Then it calculate

the probability of the topic is hot and the topic is not hot given the feature vector, and output the result based on the comparison.

C. Experiment Results

In this subsection, we will introduce the experiment results of our model. More specifically, we compared the prediction performance of different models, including BEEP, SimBEEP, PreWhether, SVM and GNB. The metrics we use include Accuracy, Precision, Recall and F1 Score. The reason we choose these metrics is that the number of the non-hot events is substantially larger than the number of the hot events. Therefore, even if the model predict all test cases to be non-hot, it could still gain a high accuracy.

In our experiment, we tested the performance of the models with different time window length, in other words, with different observation time. The results can be shown as follows.

From Fig. 3 and Fig. 4, we can draw following conslusions:

- The accuracy is not reliable. For any model, the accuracy is very similar, since they have all produced enough negative predictions which make up for a great part of the test set.
- The performance of SVM and GNB is very poor. Their precision is extremely high, both 100%, but recall is very low, which is because they have produced only a few number of positive predictions and they are all true positives. A better way to measure the performance is to refer to the F1 Score.
- The performances BEEP and SimBEEP are better than any other baseline methods. PreWhether can also present an acceptable prediction performance, but still worse than BEEP and SimBEEP.
- BEEP and SimBEEP act much better at the early stage prediction problem. In Fig. 3, we can see that the F1 Score of BEEP and SimBEEP is much higher than any other methods. When time window equals to 45 minutes, compared with PreWhether, BEEP and SimBEEP gained an improvement of 42.5%.
- SimBEEP acts similarly to BEEP when at the very early stage. In Fig. 3a, with the time window equals to 45 minutes, the performance of BEEP and SimBEEP is very close and in Fig. 3c, the performance of SimBEEP is even better than BEEP. This is because the temporal feature for the hot events and the non-hot events is not that distinctive.

V. Conclusion

Predicting hot events in online social networks at an early stage is very meaningful for marketing, advertisement and recommendation systems.

We first introduce the five features we used for the model, including sum, velocity, acceleration, tweet chain and community, study their distributions and model them using Beta Distribution, Gaussian Distribution and Gamma Distribution. After that, combining them together, we propose our model, BEEP and SimBEEP and present the corresponding learning algorithm.

Substantial experiments on real dataset have proven that our model, BEEP and SimBEEP outperform the baseline methods and make much better predictions.

References

[1] F. Giummolè, S. Orlando, and G. Tolomei, "Trending topics on twitter improve the prediction of google hot queries," in SocialCom.

[2] S. Wasserman and K. Faust, Social network analysis: Methods and applications. Cambridge University Press, 1994.

[3] R. A. Hanneman and M. Riddle, "Introduction to social network methods," 2005.

[4] A. Galeotti, S. Goyal, and J. Kamphorst, "Network formation with heterogeneous players," Games and Economic Behavior, 2006.

[5] B. Suh, L. Hong, P. Pirolli, and E. H. Chi, "Want to be retweeted? large scale analytics on factors impacting retweet in twitter network," in SocialCom, 2010.

[6] C. Li, A. Sun, and A. Datta, "Twevent: Segment-based event detection from tweets," in CIKM, 2012.

[7] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi, "Bad news travel fast: a content-based analysis of interestingness on twitter," in WebSci, 2011.

[8] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," JASIST, 2011.

[9] J. Yang and S. Counts, "Predicting the speed, scale, and range of information diffusion in twitter," in ICWSM, 2010.

[10] O. Tsur and A. Rappoport, "What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities," in WSDM, 2012.

[11] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy," in ICWSM, 2010.

[12] S. Gao, J. Ma, and Z. Chen, "Effective and effortless features for popularity prediction in microblogging network," in WWW, 2014.

[13] J. Cheng, L. A. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, "Can cascades be predicted?" in WWW, 2014.

[14] X. Zhang, Z. Li, W. Chao, and J. Xia, "Popularity prediction of burst event in microblogging," in WAIM, 2014.

[15] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on twitter," in WSDM, 2011.

[16] S. Kong, Q. Mei, L. Feng, F. Ye, and Z. Zhao, "Predicting bursts and popularity of hashtags in real-time," in SIGIR, 2014.

[17] B. Shulman, A. Sharma, and D. Cosley, "Predictability of popularity: Gaps between prediction and understanding," in ICWSM, 2016.

[18] G. Szabó and B. A. Huberman, "Predicting the popularity of online content," Communications of the ACM, 2010.

[19] L. Hong, O. Dan, and B. D. Davison, "Predicting popular messages in twitter," in WWW, 2011.

[20] W. Liu, Z. Deng, X. Gong, F. Jiang, and I. W. Tsang, "Effectively predicting whether and when a topic will become prevalent in a social network," in AAAI, 2015.

[21] Q. Dang, F. Gao, and Y. Zhou, "Early detection method for emerging topics based on dynamic bayesian networks in microblogging networks," ESWA, 2016.

[22] F. Zheng and G. I. Webb, Semi-Naive Bayesian Learning, 2010.

[23] V. A. Traag, "Faster unfolding of communities: Speeding up the louvain algorithm," CoRR, 2015.

[24] L. Jiang, H. Zhang, and Z. Cai, "A novel bayes model: Hidden naive bayes," TKDE, 2009.

[25] A. Fernández, M. Morales, C. Rodríguez, and A. Salmerón, "A system for relevance analysis of performance indicators in higher education using bayesian networks," KAIS, 2011.