

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/328439638>

An Effective Approach for Modelling Time Features for Classifying Bursty Topics on Twitter

Conference Paper · October 2018

DOI: 10.1145/3269206.3269253

CITATIONS

0

READS

25

6 authors, including:



[Xiaoyu Xiong](#)

University of Exeter

4 PUBLICATIONS 10 CITATIONS

SEE PROFILE

An Effective Approach for Modelling Time Features for Classifying Bursty Topics on Twitter

Anjie Fang
University of Glasgow, UK
a.fang.1@research.gla.ac.uk

Iadh Ounis
University of Glasgow, UK
iadh.ounis@glasgow.ac.uk

Craig MacDonald
University of Glasgow, UK
craig.macdonald@glasgow.ac.uk

Philip Habel
University of South Alabama, USA
habel@southalabama.edu

Xiaoyu Xiong
University of Exeter, UK
x.xiong@exeter.ac.uk

Hai-Tao Yu
University of Tsukuba, Japan
yuhaitao@slis.tsukuba.ac.jp

ABSTRACT

Several previous approaches attempted to predict bursty topics on Twitter. Such approaches have usually reported that the time information (e.g. the topic popularity over time) of hashtag topics contribute the most to the prediction of bursty topics. In this paper, we propose a novel approach to use time features to predict bursty topics on Twitter. We model the popularity of topics as density curves described by the density function of a beta distribution with different parameters. We then propose various approaches to predict/classify the bursty topics by estimating the parameters of topics, using estimators such as Gradient Decent or Likelihood Maximization. In our experiments, we show that the estimated parameters of topics have a positive effect on classifying bursty topics. In particular, our estimators when combined together improve the bursty topic classification by 6.9% in terms of micro F1 compared to a baseline classifier using hashtag content features.

ACM Reference Format:

Anjie Fang, Iadh Ounis, Craig MacDonald, Philip Habel, Xiaoyu Xiong, and Hai-Tao Yu. 2018. An Effective Approach for Modelling Time Features, for Classifying Bursty Topics on Twitter. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, October 22–26, 2018, Torino, Italy. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3269206.3269253>

1 INTRODUCTION

Scholars have used hashtags to study and analyse topics of conversations on Twitter. On Twitter, hashtags are commonly used to group tweets around a topic or theme. For instance, Tsur et al. [11] showed the properties of hashtags and how hashtags spread; Yang et al. [15] and Xu et al. [14] studied how the popularity of a topic can grow and fade over time. Predicting the future trend of a hashtag has been shown to be helpful for a variety of applications. For example, news media outlets may wish to capture potentially trending hashtag topics, while public relation firms can benefit from knowing the directions of policy-related conversations [1].

Previous work has studied the effectiveness of features for classifying whether a hashtag topic will burst at a future time. Such features may include the time series of a topic [7, 12] (i.e. the frequency

of tweets during time intervals), the properties of a hashtag [9], or the statistics of the involved users or tweets [11]. As reported in [6], the time dimension (i.e. timestamps) of tweets in the topic is the most important feature when predicting/classifying bursty topics.

In this paper, we explore a more effective method for leveraging time features. The beta distribution is reported to effectively model time series data [2, 5, 12]. Hence, we propose to use the beta distribution to model the popularity curve of a hashtag topic over time. To achieve this, we use a continuous density curve represented by the probability density function of the beta distribution with two parameters α and β , which we call the popularity density function (PDF) of a topic popularity curve. We argue that the current and future popularity of a topic have a connection and that such a connection can be reflected by the PDF of its popularity curve with its corresponding parameters. Therefore, we can predict the future popularity/burstiness of a hashtag topic by estimating the parameters of its PDF curve.

This paper addresses two research questions: 1) **How to estimate the parameters of a topic's PDF** using the current timestamps (i.e. the observed timestamps thus far) of tweets associated to the topic. 2) How to use the estimated parameters to predict the burstiness of a topic. For the first research question, we apply three state-of-the-art parameter estimators: Gradient Descend, Likelihood Maximisation and Markov Chain Monte Carlo-based estimators. In the estimation experiment, we show that these estimators can effectively compute the parameters for hashtag topics compared to a random estimator. For the second research question, when predicting/classifying the burstiness of a topic, we **use our estimators to predict the burstiness of a topic under two classification schemes** and compare their performances to the baseline classifiers using 19 hashtag content features. Our burst prediction experiments suggest that our estimators when combined together improve the topic burst classification by 6.9% in terms of micro F1 score compared to the best baseline topic burst classifier, Naive Bayesian classifier, in our dataset.

2 RELATED WORK

Many studies have aimed at predicting the future popularity of topics. A common approach in the literature is to classify whether a particular topic will burst in the future. To achieve this, binary classifiers (burst vs. non-burst) have been trained using content features from hashtags, users or tweets. These features can be the length of the hashtag [6, 7, 9, 11], the number of involved Twitter users/mentions [6–11], the user's followee/follower network [6–8, 10], or even the graph built from these given features [8–10]. Another important feature is time series. For instance, time-series vectors can be directly used as features for classification [6, 10].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '18, October 22–26, 2018, Torino, Italy

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-6014-2/18/10...\$15.00

<https://doi.org/10.1145/3269206.3269253>

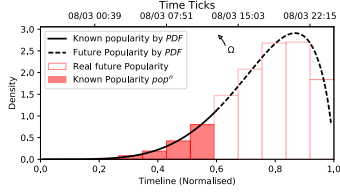


Figure 1: An example of topic popularity

The notion of time has also been widely used in other tasks for modelling the temporal trends of tweets [15] and topic modelling [12].

As reported by Kong et al. in [6], the time features have been shown to be the most influential when identifying bursty topics. However, there is little work that further examines the effectiveness of the time features for burst prediction. Hence, in this paper, we focus on how to more effectively model the time features and explore how well they can be used to predict the future burstiness of Twitter topics. To do so, as mentioned earlier, we use the beta distribution, as we find it to be effective and it was reported to best fit the time data in many prior work [5, 12].

In similar topic burst classification approaches, Fung et al. [4] used the binomial distribution to model the bursty features of documents. While we predict whether the topic will burst at a future time by estimating the parameters of a topic's PDF, Fung et al. aimed to detect and analyse the bursty events at the present time. Xu et al. [14] analysed the peak popularity of hashtag topics and predicted the peak time (different from burst) using a graph model. Similarly, Wu et al. [13] proposed a promising deep temporal context network to predict the exact popularity (e.g. the number of likes) of photos posted by users over time. The above approaches tackle a different task from ours. Hence, we do not use them as baselines, since they have not been designed for our task. Instead, we use the bursty topic classification in [6, 7] as baselines. In the following, we explore how to more effectively use time features for bursty topic classification.

3 DEFINITIONS

We define the popularity of a hashtag topic as **the frequency of tweets containing the hashtag during intervals of times**, denoted by $pop^N = \{\langle itv_1, c_1 \rangle, \dots, \langle itv_n, c_n \rangle, \dots, \langle itv_N, c_N \rangle\}$, where c_n is the number of tweets posted during time interval itv_n (e.g. a time period of 1 hour). This topic popularity representation is widely used in the literature [6, 9, 14, 15]. Note that other popularity representations have also been applied in the literature, e.g. the number of people seeing the tweets [3]. We aim to better use the popularities of the hashtag topics to enhance the prediction/classification of their popularities in the near future.

Following the existing literature [6, 7], we set a topic as **active** when $c_1 + c_2 + c_3$ is bigger than 50 (c_x is the count of tweets in the x -th time interval with 10 minutes as length). A topic starts to **burst** when c_n is bigger than $\max(c_1 + 50, 1.5 \times c_1)$. A topic will not burst (**non-burst**) if c_n never reaches the threshold. It is also worth mentioning that the burst of a topic is different from the peak of a topic, where the latter one indicates that the popularity reaches its highest value. We focus on the task of bursty topic classification, i.e. classifying whether a topic is going to burst on Twitter.

4 BURSTY TOPIC CLASSIFICATION

We first introduce how we model the topic popularity by using the popularity density function (PDF). Next, we explain our proposed parameter estimation and bursty topic classification approaches.

4.1 Modelling Topic Popularity using PDF

We use PDF - similar to the probability density - parametrised by α & β to model the future popularity trend of a hashtag topic. Several density functions could be used here. Indeed, we experimented with normal, logit-normal, gamma and beta distributions, and we find that the **beta distribution best fits the topic popularity** (i.e. it was the most effective). Hence, we will report all our results using the beta distribution as illustrated below:

$$PDF(t; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} t^{\alpha-1} (1-t)^{\beta-1} \quad (1)$$

where t is the normalised timestamp of a tweet and B is the beta function. For example, in Figure 1, the topic #InternationalWomensDay has 24 hours as the **time window** (i.e. normalised interval $[0,1]$) and the **present time** $\Omega = 0.6$ (normalised time, 12am). The real popularity over time can be obtained and shown as the solid histograms in Figure 1. We can **use the timestamps of tweets posted before 12am to estimate the parameters of the topic's PDF, which can be used to draw a density curve (solid and dashed lines in Figure 1) to present the topic popularity at the present and in the future.**

Our objective is to predict whether a topic will be bursty or not in the future. To do this, we first need to estimate the parameters of PDF for each topic. Next, we describe how to estimate the parameters (i.e. α & β) and then how to make the burstiness classification. We compare our approach against the existing bursty topic classification baselines from the literature.

4.2 Estimating Parameters of PDF

To estimate the parameters in a supervised manner, we transfer the **timestamps of a topic into labelled data**. Then, we experiment with 3 state-of-the-art parameter estimators using the labelled data:

Data transformation. We first use the present timestamps of hashtag topics to obtain the popularity of them over time intervals $pop^X = \{\langle itv_1, c_1 \rangle, \dots, \langle itv_x, c_x \rangle, \dots, \langle itv_X, c_X \rangle\}$, where x is the index of time interval and X is the total number. For each itv_x , we can have the fraction of tweets posted in itv_x :

$$y = c_x / \sum_{x'=1}^X c_{x'} \quad (2)$$

For each interval x , we can obtain its corresponding y using Equation (2), **which are the labelled data**. To estimate the parameters, the estimated y (y_{est}) values can be computed using the PDF:

$$y_{est} = \int^{itv^x} PDF(t; \alpha, \beta) dt / \int^{ITV^X} PDF(t; \alpha, \beta) dt \quad (3)$$

where $ITV^X = \{itv^1, \dots, itv^X\}$ is the present time interval and t is a time point over the interval. After the data transformation, the problem of parameter estimation for PDF is formulated as how to fit the parameters $\{\alpha, \beta\}$ to **make y_{est} as close to y given x .**

Estimation Approaches. There are a number of commonly used parameter estimation approaches in the literature, which can be easily applied using the transferred labelled data. In this paper, we choose three such approaches as well as a random estimator:

1) **GD-based Estimator** The Gradient Descent algorithm finds the local minimum cost by optimising the parameters (α, β) using a step proportional to the gradients corresponding to those parameters. The cost function (J) and gradients can be computed using the PDF:

$$J(\alpha, \beta) = \frac{1}{2X} \left\{ \sum_{x=1}^X (y_{est_x} - y_x)^2 + \lambda(\alpha^2 + \beta^2) \right\} \quad (4)$$

$$\frac{\partial}{\partial \alpha} J(\alpha, \beta) = \frac{1}{X} \left\{ \sum_{x=1}^X (y_{est_x} - y_x) \frac{\partial}{\partial \alpha} y_{est_x} + \lambda \alpha \right\} \quad (5)$$

where the regularising parameter λ is applied to avoid over-fitting. The gradient of β is computed similarly to Equation (5).

Table 1: 19 used features for all baseline classifiers.

Features	Description
1 - 3	Number of users / tweets / retweets
4 - 7	Number of tweets with mentions / URL / other hashtags/ symbols
8 - 12	Fraction of Feature 4 - 7
13 - 14	Sum / maximum number of followers of total users
15 - 17	Density / degree / size of the formed graph
18	Standard deviation of timestamps
19	Time series

2) **LM-based Estimator.** A Gaussian noise is applied to deal with the error between y and y_{est} , where ϵ is the mean (set to 0) and σ is the standard deviation:

$$y_x = y_{est_x} + \epsilon_x, \epsilon_x \sim N(0, \sigma^2) \quad (6)$$

Therefore, we maximise the likelihood $p(y|\alpha, \beta, \sigma)$:

$$p(y | \alpha, \beta, \sigma) = \sqrt{2\pi}\sigma \exp \left\{ -\frac{(y_x - y_{est_x})^2}{2\sigma^2} \right\} \quad (7)$$

3) **MCMC-based Estimator.** Similar to the LM estimator, we can design a fully probabilistic model to estimate the parameters. The difference is that all the parameters are sampled from a gamma prior (commonly used in Bayesian theory) as shown in Equation (8), where the gamma distribution has two parameters ω_1, ω_2 :

$$\epsilon \sim N(0, \sigma^2), (\alpha, \beta, \sigma) \sim \text{Gamma}(\omega_1, \omega_2) \quad (8)$$

We sample α, β following Metropolis-Hastings. The mean values of the sampled parameters are taken as the estimated parameters.

4) **Random Estimator.** We randomly generate timestamps for the missing tweets in the future time intervals using a uniform distribution. Therefore, the method of moments can directly estimate the parameters of PDF taking the completed timestamps. This random estimator is compared to the three state-of-the-art estimators¹.

4.3 Burst Prediction by PDF & its Parameters

To predict the future bursts of topics using PDF & its Parameters, we propose two classification schemes. Under these schemes, an estimator can predict the burst of a topic like a binary classifier.

• **The "Centrelines" (CNL) scheme.** If we use $x = 0.5$ (the centrelines) as a boundary line, the probability on the right of the central line ($x > 0.5$) is bigger than that on the left ($x < 0.5$) when its parameters $\alpha > \beta$ according to the beta distribution. Hence, we simply classify a topic as a bursty topic when its predicted parameter $\alpha > \beta$.

• **The "Parameters as Features" (PaF) scheme.** We directly use the α & β estimated by the three estimators as features to train a classifier. For both schemes, the estimators take the present timestamps in $[0, 0.5]$ as input to predict whether a topic will burst in the remaining time interval $[0.5, 1.0]$.

4.4 Bursty Topic Classification Baselines

Following [6, 7], we use a total of 19 content features (see Table 1) of hashtag topics to train the classifiers. Among them, the nodes and edges of the formed graph (Features 15 - 17) indicate users and mentions, respectively. The trained baseline classifiers are compared to the three estimators under the two described classification schemes.

5 EXPERIMENTS SETUP

Dataset. To answer our two research questions, we create two datasets. We first use the Twitter Streaming API (sample mode) to crawl a collection of tweet samples posted in March 2017, which allows us to collect 40 million tweets with at least one hashtag. According to the definitions of **active** and **burst** in Section 3, we

¹We simply use this random estimator for comparison when computing parameters as our main task is bursty topic classification.

Table 2: The RMSE errors of four parameter estimators.

Time Windows	(1) RSE	(2) GDE	(3) LME	(4) MCE
1h	0.076	0.034 ^{1,3,4}	0.050 ¹	0.052 ¹
5h	0.078	0.039 ^{1,3,4}	0.057 ¹	0.051 ¹
24h	0.079	0.048 ¹	0.051 ¹	0.052 ¹

obtain 2024 active topics/hashtags with different lengths of time windows. Two datasets are constructed from the 2024 topics:

• **Dataset 1: 600 active hashtag topics.** 600 hashtag topics are chosen from 2024 with $\{1, 5, 24\}$ hours as time windows (200 hashtags per setting). We use this dataset to check how well our proposed estimators are when computing parameters for topics.

• **Dataset 2: 363 bursting/non-bursting topics.** We select 177 bursting topics from the active topics that have time windows of length greater than 1 hour and at least 100 tweets². Together with the randomly selected 186 non-bursting topics, this forms the dataset for bursty topic classification.

Estimator Setup. We have three estimators: the GD-based estimator (GDE), the LM-based estimator (LME) and the MCMC-based estimator (MCE). The three estimators are compared to the random estimator (RSE) (c.f. Section 4.2). Each experiment is repeated 10 times for conducting significance tests. For LME, we set the σ to 2. This setting allows LME to deal with the errors between y and y_{est} . For MCE, the parameters of gamma prior are set to $\omega_1 = 1, \omega_2 = 0.1$, which can give a wide range of positive values for the parameters α & β . We set the present time as $\Omega = 0.5$. This means that the present time interval is always $[0, 0.5]$. An estimator is supposed to use the timestamps (normalised) in $[0, 0.5]$ to estimate the parameters and then predict the burstiness of topics in the future time interval $[0.5, 1]$ ³. We transfer the present timestamps into labelled data x & y , where the number of entries X is set to 100.

Bursty Topic Classifier Setup. Four classifiers are deployed as baseline classifiers: Naive Bayesian (NB), Support Vector Machine (SVM), Multilayer Perceptron (MLP) and C4.5 Decisions Tree (Tree) together with a random classifier (RD). These classifiers are trained using the feature listed in Section 4.3. They are compared to our proposed estimators under the two classification schemes: CNL and PaF. For the PaF scheme, we directly add the estimated parameters (6 in total) estimated by GDE, LME and MCE as features together with the 19 content features for the classifiers. We conduct a 10-fold cross validation for the burst classification.

6 RESULTS

We show the performance of our estimators when computing α & β , followed by the results of the burst classification experiments.

Results of Parameter Estimation. Table 2 shows the Root Mean Square Error (RMSE) scores of the four estimators with different time windows. A lower RMSE score means a better performance. The superscript (indices of the four estimators) indicates whether an estimator significantly (t-test, $p < 0.05$) performs better than another estimator, e.g. the GDE significantly outperforms the RSE estimator with 24h as a time window in terms of their RMSE scores. It is clear that all the three estimators (GDE, LME & MCE) perform significantly better than RSE. This suggests that the three estimators are indeed effective at predicting the correct parameters (α, β). Note that the RSE estimator performs reasonably well considering that the timestamps of a topic at the present time are

²This allows us to remove topics with little tweets, which users are unlikely to be interested in. Besides, it allows us to obtain enough context features listed in Table 1.

³In practice, the known time interval can be a few hours (e.g. 10 hours) and the objective is to predict the topic burstiness in the next few (e.g. 5) hours. Thus, the timestamps in the known time interval can be normalised using the full time interval (i.e. 15 hours). Accordingly, the present time Ω is 0.67. In this paper, we set Ω to 0.5. We will explore the impact of Ω for bursty topic classification in future work.

Table 3: Classification results.

models	Precision	Recall	micro F1
(a). Baseline classifiers (baselines)			
RD	0.489	0.489	0.49
NB	0.626	0.612	0.607
SVM	0.518	0.511	0.514
MLP	0.502	0.505	0.503
Tree	0.593	0.593	0.593
(b). Estimators under CNL scheme			
RSE	0.436	0.470	0.461
GDE	0.622	0.622	0.621
LME	0.656*	0.655*	0.656*
MCE	0.634*	0.633*	0.632*
(c). Estimators under PaF scheme			
Tree w/ α, β	0.615	0.616	616
MLP w/ α, β	0.641*	0.640*	0.640*
SVM w/ α, β only	0.657*	0.689^{†,*}	0.669*
MLP w/ α, β only	0.687^{†,*}	0.680 ^{†,*}	0.676^{†,*}

known. Besides, the GDE works better when the time windows are smaller, e.g. its *RMSE* scores are significantly smaller than the LME and MCE scores with 1h and 5h as the time windows. On the other hand, the LME and MCE estimators appear not to be sensitive to the time window sizes since only small performance differences are observed. The estimated density curve might not well fit the real topic popularity (the reason the *RMSE* scores can be high), however, it does not necessarily mean that these estimated parameters cannot be used in burst classification. Next, we report the performance of burst classification, which is our main task.

Results of Burst Classification. We list the Precision, Recall and micro F1 scores of the baseline classifiers and estimators under the CNL and PaF classification schemes in Tables 3 (a)-(c), respectively. Table 3 (a) lists the used baselines, including those approaches corresponding to Kong et al. [6, 7]. For reasons of brevity, we only list the best three classifiers (Tree, SVM & MLP) under the PaF classification scheme. The bold numbers in Tables 3 (a)-(c) indicate the best performance (selected using the mean values from 10 repeated experiments) in each sub-table. The superscript “*” in Table 3 (b) means a classifier significantly outperforms (McNemar’s test, $p < 0.05$) the NB baseline classifier, the best-performed classifier in Table 3 (a). Similarly, the superscript “†” in Table 3 (c) indicates that the classifier significantly outperforms the best performing classifier, LME, in Table 3 (b). We first note that compared to the classification performances in [6, 7] (F1 score around 35%), the baseline classifiers perform better on our dataset. In addition, among the 4 baseline classifiers in Table 3 (a) (except the random classifier, which performs worst), the NB classifiers have the best performance. Under the CNL classification scheme, the LME and MCE estimators perform better than the best baseline classifier while the GDE estimator has a similar performance compared to NB. We observe that the GDE estimator does not exhibit a better performance even though it can more accurately estimate the parameters for *PDF*. The reason might be that LME and MCE can better identify a bigger value between α and β under the CNL classification scheme. Under the PaF classification scheme, the MLP classifier (row “MLP W/ α, β ” in Table 3 (c)) is able to achieve a better performance than the NB baseline classifier when simply adding the parameters as features together with the other 19 features. However, when we only use the 6 estimated parameters (estimated by GDE, LME and MCE) as features, the MLP classifier (row “MLP W/ α, β only” in Table 3 (c)) improves the micro F1 by 6.9% compared to the best NB baseline classifier. This suggests that our proposed estimators indeed help the classifiers to more effectively predict the bursts of topics in our dataset. Furthermore, it verifies our hypothesis that the connection between

the present and future popularity of a hashtag topic can be reflected by its *PDF*. It also, to some extent, supports the conclusion that the three estimators can reasonably predict the parameters of a topic’s *PDF*. The reason why our estimators perform better than the baseline classifier might be that the time dimension is more useful than the other features. The direct use of time series, however, is not effective enough in the classification task. On the other hand, the manner in which we model timestamps, i.e. employing parametrised *PDF*, can better improve its effectiveness when predicting the burstiness of a topic. Our results suggest that our estimators can effectively predict the bursts of topics and their performances are comparable to the baseline classifiers. In addition, these proposed estimators can be readily applied, especially when there is little ground-truth data.

7 CONCLUSIONS

We proposed a novel approach to predict the bursts of hashtag topics on Twitter. To do this, we first modelled the popularity of hashtag topics as density curves represented by parametrised density functions. We then applied state-of-the-art parameter estimation approaches to predict the parameters of each topic density function. We then used two classification schemes to conduct the burst classification tasks using parameters computed by the estimators, which were compared to the 5 baseline classifiers using context features. Our bursty topic classification experiments show that our estimators predict whether a topic will burst more effectively than the baseline classifiers on our dataset. Moreover, when we use the estimated parameters as features in the classification, our estimators can improve the micro F1 score by 6.9% compared to the best baseline classifier. Overall, our work shows that the time features of hashtag topics can indeed improve the classification performance when modelled by the beta distribution.

REFERENCES

- [1] Julian Ausserhofer and Axel Maireder. 2013. Structures and topics of a networked public sphere. *Information, Communication & Society* 16, 3 (2013), 291–314.
- [2] Anjie Fang, Craig Macdonald, Iadh Ounis, Philip Habel, and Xiao Yang. 2017. Exploring Time-Sensitive Variational Bayesian Inference LDA for Social Media Data. In *Proc. of ECIR*.
- [3] António Fonseca and Jorge Louçã. 2017. Explaining the emergence of online popularity through a model of information diffusion. *Computational and Mathematical Organization Theory* (2017), 1–19.
- [4] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S Yu, and Hongjun Lu. 2005. Parameter free bursty events detection in text streams. In *Proc. of VLDB*.
- [5] Noriaki Kawamae. 2012. Theme chronicle model: chronicle consists of timestamp and topical words over each theme. In *Proc. of CIKM*.
- [6] Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye, and Zhe Zhao. 2014. Predicting bursts and popularity of hashtags in real-time. In *Proc. of SIGIR*.
- [7] Shoubin Kong, Qiaozhu Mei, Ling Feng, and Zhe Zhao. 2014. Real-time predicting bursting hashtags on Twitter. In *Proc. of WAIM*.
- [8] Zongyang Ma, Aixin Sun, and Gao Cong. 2012. Will this #hashtag be popular tomorrow?. In *Proc. of SIGIR*.
- [9] Zongyang Ma, Aixin Sun, and Gao Cong. 2013. On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology* 64, 7 (2013), 1399–1410.
- [10] Nargis Pervin, Tuan Quang Phan, Anindya Datta, Hideaki Takeda, and Fujio Toriumi. 2015. Hashtag popularity on Twitter: Analyzing co-occurrence of multiple hashtags. In *Proc. of SCSM*.
- [11] Oren Tsur and Ari Rappoport. 2012. What’s in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *Proc. of ICWSM*.
- [12] Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *Proc. of SIGKDD*.
- [13] Bo Wu, Wen-Huang Cheng, Yongdong Zhang, Qiushi Huang, Jintao Li, and Tao Mei. 2017. Sequential prediction of social media popularity with deep temporal context networks. In *Proc. of IJCAI*.
- [14] Wenwen Xu, Peng Shi, Jianyi Huang, and Feng Liu. 2017. Understanding and predicting the peak popularity of bursting hashtags. *Journal of Computational Science* (2017).
- [15] Jaewon Yang and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *Proc. of WSDM*.