Information Sciences 512 (2020) 315–326

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Predicting the active period of popularity evolution: A case study on Twitter hashtags

Jianyi Huang[a,b], Yuyuan Tang[a,b], Ying Hu[a,b], Jianjiang Li[a,b,*], Changjun Hu[a,b]

[a] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China
[b] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing, China

ARTICLE INFO

Article history:
Received 20 December 2017
Revised 12 April 2019
Accepted 17 April 2019
Available online 17 April 2019

Keywords:
Popularity evolution
Active period
Prediction
Twitter hashtags
DNN

ABSTRACT

The active period of popularity evolution indicates how long online content receives continuous attention from people. Although predicting popularity evolution has largely been explored, researches on predicting active period still remain open. If we know the duration of active period ahead of time, caching systems, online advertising, etc. can run more effectively. Therefore, predicting active period is of great importance, but it is a non-trivial task because of the two major challenges. First, numerous factors can influence the duration of active period. To predict active period accurately, it's difficult to consider what factors and how to embed them in DNN model. Second, the triggering time to predict different active periods must be decided carefully, because the durations of active periods differed from one another. This paper addresses these two challenges, focusing on Twitter hashtags as a case study. To deal with the first challenge, a DNN-based prediction framework is proposed, embedding dynamic and static factors by using LSTM and CNN respectively. To deal with the second challenge, an appropriate value of cumulative popularity is set to trigger predicting active period. Experimental and comparative results show the superiority of our prediction solution, comparing with spikeM and SVR.

1. Introduction

Enormous amounts of online content (videos, pictures, hashtags, etc.) are constantly being produced nowadays. There are around 400 h of video content uploaded to YouTube every minute and around 500 million tweets created on Twitter every day. Because of the unprecedented amount of information load, it is ever-more difficult for online content to compete for people's attention [35,36]. Hence, how much human attention aggregates on online content? Popularity evolution [2,6,10,11], a measurement for understanding human collective attention, reflects how much attention a piece of content receives over time.

Active period [16], a fundamental attribute of popularity evolution, reflects how long a piece of content can receive attention continuously. Due to the intense competition for human attention, most content dies an early death in terms of popularity, while only a few can receive attention continuously for a long time. Given a piece of content, the focus is how long its active period lasts. For instance, how long will a YouTube video keep receiving views? How long will a Twitter hashtag stay on the trending list? How long will a song stay popular?

* Corresponding author at: School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China.
E-mail address: lijianjiang@ustb.edu.cn (J. Li).

https://doi.org/10.1016/j.ins.2019.04.028
0020-0255/

From a scientific perspective, the active period of popularity evolution is at the heart of human collective behavior studies [9] and information diffusion studies [12,25]. Predicting active period helps in understanding how human attention aggregates on online content and how quickly a piece of content diffuses. From a practical perspective, active period is an important quantity for many online problems. Internet service providers can improve users' experience by smart caching according to the durations of active periods. Online advertisers can save their costs by deciding how long advertisements place on webpages. For singers and movie makers, long active periods of their works imply success and good income.

Given the importance and value of active period, predicting active period is not a trivial task due to the two challenges. First, various factors may have impacts on active period, including both dynamic and static factors. Dynamic factors include topological network [21,26], user information [5,17], etc. Static factors include online content [3,21], content age [5], etc. Considering this how can we propose a prediction solution dealing with so many factors, to make predictions as accurate as possible? Second, the durations of active periods vary from hours to weeks. Such a significant difference causes difficulties in the decision of when to trigger predicting active period. For an active period lasting for hours, triggering the prediction after days is too late. For an active period lasting for weeks, triggering the prediction after only hours cannot give us sufficient information. So how can we decide when to trigger predicting active period for each content, without knowing if it will last for hours or weeks?

This paper formulates the active period prediction as a regression problem, taking Twitter hashtags as a case study, because Twitter hashtags are typical examples of online content that can draw tons of users' attention. Given a hashtag and its historical information of popularity evolution, this paper predicts how long its active period will last. To address the first challenge, this paper proposes a DNN-based (Deep-Neural-Network-based) prediction framework, where several factors are taken into account. This paper presents how to vectorize these factors into two types of embeddings: LSTM-based (Long-Short-Term-Memory-based) dynamic embedding and CNN-based (Convolutional-Neural-Network-based) static embedding. Both dynamic embedding and static embedding are fed into a predictor of fully connected neural network together. To address the second challenge, prediction is triggered once cumulative popularity reaches a certain level (e.g. 1000), rather than after a fixed time for all hashtags.

This paper conducts the proposed prediction framework on a data set containing approximately 3 million Twitter hashtags and 40 million users. First of all, an overall prediction performance is evaluated for our solution. Second of all, this paper explores which factor has the most significant impact on active period. Finally, our solution is compared with baseline models. Experiments show that our solution outperforms baseline models. The contribution of this work is mainly three-fold.

(1) This paper brings forward a new and challenging prediction task in the field of popularity evolution: predicting active period.
(2) This paper proposes a DNN-based framework for this prediction task, where we present how to make embedding for both dynamic and static factors by using LSTM and CNN respectively. Our framework shows a good prediction performance.
(3) Among the factors explored in this paper, we find accumulation time is the most significant factor for predicting active period.

The rest of this paper proceeds as follows. Section 2 describes related work. Section 3 presents preliminaries. Section 4 introduces our solution for predicting active period. Experiments are conducted in Section 5 to evaluate our solution. Section 6 concludes this paper.

2. Related work

This section starts with reviewing studies on influencing factors of popularity evolution. Then, previous efforts on popularity evolution prediction are discussed.

2.1. Influencing factors of popularity evolution

It is not only the predicted popularity that matters, but also even more importantly, understanding of how individual parts influence the final popularity score [4]. Studies on influencing factors of popularity evolution are motivated by the question of what makes a few contents hugely popular, while others receive little attention. In one of the early studies of popularity evolution [6], the rich-get-richer principle is brought up to explain why a piece of content becomes popular at its early age and more and more popular later on. This raises a question that is what factor results in content getting popular at its early age? Borghol et al. [5] answered this question by doing correlation analysis and found that an uploader with a lot of social connections is an important factor contributing to initial popularity evolution. The above two studies were conducted on YouTube videos, but Bandari et al. [3] conducted their research on news articles. They considered four factors in an article, including news source, news category, subjectivity and named entities. They found that news source plays a more important role compared to other factors. There are also other research works on Twitter hashtags [1,21]. Ma et al. [21] took into account many factors that covered the spectrum of hashtags, including hashtag lexical feature, hashtag content feature, user count feature, network community feature, etc. They fed each of these features to classifiers in order to see impacts of these features. They observed that user count feature is the most important one.

Considerable recent work has involved influencing factors of popularity evolution as well. Besides factors like uploader feature and item content itself, there were works [11,19] focusing on referrers (links used by users to access the video). The referrer is the most important mechanism that drives users to access the content by statistical analyses. Cheng et al. [8] analyzed how sharer characteristics affect recurrence of popularity evolution. They quantified homophily or diversity of sharers by measuring the entropy of demographic characteristics distribution, observing that diversity encourages recurrence. Other than these previous studies using traditional methods, leading studies have started to use deep neural networks [7]. Zhang et al. [40] incorporated the factors including users' information, authors' information and similarity information between tweets and user interests.

The factors we choose to incorporate in this work are inspired by the above studies. However, the difference between our study and the above ones is that this paper distinguishes these factors into dynamic and static factors. Since popularity evolution is a dynamic process, we believe that learning its dynamic temporal behavior is critical. This paper utilizes LSTM to embed dynamic factors so that dynamic temporal behavior of popularity evolution can be learned.

2.2. Popularity evolution prediction

Popularity evolution prediction has been the target of recent studies. Given a piece of content, most studies have concentrated on predicting its popularity volume and forwarding possibility in future. These studies can be categorized into three types: statistics-based methods, model-based methods, and DNN-based methods.

A. Statistics-based methods

Most of the early studies of popularity evolution prediction fall under this category. To predict future popularity volume, Szabo and Huberman [32] presented a univariate regression model (SH model) based on the strong linear correlation between the logarithmically transformed popularity of YouTube videos at previous and future times. Pinto et al. [29] extended SH model to a multivariable regression model by taking into account popularity at historical time points and similarity features. Instead of predicting future popularity volume, Yu et al. [39] proposed the phase as a new representation for popularity evolution. They examined phase statistics with respect to content category and popularity, then discussed observations on how phases evolve over time. Xu et al. [37] performed a temporal analysis, investigating the time-evolving properties of the subgraphs formed by users discussing each hashtag based on a Twitter data set. Then they proposed a model to predict the peak time and peak volume of bursting hashtags.

B. Model-based methods

After using statistics-based methods at first, researchers have started to use model-based methods. He et al. [13] considered two types of sources in the comments of online information: timestamps for obtaining a temporal factor and usernames for mining potential social influence, in order to model comments as a time-aware bipartite graph to predict future popularity. To predict the final popularity volume of Twitter tweets, Zhao et al. [41] proposed a self-exciting point process model to capture the rich-get-richer phenomenon in popularity evolution. Rather than separately studying popularity evolution prediction on different social media platforms (Twitter and YouTube), Rizoiu et al. [30] supplied the missing link between exogenous inputs from Twitter and endogenous responses within YouTube. They developed HIP (Hawkes intensity process) model, which revealed an analytical relationship between endogenous and exogenous popularity factors. Lv et al. [20] extracted visual and social features of images, then fed a fusion of multi-feature to Linear Regression, Matrix Factorization based on Time and feature Cluster, and Support Vector Regression models respectively.

C. DNN-based methods

Recently, more and more researchers have utilized DNN to predict popularity evolution since the emergence of deep learning. In order to predict future popularity volume (or future size of cascades), Li et al. [18] learned the representation of individual cascade graphs in the context of global network structure and then combined the representation with biGRU (bidirectional gated recurrent unite) and attention mechanism. In order to predict whether a tweet will be retweeted by a user, Zhang et al. [40] developed an attention-based deep neural network to incorporate contextual and social factors for this task. Their experimental results showed that both contextual and social factors can significantly improve prediction performance. To predict the view count of the post, Hsu et al. [14] proposed a framework that integrates CNN-based multi-modal residual learning with random forest regression for social information.

In recent years, existing studies have focused on predicting future popularity volume or forwarding possibility in future from text-based, video-based and photo-based social media [23]. Most of them have employed a similar pipeline to compute popularity scores for different types of social media content. In contrast, this paper considers the problem from another angle: predicting active period. Active period is a fundamental attribute of popularity evolution and is valuable for both academic and industrial circles. To the best of our knowledge, this work is the first one discussing how to predict active period.

3. Preliminaries

This section gives definitions and introduces our data set.

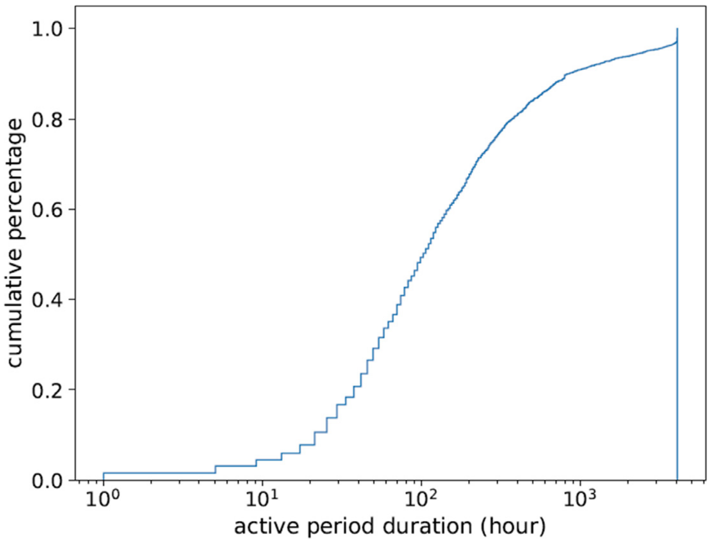

Fig. 1. The distribution of active periods. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

3.1. Definitions

Popularity. Given a piece of content z, the popularity y_t of z is defined as the amount of attention it receives at time $t \in \{1, 2, \ldots, L\}$, such as the number of users watching a video, or the number of users discussing a hashtag. For example, if the time unit is set to one hour, y_{10} will denote the popularity received during the tenth hour.

Popularity evolution. Given the observations of the popularity of z over its life span $L \in N^+$, the popularity evolution of z is given by the time series PE_z as follows:

$$PE_z = \{y_1, y_2, \ldots, y_L\} \quad (1)$$

Active period. Note that most contents undergo both active and inactive periods [16]. This paper uses the same method as in [16] to distinguish between active and inactive periods: a piece of content is inactive if it gains no popularity for 24 h. Some contents undergo multiple active periods. For example, the Olympic Games and the US elections are held once every four years. Popularity goes active for a while every four years. To simplify the problem, this paper studies the single active period during which the most popularity volume accumulates.

The goal of this paper is to predict the durations of active periods given the historical data of both static and dynamic factors.

3.2. Data set

Our primary data come from a portion of the tweet7 data set crawled by Yang and Leskovec [38]. The data set comprises 65 million tweets. We identify 3.3 million hashtags and 40 million users in these tweets. In our data set, most of the hashtags gain very small popularity, whereas only a few hashtags gain large popularity. Therefore, this paper selects the 30 thousand most popular hashtags (ranked by the peak value of popularity evolution).

The empirical cumulative distribution for the active periods of popularity evolution of these hashtags is shown in Fig. 1. This paper logarithmically rescales the horizontal axis in this figure due to the large variances present among active periods of different hashtags (notice that they range from one hour to several thousand hours). For each observed value on the blue line, the empirical cumulative distribution shows the fraction of hashtags for which the durations of active periods are shorter than this value. For more than 15% of hashtags, their durations of active periods are shorter than 24 h. For about 60% of hashtags, their durations of active periods are shorter than 100 h. For about 20% of hashtags, their durations of active periods are longer than a week.

4. Prediction solution

This section first discusses how to address the challenge of deciding the triggering time of prediction. Then, we present how to embed dynamic factors based on LSTM and how to embed static factors based on CNN. Finally, Dynamic and static factors are concatenated together and fed into a predictor of fully connected neural network, as shown in Fig. 2.

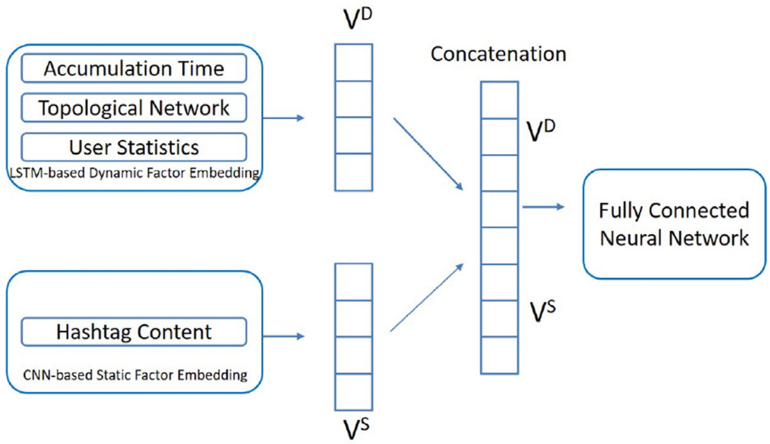

Fig. 2. The DNN-based framework.

4.1. Triggering time of prediction

According to Fig. 1, active periods vary significantly, ranging from one hour to thousands hours. This leads to one challenge for predicting active period. It is difficult to decide when to trigger predictions for different active periods, because triggering a prediction after a fixed time can either make a prediction suffer from insufficient historical information for long active periods or cause a late prediction for short active periods. Considering this problem, how can we decide when to trigger a prediction without knowing if an active period will last for hours or weeks.

This paper comes up with the idea that predictions are triggered once cumulative popularity reaches a certain level (e.g. 1000), ensuring historical information can be obtained. If this level is set larger, we can get more historical information. Furthermore, triggering predictions in this way, doesn't cause a late prediction, because popularity always reaches a certain level in an active period.

4.2. Dynamic factors embedding

For dynamic factors, this paper considers accumulation time, user statistics, and topological network formed by the users discussing a hashtag. These three factors change over time. To capture the temporal behaviors of popularity evolution, this paper creates a series of embeddings made at different time points for both topological network and user statistics. Then these two series are combined into one. Next, the combined series is fed into LSTM, a specific type of recurrent neural network. Finally, the output of LSTM and the embedding of accumulation time factor are concatenated together as the embedding of dynamic factors. The process of dynamic factor embedding is shown in Fig. 3.

Accumulation time embedding. This factor gives the amount of time it takes when cumulative popularity reaches a certain level, which is also the period before a prediction is triggered. This paper denotes the embedding of accumulation time by $[V_t^A]$, where t is the triggering time of prediction.

Topological network embedding. This aims to learn a low-dimensional dense vector for each node [33,34]. This paper denotes the cumulative evolving network by N_t for a hashtag at time $t \in \{1, 2, \ldots, L\}$. The vertices of N_t are users who have tweeted on the hashtag in hours 0 through t. If vertex u and vertex v have a follower-following relationship, an edge between u and v is added. The follower-following relationships come from a data set collected by Kwak et al. [16].

Then this paper utilizes DeepWalk [28] algorithm to make an embedding for N_t. DeepWalk is used for learning latent representations of vertices in a network. These latent representations encode social relations. DeepWalk uses local information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences, so that language modeling techniques like SkipGram [24] algorithm can be adopted for encoding these random walks. In this paper, $[V_t^T]$ denotes the embedding of N_t. Since N_t can have millions of vertices, we don't choose to take every single vertex into account for the sake of computation speed in handling neural networks. This paper chooses the top k significant vertices $(v_1, v_2, \ldots, v_k)$ in the network [31]. By top k significant vertices, we mean the vertices that have the top k number of neighbors. V_t^T is given by Eq. (2).

$$V_t^T = [\Phi(v_1), \Phi(v_2), \ldots, \Phi(v_k)] \tag{2}$$

where $\Phi(v)$ is the embedding of vertex v learned by DeepWalk algorithm. Φ is the mapping function. This mapping Φ represents the latent social representation associated with vertex v in N_t. According to DeepWalk algorithm, $\Phi(v)$ is calculated by SkipGram [24] algorithm and random walks starting from v. Hence, the series of topological network embedding is given by $\{V_{t_0}^T, V_{t_1}^T, \ldots, V_{t_n}^T\}$, where t_n is the triggering time of prediction. $n+1$ is the number of items in this series. The time unit can be set to one hour.

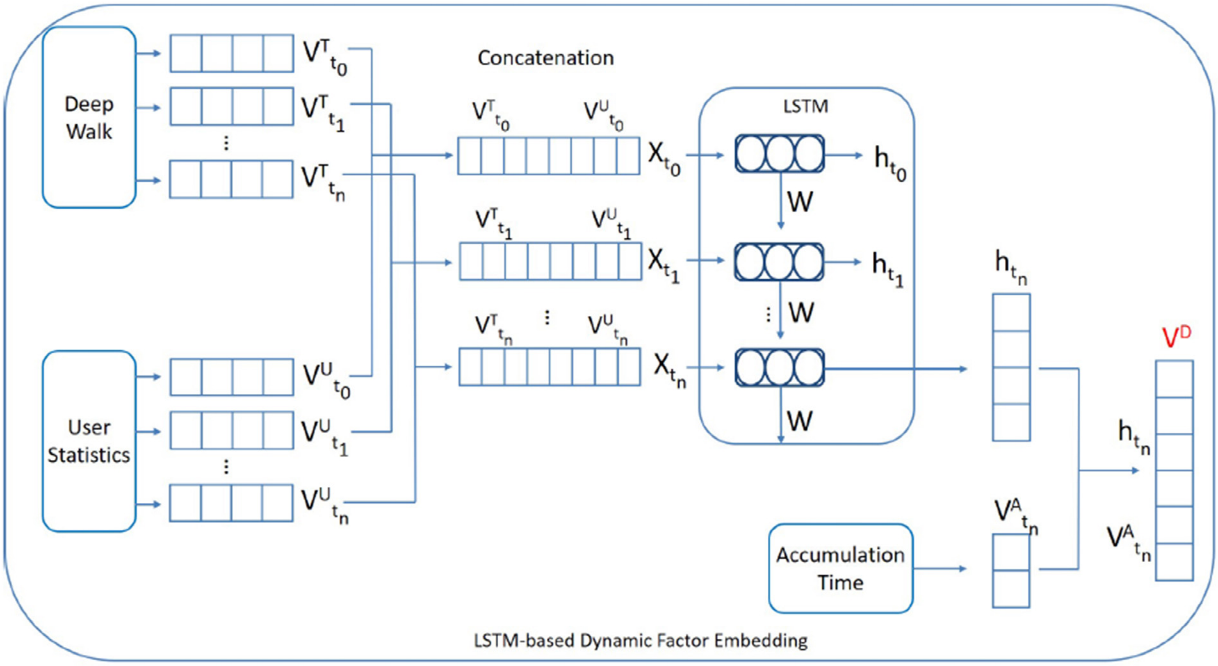

Fig. 3. The LSTM-based dynamic factor embedding.

User statistics embedding. Celebrity involvement in discussing a hashtag can prolong its active period. This paper collects statistics for celebrities and their fans to make an embedding denoted as follows.

$$V_t^U = [m_t^0, m_t^1, m_t^2, m_t^3, m_t^4] \tag{3}$$

Given a hashtag, m_t^0 represents the number of celebrities involved until the time point t. m_t^1, m_t^2, m_t^3 and m_t^4 represent the overall sum, maximum, median and average of the number of fans respectively. Hence, the series of user statistics embedding is given by $\{V_{t_0}^U, V_{t_1}^U, \ldots, V_{t_n}^U\}$.

The embeddings of topological network and user statistics are concatenated together (Eq. (4)) and then sent into LSTM.

$$X_t = [V_t^T, V_t^U] \tag{4}$$

The first step of LSTM is to decide what information we're going to throw away from the cell state as follows.

$$f_t = \sigma(W_f \bullet [h_{t-1}, X_t] + b_f) \tag{5}$$

where f_t is the forget gate layer with a sigmoid function. h_{t-1} is the output of the predecessor. W_f is the weight matrix of forget gate layer. σ is a sigmoid function. b_f is a bias term.

The next step is to decide what new information to be stored in the cell state, as shown in Eqs. (6) and (7). i_t is the input gate layer with a sigmoid function and this layer decides which values will be updated. Then a tanh layer creates a vector of new candidate values $\tilde{C}_t$, that could be added to the state. W is a weight matrix. b is a bias term.

$$i_t = \sigma(W_i \bullet [h_{t-1}, X_t] + b_i) \tag{6}$$

$$\tilde{C}_t = \tanh(W_C \bullet [h_{t-1}, X_t] + b_C) \tag{7}$$

The old cell state C_{t-1} is updated into the new cell state C_t as follows.

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{8}$$

The final step is to decide what to be output, as shown in Eqs. (9) and (10). o_t is the sigmoid layer which decides what parts of the cell state to be output. The cell state C_t is put through tanh and multiply it by the output of sigmoid gate.

$$o_t = \sigma(W_o \bullet [h_{t-1}, X_t] + b_o) \tag{9}$$

$$h_t = o_t \times \tanh(C_t) \tag{10}$$

The embedding of dynamic factors is represented as shown in Eq. (11) by concatenating h_t and V_t^A. V_t^A is the embedding of accumulation time at the time point t.

$$V^D = [h_t, V_t^A] \tag{11}$$

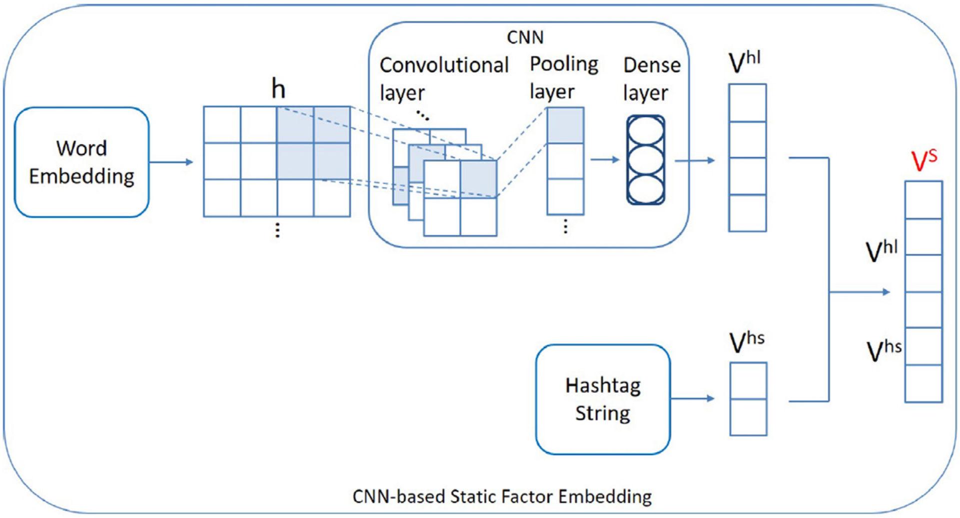

Fig. 4. The CNN-based static factor embedding.

4.3. Static factors embedding

For static factors, this paper considers hashtag content. The process of static factors embedding is shown in Fig. 4. To embed hashtag content, this paper extracts two types of features from a hashtag: hashtag string feature and hashtag lexical feature. The hashtag string feature is represented by a vector, composed of string length and the number of individual words in this string. The hashtag lexical feature is extracted by word embedding and CNN. Next, the embeddings of string feature and lexical feature are concatenated together.

Hashtag content embedding. Two types of features are extracted: hashtag string feature and hashtag lexical feature. For the hashtag string feature, we manually separate hashtag strings into individual words and count the number of individual words. For example, a hashtag "#a love like this is on your side" has eight individual words. Hashtags which have no clear meaning, like "#abcdefg" and "#bbmg", are considered as one word. The hashtag string feature vector V^{hs} is composed of string length len_str and the number of individual words num_wrd in this string as follows.

$$V^{hs} = [num\_wrd, len\_str] \tag{12}$$

Hashtag lexical feature is extracted as follows. A hashtag can be treated as a sequence of words. This paper converts each word to a word vector according to a pre-trained wiki text corpus. First, these word vectors are concatenated to build a matrix. In the matrix, each column corresponds to a word as shown in Eq. (13). Second, a convolutional layer is applied to the matrix h as shown in Eqs. (14) and (15).

$$h = \begin{bmatrix} | & | & | \\ h_1 & \cdots & h_n \\ | & | & | \end{bmatrix} \tag{13}$$

$$c = [c_1, c_2, \ldots, c_n] \tag{14}$$

$$c_j = g\left(W_c \bullet \begin{bmatrix} h_j \\ \vdots \\ h_j + l - 1 \end{bmatrix} + b_c \right) \tag{15}$$

where n is the number of filters and l is the window size of a filter. W_c is the filter weight matrix of convolutional layer. g is a non-linear activation function, such as ReLu. b_c is a bias term.

Third, a max pooling layer is applied to the output of convolutional layer as follows.

$$c_m = \begin{bmatrix} \max(c_1) \\ \vdots \\ \max(c_n) \end{bmatrix} \tag{16}$$

Fourth, a dense layer with a fixed number of neurons is applied to the output of the max pooling layer, as shown in Eq. (17). So the lexical feature extracted from variable-length hashtags can be a fixed-length vector,

$$V^{hl} = g(W_d \bullet c_m + b_d) \tag{17}$$

where W_d is the weight matrix of the dense layer. b_d is a bias term.

The embedding of static factors is represented as follows.

$$V^S = [V^{hs}, V^{hl}] \tag{18}$$

4.4. Normalization

For different factors, their values of embeddings vary a lot. For instance, the topological network embedding is represented by a vector with each component between −1 and 1. The user statistics embedding is represented by a vector with each component ranging from zero to millions. The accumulation time embedding is represented by a vector with each component ranging from one to thousands. The hashtag string embedding is represented by a vector with each component ranging from one to several tens. The hashtag lexical embedding is represented by a vector with each component ranging from 0 to 1. Therefore, we need to perform normalization to make all these embeddings lie in the same range. This paper applies the max-min normalization to both the accumulation time embedding and hashtag string embedding. Since the values of the user statistics embedding can be very large, this paper takes the following method to perform normalization for the user statistics embedding.

$$x' = \frac{\log(x+1) - \min(\log(x+1))}{\max(\log(x+1)) - \min(\log(x+1))} \tag{19}$$

The component of user statistics embedding can be 0. So this paper adds 1 to each original value before taking the logarithm operation, in order to reduce its scale. Next, the max-min normalization is employed to the logarithm values.

4.5. Cost function

The model is trained by minimizing the cost function $J(\Theta)$ as shown in Eq. (20) with the stochastic gradient descent algorithm.

$$J(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \frac{h_i((V^D, V^S; \Theta) - d_i)^2}{d_i^2} \tag{20}$$

where i is the sample. d_i is the actual value of active period. $h_i((V^D, V^S; \Theta) - d_i)$ is the output of fully connected neural network that is given in Fig. 2. V^D is the embedding of dynamic factors and V^S is the embedding of static factors. Θ is a parameter vector that consists of all parameters in our neural networks. N is the number of samples in the training set.

The stochastic gradient descent algorithm minimizes the cost function by keeping updating each θ as follows.

$$\theta_j := \theta_j - \alpha \times \frac{\partial J(\Theta)}{\partial \theta_j} \tag{21}$$

$$\theta_j := \theta_j - \frac{\alpha}{N} \times \frac{\partial}{\partial \theta_j} \sum_{i=1}^{N} \frac{(h_i(V^D, V^S; \Theta) - d_i)^2}{d_i^2} \tag{22}$$

$$\theta_j := \theta_j - \frac{2\alpha}{N} \sum_{i=1}^{N} \frac{(h_i(V^D, V^S; \Theta) - d_i)}{d_i^2} \frac{\partial h_i(V^D, V^S; \Theta)}{\partial \theta_j} \tag{23}$$

5. Results

This section first discusses experimental setup, and then presents the overall prediction performance. For the next part, it explores the question that which factor has the most significant impact on the active period, among the factors of accumulation time, topological network, user statistics and hashtag content. Finally, this section compares our solution with baseline models. All predictions are evaluated in terms of absolute percentage error (APE) as follows.

$$APE = \frac{|d - d'|}{d} \tag{24}$$

where d is the actual value and d' is the predicted value.

5.1. Experimental setup

Experimental data include 20 thousand samples on the training set, 5000 samples on the validation set and 5000 samples on the test set. Predictions are triggered at different time points (e.g. the time points once cumulative popularity reaches 100, 500, 800, 1500 and 5000).

The hidden state h_t of each step for LSTM is represented by a m-dimensional vector, m can be set to 8, 16, 32 or 64. The number of neurons in the dense layer for the static embedding is n and n can be set to 8, 16, 32 or 64. Note that we have

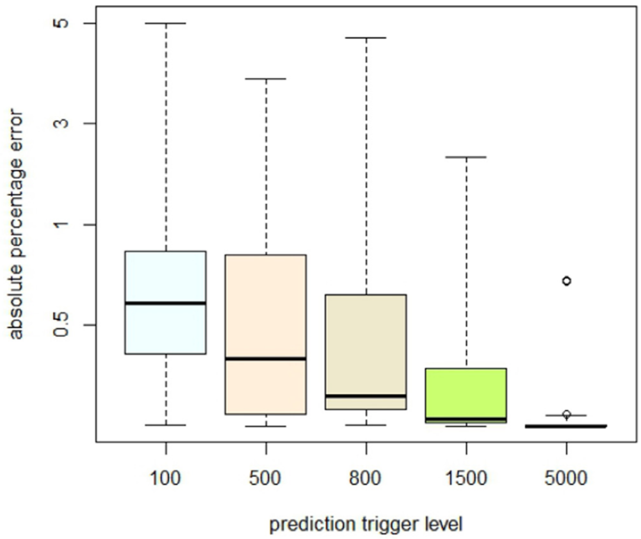

Fig. 5. The overall prediction performance.

Table 1
. Means and standard deviation for APE.

prediction trigger level	100	500	800	1500	5000
mean	0.623	0.478	0.363	0.226	0.012
standard deviation	0.355	0.416	0.382	0.341	0.066

another hyper-parameter k for top k significant vertices in DeepWalk, k can be set to 16, 32 or 64. These hyper-parameters are chosen through 5-fold cross validations by minimizing APE. The best combination of hyper-parameters is $m = 16$, $n = 16$ and $k = 32$. These hyper-parameters cannot be set to large values, otherwise it will cause a large APE. We deduce that our data set is not big enough, so the training gets heavily overfitted with large hyper-parameters.

5.2. Prediction performance

The overall performance of our solution is evaluated by the minimum, maximum, and quartiles of APE, as shown in Fig. 5. A box-and-whisker plot shows the minimum, quartiles, and maximum error values. The bottom and top of the box are the first and third quartile error values respectively. The band inside the box is the median error value. The upper and lower whiskers are the maximum and minimum error values respectively. The small circles in Fig. 5 are outliers whose values are more than 10 times larger than median values. The majority of our error values lie between 0 and 1. A few values are larger than 1. For better visualization, the vertical axis in Fig. 5 is rescaled unevenly in order to minimize the unimportant areas.

For the predictions triggered when cumulative popularity reaches 100, the median error value is about 0.6 and the maximum error value is almost 5. The reason for such a large maximum value is that the actual value of active period is very low. For example, if the actual value of active period is 1 and the predicted value of active period is 6, the error value can reach 5. For the predictions triggered when cumulative popularity reaches 800, the median error value is about 0.15, which indicates that accurate results can be achieved if predictions are triggered after cumulative popularity reaches 800. Even from the last box (the predictions are triggered when cumulative popularity reaches 5000), we can see that the maximum error value is smaller than 0.15. The positions of the median band become lower as trigger levels of prediction increase, which indicates that the overall prediction performance improves as trigger levels of prediction increase. This is because the later we trigger a prediction, the more information we can get.

The means and standard deviations for APE are calculated and presented in Table 1 for further analysis. (Note that medians are shown in Fig. 5. Means and standard deviations are getting lower as trigger level of prediction goes higher, which indicates better prediction performance.

5.3. Impact of factors

In order to understand which factor has the most significant impact on the active period, this paper separately sends the predictor of fully connected neural network each of these four factors, namely AT (accumulation time), TN (topological network), US (user statistics) and HC (hashtag content). Table 2 shows the results of different predictions, specifically the 1st, 2nd and 3rd quartiles of APE. The following observations can be made.

Table 2
The quartiles of APE by different trigger level of prediction.

	1st				2nd				3rd			
	AT	TN	US	HC	AT	TN	US	HC	AT	TN	US	HC
100	0.697	0.587	0.408	0.296	0.879	0.889	0.675	0.508	0.928	0.961	0.913	0.611
500	0.272	0.509	0.240	0.296	0.641	0.732	**0.432**	0.508	0.914	0.935	0.754	0.611
800	0.156	0.431	0.217	0.296	**0.227**	0.545	0.365	0.508	0.640	0.789	0.712	0.611
1500	0.074	0.259	0.229	0.296	**0.095**	0.443	0.331	0.508	0.417	0.633	0.481	0.611
5000	0.001	0.262	0.221	0.296	**0.003**	0.439	0.338	0.508	0.07	0.664	0.452	0.611

At different stages of popularity evolution, different factors have different levels of impact on the active period. At the initial stage of popularity evolution, none of these four factors has significant impact on the active period. At the early stage, only the user statistics factor contributes to active period predictions. The reason is probably that celebrity involvement occur more often at the early stage of popularity evolution. The more celebrity involvement there is and the more fans a celebrity has, the longer the active period can be. At the middle and late stages, the accumulation time is the most efficient factor contributing to the active period. If predictions only with the accumulation time factor are triggered after cumulative popularity reaches 1500, medians of APE are no larger than 0.01. The performances of predictions only with the accumulation time factor are almost the same as those of predictions with all four factors, which indicates that the accumulation time factor takes almost all credit for accurate predictions.

As predictions are triggered at higher and higher levels, both the user statistics factor and accumulation time factor perform better and better in active period predictions. The accumulation time factor is generally more efficient than the user statistics factor.

The topological network factor is surprisingly not that useful for active period predictions. Even at the late stage of popularity evolution, sufficient information can be obtained, medians of APE are larger than 0.4 though. This implies that accurate predictions cannot use only the topological network factor.

The hashtag content factor plays the least significant role at middle and late stages. The performance of predictions with the hashtag content factor doesn't change with different trigger levels of prediction, because the hashtag content is a static factor.

5.4. Comparison

To validate the effectiveness of our solution, this paper compares our solution with the following solutions [27]. This section conducts comparisons for predictions triggered when cumulative popularity reaches different levels (500, 1500 and 5000).

Using no normalization (NN). To show the effectiveness of our normalization, this section compares the solution using our normalization with the solution using no normalization.

Support vector regression (SVR). We would like to know how well machine learning methods perform on this task. Hence, this paper chooses SVR, a powerful state-of-the-art algorithm used for many regression tasks. This paper feeds SVR hand-crafting features as in [15], such as network topological features (average node degree, maximum node degree, global clustering coefficient, etc.), user features (the number of celebrities, the overall sum and maximum of the numbers of their followers) and accumulation time feature.

SpikeM. Since the fact that predicting active period is a new prediction task, most of the existing models for popularity evolution predictions are not capable to solve it. To compare our solution with existing works, SpikeM model [22] has been chosen, which can solve this task but is not specialized for it. This paper trains the SpikeM model by using historical popularity data starting from the beginning of popularity evolution up to the triggering time of prediction. Then the durations of active periods can be inferred from historical popularity data.

Fig. 6 shows comparison results. The vertical axis is rescaled again for better visualization, because most of the values of APE lie between zero and one. From each subfigure, we can see that the positions of median bands inside the boxes of our solution are lower than those of other solutions, which implies that our solution achieves better overall performance. Particularly, in Fig. 6(a) (predictions are triggered once cumulative popularity reaches 500), the first and second quartiles of our solution are lower than those of other solutions. However, the third quartile of our solution is higher than those of NN and SVR, which implies that we have relatively more cases with APE larger than 0.75 (the third quartile of SVR is around 0.75). We look into the cases with APE larger than 0.75 and find that these cases have relatively longer active periods. Therefore, it can be deduced that the dynamic factors are not efficient for cases with longer active periods, when the level of cumulative popularity is not high.

In Fig. 6(b) and (c) (predictions are triggered once cumulative popularity reaches 1500 and 5000), all quartiles and maxima are lower. This indicates that as the trigger level of prediction increases, the improvement level of our solution (which benefits from more historical information) compared to other solutions becomes larger. A possible reason for this fact is that the dynamic factors considered in our solution make great contributions to active period predictions.

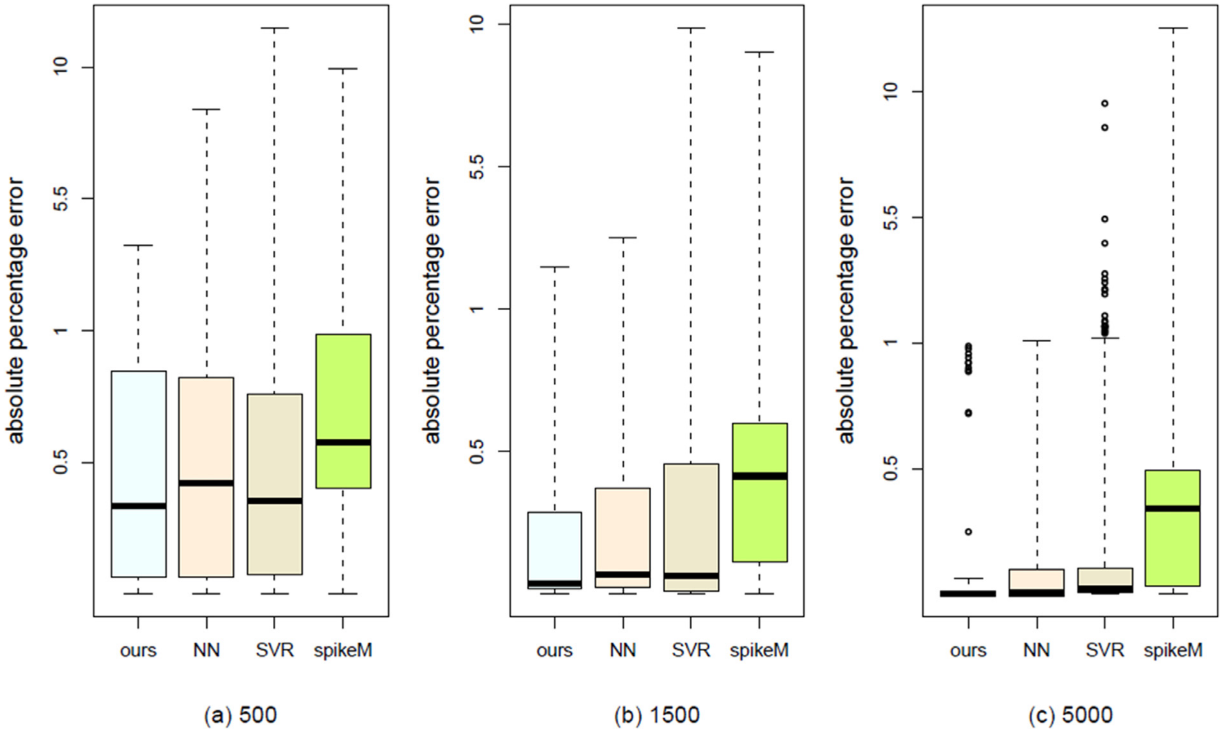

Fig. 6. Comparison results.

Another interesting observation can be seen from Fig. 6(c). The performance difference between our method, NN and SVR is not much. These three methods all perform well. What we can be told is that the performance difference between these three methods will become smaller and smaller as the trigger level of prediction goes higher. It is because these three methods all take the accumulation time factor into account. As the trigger level of prediction goes higher and higher, the accumulation time factor becomes more and more helpful and dominating for making predictions. This is consistent with what we can draw from Table 2. Sometimes different machine learning algorithms don't make a big difference and it is the feature that matters. Actually, the superiority of our method over other methods is more reflected in early predictions than it is in late predictions, which is still an advantage because in reality predictions should be triggered earlier for the sake of timeliness.

6. Conclusions

This paper puts forth a new and challenging prediction task in the field of popularity evolution: predicting the duration of active period. This paper presents a rigorous study that rises to the two challenges of this task. The first challenge concerning various factors is addressed by a DNN-based framework that makes embeddings for both dynamic and static factors. The second challenge concerning when to trigger a prediction is addressed by triggering the prediction when popularity reaches a certain level. The experiments are conducted on a Twitter hashtags data set. Experimental results show that our solution outperforms baseline methods. Furthermore, among the accumulation time, topological network, user statistics and hashtag content factors, this paper finds the accumulation time factor most significant for active period predictions of Twitter hashtags.

In future work, we will consider more factors to improve prediction performance, such as user interests and tweet content.

Declaration of interest statement

We state that this paper is an original research and there are no conflicts of interest. The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

We are especially grateful to Jure Leskovec and Sebastien Ardon for providing us Twitter data set. This work is supported by the National Key R&D Program of China under Grant no. 2017YFB0803302.

References

[1] S. Ardon, A. Bagchi, A. Mahanti, A. Ruhela, A. Seth, R.M. Tripathy, S. Triukose, Spatio-temporal and events based analysis of topic popularity in Twitter, in: Proceedings of the Twenty-Second ACM International Conference on Information & Knowledge Management, 2013, pp. 219–228.
[2] Z. Avramova, S. Wittevrongel, H. Bruneel, D.D. Vleeschauwer, Analysis and modeling of video popularity evolution in various on-line video content systems: power-law versus exponential decay, in: Proceedings of the First International Conference on Evolving Internet, 2009, pp. 95–100.
[3] R. Bandari, S. Asur, B.A. Huberman, The pulse of news in social media: forecasting popularity, in: Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media, 2012, pp. 26–33.
[4] A. Bielski, T. Trzcinski, Pay Attention to Virality: understanding popularity of social media videos with the attention mechanism, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 2335–2337.
[5] Y. Borghol, S. Ardon, N. Carlsson, D. Eager, A. Mahanti, The untold story of the clones: content-agnostic factors that impact youtube video popularity, in: Proceedings of the Eighteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1186–1194.
[6] M. Cha, H. Kwak, P. Rodriguez, Y.Y. Ahn, S. Moon, I tube, Youtube, everybody tubes: analyzing the world's largest user generated content video system, in: Proceedings of the Seventh ACM SIGCOMM Conference on Internet Measurement, 2007, pp. 1–14.
[7] B. Chang, Y. Park, S. Kim, J. Kang, DeepPIM: a deep neural point-of-interest imputation model, Inf. Sci. 465 (2018) 61–71.
[8] J. Cheng, L.A. Adamic, J.M. Kleinberg, J. Leskovec, Do cascades recur? in: Proceedings of the Twenty-Fifth International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, WWW, 2016, pp. 671–681. pp.
[9] R. Crane, D. Sornette, Robust dynamic classes revealed by measuring the response function of a social system, Proc. Natl. Acad. Sci. 105 (41) (2008) 15649–15653.
[10] F. Figueiredo, J.M. Almeida, M.A. Gonçalves, F. Benevenuto, Trendlearner: early prediction of popularity trends of user generated content, Inf. Sci. 349 (2016) 172–187.
[11] F. Figueiredo, J.M. Almeida, M.A. Gonçalves, F. Benevenuto, On the dynamics of social media popularity: a Youtube case study, ACM Trans. Internet Technol. (TOIT) 14 (4) (2014) 24.
[12] D. Gruhl, R. Guha, D. Liben-Nowell, A. Tomkins, Information diffusion through blogspace, in: Proceedings of the Thirteenth International Conference on World Wide Web, 2004, pp. 491–501.
[13] X. He, M. Gao, M.Y. Kan, Y. Liu, K. Sugiyama, Predicting the popularity of web 2.0 items based on user comments, in: Proceedings of the Thirty-Seventh International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014, pp. 233–242.
[14] C.C. Hsu, Y.C. Lee, P.E. Lu, S.S. Lu, H.T. Lai, C.C. Huang, C. Wang, Y.J. Lin, W.T. Su, Social media prediction based on residual learning and random forest, in: Proceedings of the Twenty-Fifth ACM International Conference on Multimedia, 2017, pp. 1865–1870.
[15] Y. Hu, C. Hu, S. Fu, M. Fang, W. Xu, Predicting key events in the popularity evolution of online information, PloS one 12 (1) (2017) e0168749.
[16] H. Kwak, C. Lee, H. Park, S. Moon, What is twitter, a social network or a news media? in: Proceedings of the Nineteenth International Conference on World Wide Web, 2010, pp. 591–600.
[17] K. Lerman, A. Galstyan, Analysis of social voting patterns on dig, in: Proceedings of the First Workshop on Online Social Networks, 2008, pp. 7–12.
[18] C. Li, J. Ma, X. Guo, Q. Mei, Deepcas: an end-to-end predictor of information cascades, in: Proceedings of the Twenty-Sixth International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, WWW, 2017, pp. 577–586. pp.
[19] Y.J. Li, Y. Peng, Z. Zhang, M.J. Wu, Q.Q. Xu, H.Z. Yin, A deep dive into user display names across social networks, Inf. Sci. 447 (2018) 186–204.
[20] J. Lv, W. Liu, M. Zhang, H. Gong, B. Wu, H. Ma, Multi-feature fusion for predicting social media popularity, in: Proceedings of the Twenty-Fifth ACM international conference on Multimedia, 2017, pp. 1883–1888. pp.
[21] Z. Ma, A. Sun, G. Cong, On predicting the popularity of newly emerging hashtags in Twitter, J. Assoc. Inf. Sci. Technol. 64 (7) (2013) 1399–1410.
[22] Y. Matsubara, Y. Sakurai, B.A. Prakash, L. Li, C. Faloutsos, Rise and fall patterns of information diffusion: model and implications, in: Proceedings of the Eighteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 6–14.
[23] M. Meghawat, S. Yadav, D. Mahata, A multimodal approach to predict social media popularity, in: Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval, 2018, pp. 190–195. pp.
[24] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv: 1301.3781, 2013.
[25] S.A. Myers, C. Zhu, J. Leskovec, Information diffusion and external influence in networks, in: Proceedings of the Eighteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 33–41.
[26] S.A. Myers, J. Leskovec, The bursty dynamics of the twitter information network, in: Proceedings of the Twenty-Third International Conference on World Wide Web, 2014, pp. 913–924.
[27] A.R.S. Parmezan, V.M.A. Souza, G.E.A.P.A. Batista, Evaluation of statistical and machine learning models for time series prediction: identifying the state-of-the-art and the best conditions for the use of each model, Inf. Sci. 484 (2019) 302–337.
[28] B. Perozzi, R.A.S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of the Twentieth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
[29] H. Pinto, J.M. Almeida, M.A. Gonçalves, Using early view patterns to predict the popularity of Youtube videos, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, 2013, pp. 365–374.
[30] M.A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, P.V. Hentenryck, Expecting to be hip: Hawkes intensity processes for social media popularity, in: Proceedings of the Twenty-Sixth International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 735–744.
[31] C. Song, X. Liu, T. Ge, Y. Ge, Top-k frequent items and item frequency tracking over sliding windows of any size, Inf. Sci. 475 (2019) 100–120.
[32] G. Szabo, B.A. Huberman, Predicting the popularity of online content, Commun. ACM 53 (8) (2010) 80–88.
[33] Q. Wang, Z. Wang, X. Ye, Equivalence between LINE and matrix factorization, arXiv preprint arXiv: 1707.05926, 2017, pp. 1–5.
[34] Z. Wang, X. Ye, C. Wang, Y. Wu, C. Wang, K. Liang, RSDNE: exploring relaxed similarity and dissimilarity from completely-imbalanced labels for network embedding, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 475–482.
[35] F. Wu, B.A. Huberman, Popularity, novelty and attention, in: Proceedings of the Ninth ACM Conference on Electronic Commerce, 2008, pp. 240–245.
[36] F. Wu, B.A. Huberman, Novelty and collective attention, Proc. Natl. Acad. Sci. 104 (45) (2007) 17599–17601.
[37] W. Xu, P. Shi, J. Huang, F. Liu, Understanding and predicting the peak popularity of bursting hashtags, J. Comput. Sci. 28 (2018) 328–335.
[38] J. Yang, J. Leskovec, Patterns of temporal variation in online media, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, 2011, pp. 177–186. pp.
[39] H. Yu, L. Xie, S. Sanner, The lifecyle of a Youtube video: phases, content and popularity, in: Proceedings of the Ninth International AAAI Conference on Web and Social Media, 2015, pp. 533–542.
[40] Q. Zhang, Y. Gong, J. Wu, H. Huang, X. Huang, Retweet prediction with attention-based deep neural network, in: Proceedings of the Twenty-Fifth ACM International on Conference on Information and Knowledge Management, 2016, pp. 75–84.
[41] Q. Zhao, M.A. Erdogdu, H.Y. He, A. Rajaraman, J. Leskovec, Seismic: a self-exciting point process model for predicting tweet popularity, in: Proceedings of the Twenty-First ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1513–1522.