Popularity Prediction for Single Tweet Based on Heterogeneous Bass Model

Xiaofeng Gao, *Member, IEEE*, Zuowu Zheng, Quanquan Chu, Shaojie Tang, *Member, IEEE*, Guihai Chen, *Member, IEEE*, and Qianni Deng

Abstract—Predicting the popularity of a single tweet is useful for both users and enterprises. However, adopting existing topic or event prediction models cannot obtain satisfactory results. The reason is that one topic or event that consists of multiple tweets, has more features and characteristics than a single tweet. In this article, we propose two variations of Heterogeneous Bass models (*HBass*), originally developed in the field of marketing science, namely Spatial-Temporal Heterogeneous Bass Model (*ST-HBass*) and Feature-Driven Heterogeneous Bass Model (*FD-HBass*), to predict the popularity of a single tweet at the early stage and the stable stage. We further design an Interaction Enhancement to improve the performance, which considers the competition and cooperation from different tweets with the common topic. In addition, it is often difficult to depict popularity quantitatively. We design an experiment to get the weight of favorite, retweet and reply, and apply the linear regression to calculate the popularity. Furthermore, we design a clustering method to bound the popular threshold. Once the weight and popular threshold are determined, the status whether a tweet will be popular or not can be justified. Our model is validated by conducting experiments on real-world Twitter data, and the results show the efficiency and accuracy of our model, with less absolute percent error and the best *Precision* and *F-score*. In all, we introduce Bass model into social network single-tweet prediction to show it can achieve excellent performance.

Index Terms—Heterogeneous bass model, single tweet popularity, time series prediction, Twitter social network

1 Introduction

Twitter, centered by users and communications, is one of the best-known social networks in the world. In recent years, research on prediction in social networks has received increased attention from both academia and industry. Many items in social networks are worth predicting, such as user's personality [1], popular stories [2], and interesting events [3]. Even a film's box office [4] and stock trend [5] with little relevance to social networks can be predicted through the contents posted by users.

Our goal is to quantitatively predict the popularity of a single tweet at any given time during its life cycle. Meanwhile, we also want to make a qualitative prediction by classifying a tweet as popular or unpopular. This work is significant for both ordinary users and companies. For users, it provides a tool to help them filter through a large amount of new content in order to identify interesting items in a timely manner. Another important application of this work is to help companies seize the opportunity to lead and generate a trend or hot topic. Lastly, abnormal popular tweets can set alarm for disaster, crime, or catastrophe. For example, Facebook is helping to catch criminals. Sometimes, the suspect inevitably brags about his deviant behavior on the social networks, which captures the user's attention. The police can get tips from this abnormal popular content.

Majority of existing work focus on predicting the popularity of topics or events, which consist of sets of single tweets. However, there are a few works that focus on predicting the popularity of a single tweet, because it only uses its own textual information and user's information with a timeline. Moreover, the lifespan of a single tweet is often shorter than a topic or event, which lacks enough time to compare the prediction with real trend.

Several works related to single tweet [6], [7], [8], [9], [10], [11] mostly concentrate on qualitative prediction. They predicted whether a tweet will be retweeted, which is a 2-class classification problem. In addition, the prediction of the tweet's trend can be categorized as a regression problem. However, the accuracy of regression models are often unstable, because regression models cannot capture the randomness that exists in the trend after a tweet has been posted. On the other hand, several works related to feature based methods [2], [12], [13], [14], [15], [16], [17], [18], [19] relied heavily on the most effective features for estimating popularity. Meanwhile, time-series methods have a better performance than feature based methods overall. Time-series based methods [20], [21], [22], [23], [24], [25] relied on statistical models or point process based methods, and those models are widely used in recent years. However, time-series based methods also have many drawbacks. For instance, Q. Zhao et al. [20] and S. Mishra et al. [21] built a model based on the random point process to predict the

- X. Gao, Z. Zheng, Q. Chu, G. Chen, and Q. Deng are with the Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {gao-xf, gchen, deng-qn}@cs.sjtu.edu.cn, {waydrow, spring_chu}@sjtu.edu.cn.
- S. Tang is with the Naveen Jindal School of Management, University of Texas at Dallas, Richardson, TX 75080 USA. E-mail: shaojie.tang@utdallas.edu.

Manuscript received 7 May 2018; revised 24 Sept. 2019; accepted 29 Oct. 2019. Date of publication 11 Nov. 2019; date of current version 1 Apr. 2021.
(Corresponding author: Xiaofeng Gao.)
Recommended for acceptance by S. Sadiq.
Digital Object Identifier no. 10.1109/TKDE.2019.2952856

trend of a tweet. However, it requires hard-collected features, such as the post time of all retweets related to the original tweet and the number of followers of each retweet user. Most of the above works use traditional machine learning methods or random point process methods. Nevertheless, those methods [20], [26] overlooked human activities that may contribute to popularity, such as favorite or reply. Furthermore, J. Xin et al. [27] introduced the Bass model into flickr prediction. However, they did not incorporate the characteristics of the social networks.

In this paper, we design the Heterogeneous Bass model (*HBass*) which contains two varieties, namely Spatial-Temporal Heterogeneous Bass Model (*ST-HBass*) and Feature-Driven Heterogeneous Bass Model (*FD-HBass*), to predict the popularity of a single tweet. The Bass model [28] is one of the most widely applied models in management science. It is originally used to model the sales of a newly put-on-market product to a group of people, which can predict the popularity of a newly posted single tweet in a social community. Shen [56] created 12 features for each topic and forecasted topic popularity using standard bass diffusion model in Twitter. However, due to the limitation of parameters number and the spatial and homogeneity assumption, it is not immediately clear the Bass model is applied to single tweet prediction.

Therefore, we propose two Bass models based approaches, ST-HBass and FD-HBass. ST-HBass focuses more on the spatial and temporal heterogeneity, while FD-HBass focuses more on the effect of different features. As a result, FD-HBass has better performance than ST-HBass in quantitative prediction and qualitative prediction. In reality, a series of tweets may concentrate on one common topic. Those tweets often influence each other's popularity. Considering the interaction between those tweets, we propose an Interaction Enhancement to measure the competition and cooperation from different tweets with the common topic. Furthermore, the performance of HBass is improved, which forms two variations, the STI-HBass and FDI-HBass. As a result, FDI-HBass has the best performance.

Moreover, compared with many previous works, we redefine the quantitative definition of popularity and the threshold to classify popular and unpopular tweets. We adopt a linear regression model to calculate popularity by combining favorite count, retweet count, and reply count as features. Instead of choosing the threshold by experience, we use *DBSCAN* and *k-medoids* clustering algorithms to get the median of two cluster centers, and multiply the indicators' weights respectively to get the threshold. Hence, we get the popularity and threshold from real datasets.

We focus on predicting the trend of a single tweet in all its life cycle. We further justify whether a single tweet will be popular. Generally, our model is not only suitable for predicting a single tweet's popularity, but also suitable for other social networks, which have asymmetric following-follower relationship, such as Weibo and Digg.

We summarize our contributions as follows:

- We incorporate Twitter features into the Bass model in social network single-tweet prediction to form the HBass model. To be specific, we aim to predict the trend of a single tweet, and whether the tweet will be popular in the end.
- We propose the Interaction Enhancement to consider the real situation that the different tweets with common topic have the interaction of competition and cooperation between each other.
- We redefine the quantitative definition of popularity that combines the relationship among favorite, retweet, and reply, and threshold to classify popular and unpopular tweets based on clustering method, instead of choosing the threshold by experience.
- We use real-world Twitter data to examine the efficiency of HBass. The simulation results show that the efficiency and accuracy of the quantitative prediction with less absolute percent error and the qualitative prediction with a better classification detection.

The rest of the paper is organized as follows. In Section 2, we introduce the related work in the field of social network prediction. In Section 3, we describe the overview of our work. In addition, we introduce our proposed HBass and Interaction Enhancement in Section 4. In Section 5, we describe the method to calculate popularity and the threshold. In Section 6, we illustrate our experiments on Twitter dataset with discussions. Finally, we conclude our work in Section 7.

2 Related Work

With the development of the social network, researchers began to explore it from numerous perspectives. First, they analyzed social network by discovering and finding the popular items and influenced users [29]. Second, they classified items and detect items [30]. Third, they recommended items to users [31], and found interesting items based on the location server [32]. Fourth, they found the similarity user through multiple social network [33].

As a whole, the prediction in social networks can be divided into two categories. One is user behavior prediction, which focuses on knowing a behavior will happen in advance or not, such as the retweet behavior [34]. The other is content prediction, which focuses on prediction for content. Our prediction in single tweet obviously belongs to content prediction. Therefore we introduce it in detail.

The researches for content prediction include events, topics [3], [35], and single post [22], [36], [38]. Most of the content prediction paid attention to predict events or topics that a group of people created. These predictions tend to predict whether a topic or event will be popular or how popular in the future. All these topics or events models need extra tools to generate topics or events at the first step, and then use self-designed model with machine learning methods to achieve their goals.

We summarize related literature on content prediction for single post in social network as Table 1. In this table, we choose 6 critical metrics to describe the related works, and their meanings are listed as follows.

- *Content*. Content is used to express the detail content type, such as text, video, image and media.
- *Dataset*. Dataset shows the data sources in different social network of those works.

TABLE 1
Related Literature on Content Prediction for Single Post in Social Network

Reference	Content	Dataset	Prediction Type	Reference Objective	Methodology	Features
AAAI2015 [6]	Text	Twitter	Boolean	Time-series	Hierarchical Dirichlet Process	Structural; Textual; Temporal
AOAS2014 [22]	Text	Twitter	Numerical	Time-series	Probabilistic Model based on Bayesian	-
Bigdata2017 [37]	Text	Twitter	Numerical	Time-series	Personalized Self-Excited Hawkes Process	-
CIKM2012 [13]	Text	Twitter	Numerical	Feature	Gradient Boosted Decision Tree	Social; Content; Time; Nodes
CIKM2015 [38]	Video	YouTube	Numerical	Time-series	Dual Sentimental Hawkes Process	-
CIKM2016 [7]	Text	Twitter	Boolean	Feature	Deep Nerual Network	Content; Interests
CIKM2016 [23]	Text	SWeibo	Numerical	Time-series	Hawkes Process	-
CIKM2016 [21]	Text	Twitter	Numerical	Time-series	Feature Driven and Point Process	User; Temporal
DASFAA2013 [24]	Text	TWeibo	Numerical	Time-series	Susceptible-Infected-Susceptible Epidemic	-
DASFAA2015 [14]	Text	Twitter	Numerical	Feature	Collaborative Filtering	Interests;Content; Influence
DASFAA2016 [39]	Video	Acfun.TV	Numerical	Feature	Dynamic Herding Effect Model	Content
ICDM2014 [16]	Text	SWeibo	Numerical	Feature	State Transition Based Prediction	Content
ICDM2014 [17]	Text	SWeibo	Numerical	Feature	BCI Model	Behavior; Content; Influence
ICDM2015 [25]	Text	TWeibo	Numerical	Time-series	Networked Weibull Regression	-
ICDM2015 [18]	Text	Twitter	Numerical	Feature	Generic Regression	Content; User
ICMR2014 [36]	Image	Flickr	Boolean	Feature	Support Vector Machine	Image Context; Image Content; User
ICWSM2012 [12]	Text	NewsKnife; Twitter	Numerical	Feature	K-Nearest Neighbors Regression	Category Score; Subjectivity; Named Entities; Source Score
ICODSE2014 [15]	Text	Twitter	Numerical	Feature	Support Vector Regression	Temporal; Structural; Content; Poster
SIGIR2014 [8]	Text	TWeibo	Boolean	Time-series	Diffusion-Targeted Influence Model	-
SIGIR2014 [40]	Media	YouTube;	Numerical	Feature	Regularization-Based Ranking Algorithm	Comments; Temporal; Social
SIGKDD2013 [9]	Text	TWeibo	Boolean	Feature	Orthogonal Sparse Logistic Regression	Important Nodes
SIGKDD2015 [20]	Text	Twitter	Numerical	Time-series	Self-Excited Hawkes Process	-
SocialCom2010 [10]	Text	Twitter	Boolean	Feature	Generalized Linear Model	Hashtag; Mention; Follower; Followee; Time; Status
TSIPN2018 [42]	Text	Twitter	Boolean	Time-series	Temporal User Topic Participation	-
WSDM2013 [41]	Video	Youtube	Numerical	Feature	Multivariate Linear Model; Multivariate Radial Basis Functions	Daily View Counts
WSDM2018 [45]	Media	Pinterest	Boolean	Feature	Tree-Structured Long Short-Term Memory Network	Image Propagation; User
WWW2010 [2]	Text	Digg	Numerical	Feature	Social Dynamic Model	Behavior
WWW2013 [19]	Text	SWeibo	Numerical	Feature	Logarithmic Linear Regression	Retweet Records
WWW2013 [11]	Text	Twitter	Boolean	Feature	Probabilistic Model	User; Tweet
WWW2015 [43]	Media	Behance; Twitter	Numerical	Time-series	Tensor-based Hierarchical Prediction	-
WWW2017 [44]	Media	Twitter; YouTube	Numerical; Boolean	Time-series	Hawkes Intensity Processes	-
WWW2017 [46]	Video	YouTube	Numerical; Boolean	Time-series	Hawkes Process	-

TWeibo represents the Tencent Weibo; SWeibo represents the Sina Weibo.

- *Prediction Type*. Prediction type can be divided into Boolean and Numerical, where Boolean represents that works aim at qualitative prediction and Numerical represent that they are quantitative prediction.
- *Reference Objective*. Reference objective represents the common method type for prediction. Generally, there are briefly two different methods to predict those situations, feature based methods and time-series based methods. As for feature based methods, many researches studied that the different features have a different effect on the popularity. Therefore, they always adopted the methods rely on data to find out the most effective feature for popularity. The feature-based methods have a moderate performance, which is stable during the peaking time. Several time-series based approaches have been proposed recently to predict the popularity, which improves greatly over time and produce quite satisfactory result.
- *Methodology*. Methodology shows the basic methodology of those works.
- *Feature*. If the method belongs to feature based methods or some special time-series methods, we will list the main features those works used.

Since the information spreads over time in Bass diffusion model, we need to consider the role of time. Dong et al. [54] mentioned that it is through historical data that one can understand the trends in the past and make predictions about the future. [55] exploited all diversified temporal factors to make a better prediction about places. Correspondingly, we combine the characters of feature based methods and time-series methods. That is, it not only considers the diffusion with time but also combines the Twitter features. Therefore, they are more close to reality. Besides, few existing researches give a definition of popularity moderately. Zaman et al. [22], [43], [47] considered the number of retweets as the popularity. We design an experiment to calculate the popularity and further calculate the threshold based on real world data, which is more reliable.

For works related to our work most, they often use time series based method to achieve prediction task. The newly published [20], [21] and [37] are the most relevant works that predict the popularity of a single tweet. [20] designed a Self-Exciting Point Process model (*SEISMIC*), and [37] used a Personalized Self-Exciting Point Process model (*PSEISMIC*),which performed cluster analysis to group all tweets so that the coefficient of memory kernel can be estimated for every cluster. Mishra et al. [21] combined the Hawkes process and features to get the (*Hybrid*). However, as the shortages mentioned in Section 1, the features related to the information of retweet are hard to track when generating the data set in real time. Meanwhile, they ignored to predict the popularity in all its life cycle and the status whether the tweet is popular or not.

3 Overview of Prediction Process

This section introduces the prediction process as a whole. First, the notations and concepts, including the problem statement, are described. Then, the prediction process will be introduced overall.

3.1 Problem Statement

After a tweet is posted by a user, the followers of the user will see this tweet. At any time t after it has been posted, the followers can like, reply or retweet this tweet, which increases its favorite count $fc(t)$, reply count $pc(t)$, or retweet count $rc(t)$, respectively. Once a follower retweets this tweet, it can be seen by the follower's followers. Therefore, $fc(t)$, $rc(t)$, and $pc(t)$ keep increasing until no one interacts with it. In order to state our work clearly, we give some definitions first.

Definition 1 (Popular Determined Time). *The popular determined time T is the time to classify the status of a tweet with favorite count, retweet count and reply count.*

Definition 2 (Tweet Life Cycle). *A tweet life cycle is the period in which the tweet receives favorite count, retweet count or reply count continuously. After that period, the popularity of tweet remains unchanged.*

The common life cycle of a tweet includes early stage ($1h-24h$ after posted) and stable stage ($25h-240h$ after posted). The popular determined time of hot tweets can set as the end of each stage.

Some researches [20], [26] regarded the retweet count or favorite count as popularity count. However, such approach does not depict reality in some way. The retweet count can only represent part of the one who likes sharing messages, but can poorly express the one who concentrates on the messages with no sharing. In order to reflect the situation factually, our model introduces favorite count, retweet count, and reply count when defined popularity count.

Definition 3 (Popularity Count). *Favorite count, retweet count and reply count all represent a tweet's popularity. We define the popularity count $Y(t)$ as:*

$$Y(t) = w_1 fc(t) + w_2 rc(t) + w_3 pc(t), \tag{1}$$

where $0 < t \leq T$, and w_1, w_2 and w_3 respectively represent the weight of favorite, retweet and reply, reflecting the user's preference of behaviors to some extent.

TABLE 2
Symbol Description

Symbol	Description
n	The number items in the dataset
t	Any time in the tweet's life cycle
T	The popular determined time
$fc(t), rc(t), pc(t)$	The count of favorite, retweet, and reply at time t
$\bar{fc}_t, \bar{rc}_t, \bar{pc}_t$	Average count at time t for favorite, retweet and reply
$f(t)$	Probability density function of time t
$F(t)$	Cumulative distribution function of time t
$Y(t)$	The prediction popularity at time t
$Y^{'}(t)$	The popularity at time t after Interaction Enhancement
$G(t)$	The real popularity at time t
$\mathbf{w}$	The vector of weight in indicators
$\mathbf{x}, \mathbf{y}$	The vector of user features and single-tweet features
$\boldsymbol{\alpha}$	The parameter vector of user features
$\boldsymbol{\beta}$	The parameter vector of single-tweet features
γ	The popularity threshold
p, q	The coefficient of *innovators* and *imitators*
$\delta_1, \delta_2, \delta_3$	Constant for Interaction Enhancement
C	Constant for integration

Sometimes, users only intend to acquire whether a tweet is popular or not. They may not care about the number of the popularity count. Thus, our model uses a threshold to classify the popularity to get the popular status. Then we define the popular threshold.

Definition 4 (Popular Threshold). *Popularity Threshold γ is the boundary of a tweet which belongs to Popular or Unpopular. It is defined as:*

$$\begin{cases} Y(T) \geq \gamma & \text{the tweet is popular} \\ Y(T) < \gamma & \text{the tweet is unpopular} \end{cases},$$

where T is the popular determined time and $Y(T)$ is the approximate final popularity count of a tweet.

The method to calculate the popular threshold is introduced at Section 5.2. For clarity, we summarize the symbols with their meaning in Table 2.

3.2 Framework Overview

In this paper, we aim to predict the popularity $Y(t)$ in all its life cycle with the user features and single-tweet features. After that, we can get the trend of a tweet. Finally, we can estimate the status whether the tweet is popular or unpopular with the popular threshold γ and the approximate final popularity count $Y(T)$.

To solve the problem, we design a framework for our prediction system. The overall framework is shown as Fig. 1. The data preprocessing module is used to filter the raw user features and single-tweet features, and select the popularity indicators. The parameter setting module uses all the popularity indicators dataset that are formed from the data preprocessing module to get the indicators' weight and the final popularity threshold. The prediction module is used to predict the trend of a

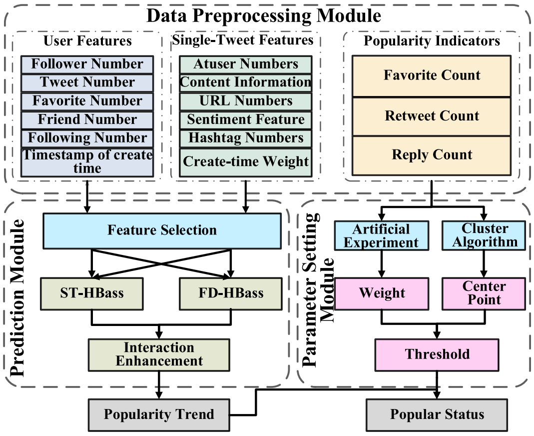


Fig. 1. Overall framework for our prediction system.

tweet. After that, we can determine whether it will be popular or not at the end.

The data preprocessing module is introduced in Section 6.4. Then, the prediction module is introduced in Section 4, and the parameter setting module is introduced in Section 5.

4 The Heterogeneous Bass Model

The Bass model [28] was proposed to predict the sales of a new product. Now, it is being widely used in many kinds of researches. Since no need of large numbers of the training set, given the first several days or months of sales of a new product, we can easily predict the performance of the product later using only two parameters.

Although the Bass model is an excellent model in economic fields, it is not straightforward to transplant Bass model directly for single tweet prediction due to the limitation of parameters number and the spatial and temporal homogeneity assumption. According to [48], there are three kinds of terms to reflect the spatial heterogeneity: the intrinsic probability of adoption, the susceptibility to intra population linkages and the infectiousness of adopters. For this limitation, we can enroll Twitter features into the original model, and relax it to individual-level heterogeneity. Due to the character of Twitter features, we propose Heterogeneous Bass model (*HBass*), which has two variations, namely Spatial-Temporal Heterogeneous Bass model (*ST-HBass*) and Feature-Driven Heterogeneous Bass model (*FD-HBass*), from different perspectives, introduced in Sections 4.1 and 4.2, respectively. In addition, we design the Interaction Enhancement to improve the performance by external factors in Section 4.3.

4.1 ST-HBass Model

The standard Bass model assumes spatial and temporal homogeneity, leading to no distinction of individuals, which is unsuitable for tweet prediction. To relax the limitation, we propose the ST-HBass model, which focuses more on the spatial and temporal heterogeneity.

From the standard model [28], we can get the standard formulation of a diffusion process as:

$$f(t) = (p + qF(t))(1 - F(t)),$$

where p is the coefficient of *innovators* (or the impact of factors outside the population), q is the coefficient of *imitators* (or the impact of contacts within the population). $f(t)$ is the hazard rate of adoption, which is also known as the likelihood of purchasing at time t. $1 - F(t)$ is the probability that one has not yet adopted at time t. Obviously, $F(t) = \int_0^t f(x)dx$.

Assuming the size of potential buyers is fixed and known as m, the number of purchases at t is:

$$S(t) = mf(t) = pm + (q-p)Y(t) - \frac{q}{m}[Y(t)]^2. \tag{2}$$

Where $Y(t) = \int_0^t S(x)dx$ is the accumulative number of sales. Then Eqn. (2) is the formulation of the standard Bass model.

Endow the features we can get from Twitter, we derive the model as below:

$$S(t) = pm + (q-p)Y(t) - \frac{q}{m}[Y(t)]^2 + \boldsymbol{\alpha x} + \boldsymbol{\beta y}.$$

Here $\boldsymbol{x}$ is the user features vector and $\boldsymbol{y}$ is the single-tweet features vector that extracted from a tweet.

In order to find $Y(t)$ we must solve the non-linear differential equation:

$$\frac{dY}{dt} = pm + (q-p)Y(t) - \frac{q}{m}[Y(t)]^2 + \boldsymbol{\alpha x} + \boldsymbol{\beta y}.$$

For simplicity, let $V = pm + \boldsymbol{\alpha x} + \boldsymbol{\beta y}$. Then we have

$$\frac{mdY}{q[Y(t)]^2 + m(p-q)Y(t) - mV} = -dt. \tag{3}$$

Factoring the denominator on the left of Eqn. (3), we have

$$\frac{mdY}{(Y(t)-y_1)(Y(t)-y_2)} = -dt, \tag{4}$$

with $y_1 = \frac{m(q-p)+\sqrt{\Delta}}{2q}$, $y_2 = \frac{m(q-p)-\sqrt{\Delta}}{2q}$, and $\Delta = m^2(p-q)^2 + 4mqV$.

Change Eqn. (4) into:

$$\left(\frac{1}{Y(t)-y_1} - \frac{1}{Y(t)-y_2}\right)\frac{mdY}{y_1 - y_2} = -dt.$$

Then we can do integration on both sides of the equation,

$$\int_0^T \left(\frac{1}{Y(t)-y_1} - \frac{1}{Y(t)-y_2}\right)\frac{mdY}{y_1 - y_2} = \int_0^T -dt.$$

The solution is: $Y(t) = \frac{y_2 e^{-\frac{\sqrt{\Delta}}{mq}t+C} + y_1}{1+e^{-\frac{\sqrt{\Delta}}{mq}t+C}}$.

Because $Y(0) = 0$, the constant C generated by the integration can be solved by: $C = \ln(-\frac{y_1}{y_2})$. We have:

$$Y(t) = \frac{y_2 e^{-\frac{\sqrt{\Delta}}{mq}t+\ln(-\frac{y_1}{y_2})} + y_1}{1 + e^{-\frac{\sqrt{\Delta}}{mq}t+\ln(-\frac{y_1}{y_2})}}, \tag{5}$$

where $y_1 = \frac{m(q-p)+\sqrt{\Delta}}{2q}$, $y_2 = \frac{m(q-p)-\sqrt{\Delta}}{2q}$, and $\Delta = m^2(p-q)^2 + 4mqV$. Hence we get the ST-HBass model.

4.2 FD-HBass Model

From another perspective, we focus more on the effect of different features based on heterogeneity to the standard Bass model, which is a useful method to relax the limitation of the original Bass model. To distinguish the different effect on the two kinds of features, we propose the FD-HBass model. When considering the single-tweet features, they only impact the popularity count through the characteristics of the tweet itself. To a certain degree, they are similar to the *innovators* in the standard Bass model. Simultaneously, user features can reflect the propagation from a user to another user, to some extent, which similar to the *imitators*. Then, we can use the single-tweet features to replace the *innovators* $p = \beta y$. Meanwhile, use the user features to replace the *imitators* $q = \alpha x$.

In the standard Bass model [28], there is a basic assumption: the probability that the purchase will be made at t which has not been made, denoted as $P(t)$, is a linear function of the number of previous buyers $Y(t)$:

$$\frac{f(t)}{1-F(t)} = P(t) = p + \frac{q}{m}Y(t).$$

For $F(t) = \frac{Y(t)}{m}$, we have:

$$f(t) = [p + qF(t)][1 - F(t)] = p + (q-p)F(t) - q[F(t)]^2.$$

In order to obtain the $F(t)$, we must solve this non-linear differential equation:

$$dT = \frac{dF}{p + (q-p)F - qF^2}.$$

Then the solution is: $F(t) = \frac{q - pe^{-(t+C)(p+q)}}{q(1+e^{-(t+C)(p+q)})}$.

Since $F(0) = 0$, then the integration constant may be evaluated: $-C = \frac{\ln\frac{q}{p}}{p+q}$. Hence,

$$F(t) = \frac{1 - e^{-(p+q)t}}{1 + \frac{q}{p}e^{-(p+q)t}}$$

$$Y(t) = mF(t) = \frac{m(1 - e^{-(p+q)t})}{1 + \frac{q}{p}e^{-(p+q)t}}.$$

Then the solution is:

$$Y(t) = \frac{m(1 - e^{-(\beta y + \alpha x)t})}{1 + \frac{\alpha x}{\beta y}e^{-(\beta y + \alpha x)t}}. \tag{6}$$

As a result, we get the Feature-Driven Heterogeneous Bass model.

4.3 Interaction Enhancement

As we know, most of the tweets always describe one or more items, which are always summarized as hashtags. As a matter of fact, not every tweet has a hashtag, but each tweet has its own topic. Although, hashtag is the best generalization of the topic because it is a summary by manual method.

From the real situation, different tweets with common topic have the interaction of competition and cooperation.

As for the competitive interaction, people always tend to concern one of the tweet emphatically which attracts them most among various tweets with common topic. Then, the final popularity of the tweet is to balance the popularity that other tweets gain from this tweet and the popularity of this tweet obtains from other tweets with the common topic, which can be represented as:

$$Y^{'}(t) = Y(t) - \delta_1 + \delta_2. \tag{7}$$

Where $Y^{'}(t)$ is the final popularity, and $Y(t)$ is the HBass in Eqns. (5) or (6). Then δ_1 indicates the popularity of other tweets gain from this tweet and δ_2 shows that the popularity of this tweet obtains from other tweets with the common topic.

On the other hand, as for cooperation interaction, when the phenomenon of several tweets with common topic appears in a time period, it might attract more users to concern about this topic, which can increase the popularity of each tweet in turn. Then, we have

$$Y^{'}(t) = Y(t) + \delta_3. \tag{8}$$

Where δ_3 represents the cooperation interaction parameter, which express the cooperation interaction of tweets with a common topic.

Then, we further combine the Interaction Enhancement and HBass with Eqns. (7) and (8), to get the STI-HBass as:

$$Y(t) = \frac{y_2 e^{-\frac{\sqrt{\Delta}}{mq}t + \ln(-\frac{y_1}{y_2})} + y_1}{1 + e^{-\frac{\sqrt{\Delta}}{mq}t + \ln(-\frac{y_1}{y_2})}} - \delta_1 + \delta_2 + \delta_3.$$

FDI-HBass as:

$$Y(t) = \frac{m(1 - e^{-(\beta y + \alpha x)t})}{1 + \frac{\alpha x}{\beta y}e^{-(\beta y + \alpha x)t}} - \delta_1 + \delta_2 + \delta_3.$$

Hence, we successively get the four HBass in different ways. By means of considering the tweet from different perspectives and aspects, our model can be more realistic to express the popularity of real world tweet.

In addition, we adopt the *Least Square Method*, which is one of the recommended mathematical methods to solve the standard Bass model, to get the parameters of the model. With the parameters, we can easily predict the popularity count of a tweet at any time t ($t > 1h$) in all its life cycle.

5 The Measurement of Popularity

Popularity is a target to measure the degree of the popular tweet. However, there are few formal methods to measure the popularity. Most researches tend to choose the threshold by experience. In this section, we first introduce the method to calculate the popularity in Section 5.1. Then, the clustering method to bound the popularity threshold is introduced in Section 5.2.

5.1 Popularity Calculation

In order to measure the degree of the popular tweet, we do the experiment to estimate the weight of each indicator in popularity. The popularity can be represented by three indicators, favorite, retweet and reply, and further extend them to the same size, which are written as $\mathbf{z_1}$, $\mathbf{z_2}$ and $\mathbf{z_3}$. Then the dataset of popularity can be represented as $\mathbf{Z} = (\mathbf{z_1}^T, \mathbf{z_2}^T,$

$\mathbf{z_3}^T$). Here $\mathbf{Z}$ is a matrix of dimensions $n \times 3$, where n is the number of the data.

First, we calculate the favorite count, retweet count and reply count for each tweet in dataset, respectively. Then the average count of each indicator $\bar{z}_i$ can be determined by $\bar{z}_i = \frac{\sum_{j=0}^{n} z_i^j}{n}$, where $i = 1, 2, 3$ and z_i^j is the $i-th$ indicator count of $j-th$ tweet. Lastly, the weight of $i-th$ indicator w_i can be represented as $w_i = \frac{\bar{z}_i}{\sum_{j=1}^{3} \bar{z}_j}$.

After calculating the weight of the three indicators, it is simple to obtain the $Y(T)$ with Eqn. (1).

5.2 Threshold Selection

To classify the tweet as popular or unpopular, it is difficult to select an appropriate threshold. If the threshold is too small, many suppositional popular tweets will cover the factually popular tweet. Otherwise, if the threshold is too large, few tweets will attract our attention. For some related work [47], [49], they just selected the threshold subjectively, which may lead to a terrible result if the threshold is not suitable.

In this section, we design a method based on real world dataset to acquire the threshold suitably. In detail, we filter out some exception data and then cluster with three popularity indicators, favorite count, retweet count and reply count, into two clusters which respectively represent the congregation of popular tweet and unpopular tweet. It is effective to use the *DBSCAN* and $k-medoids$ clustering algorithm with k = 2. The popularity dataset has obvious geometric characteristics, so we adopt the euclidean Distance to calculate the distance.

Algorithm 1. Popularity Threshold Algorithm

Input: Dataset matrix $\mathbf{Z}$ contains three indicators of favorite $\mathbf{z_1}$, retweet $\mathbf{z_2}$ and reply count $\mathbf{z_3}$.
Output: The popular threshold γ.

```
Phase 1: calculate the indicators' weight w
```
Calculate the favorite count, retweet count and reply count for each tweet in dataset, respectively.
Calculate the average count of each indicator $\bar{z}_i$
The weight of $i-th$ indicator $w_i = \frac{\bar{z}_i}{\sum_{j=1}^{3} \bar{z}_j}$

```
Phase 2:filter out outliers points
```
Find the points in the radius neighborhood of every point, and identify the core points with more than minPts neighbors
Find the connected components of core points on the neighbor graph, ignoring all non-core points
Assign each non-core point to a nearby cluster if it is an radius neighbor, otherwise assign it to outliers

```
Phase 3:calculate the center of clusters
```
Randomly select two of n points as the medoids
Associate each point to the closest medoid
repeat
  foreach *medoid* m *and non-medoid point* p **do**
    swap m and p, connect each point to the closest medoid, recompute the total distances
    if *total distances increased* **then**
      undo the swap
until *total distances of points to their medoid increases*;
Calculate the average point $\overline{\mathbf{p}}$ of $\mathbf{c_1}$ and $\mathbf{c_2}$
return $\gamma = \mathbf{w}\overline{\mathbf{p}}$;

However, outliers points in datasets may lead to poor clustering results. To better distinguish the two different status, we first calculate the popularity with the points and its weight, and then use *DBSCAN*, a density-based clustering algorithm, to filter out those outliers points. The radius and minPts (the minimal points of each circle) are adjusted dynamically according to the Silhouette Coefficient, which is commonly used to evaluate how well each point has been classified. Then we get two relatively fixed cluster centers based on the $k-medoids$ algorithm with k = 2. After acquiring the two cluster centers, we set the center point of the two cluster center points as the threshold point, and represent it as $\overline{\gamma} = (\overline{x_1}, \overline{x_2}, \overline{x_3})$. The popular threshold can be represented objectively as $\gamma = w\overline{\gamma} = w_1\overline{x_1} + w_2\overline{x_2} + w_3\overline{x_3}$.

In order to evaluate the final clustering performance, we use two metrics: Silhouette Coefficient and Davies-Bouldin index (DBI). The Silhouette ranges from -1 to 1, where higher score means more distinct clusters, while lower DBI score means better separation between clusters. The results of two metrics are 0.83 and 0.35, respectively, which shows that the clustering quality is good. In addition, we study the impact of the radius of DBSCAN in Fig. 3, which indicates that improper radius can lead to poor clustering results.

All the process of calculating the popularity threshold γ are summarized in Algorithm 1. After obtaining the popular threshold γ, we deem the tweet as popular if $Y(T)$ is larger than γ, otherwise, it is unpopular.

6 Experiments

In this section, we first dig the characteristic of tweet and extract the useful features. Then we show the results of our proposed models on the real-world dataset and provide comparisons with various baseline methods.

6.1 Dataset Description

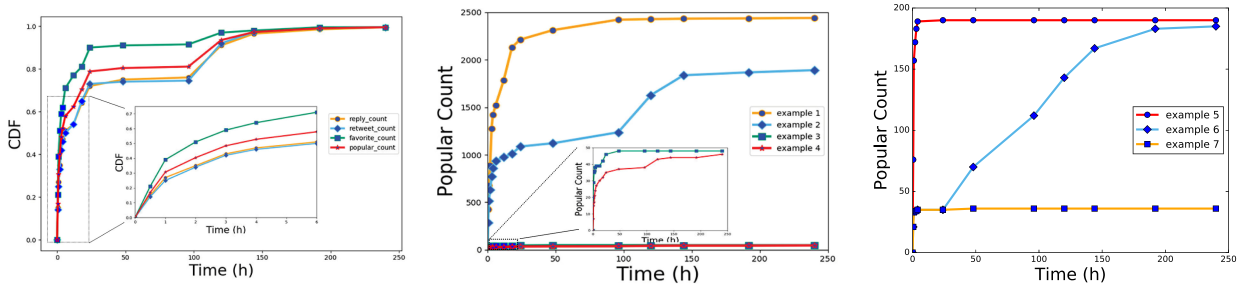
RANDOMTWEET. The dataset is randomly sampled from the Twitter API[1] ranging from 5th December, 2017 to 23rd February, 2018. We just crawl information about original tweets that filter out retweets, since the original tweets would be popular if some of its retweets are popular. Due to IP restriction in Twitter API and network restriction of policy, our crawler program runs effectively for about two hours a day, which limits the scale of dataset. Then, we only keep the tweets in English to keep consistency. Therefore, the final dataset contains 2516440 tweets with different time point information and 2122135 users. The detail of user-tweet relationship is as Table 3.

6.2 Characteristic Analysis for Tweet

To get the useful features, it is necessary to analyze the single tweet. There are several characteristics of single tweet.

Distribution Characteristic. From Table 4 we can see that 70.23 percent of the tweets receive no interactions. This is because the users do not have enough followers, or the contents of these tweets are not interesting, e.g., something meaningless. The rest 26.52 percent non-zero tweets mostly gain 1-10 popular counts. Only 3.25 percent tweets gain over 10 popular counts.

1. https://dev.twitter.com/



(a) CDF of tweet (b) Different trends of tweets (c) Different types of tweets

Fig. 2. Characteristic analysis for tweet.

Rising Characteristic. The popularity count of a single tweet increases rapidly during the first several hours after it is posted. As shown from Fig. 2a, whose mini box represents the relationship at the early time, about 90 percent tweets' favorite count, 70 percent retweet count and reply count remain unchanged. Referring to the distribution characteristic, we know that almost every one of these tweets has no interactions. There is an obvious increase at 100 hours. At this time, 90 percent tweets reach their final counts, and nearly 98 percent tweets become stable after $240h$. The rest tweets still receive interactions, although at a very slow rate.

Indicator Characteristic. We can also see from Fig. 2a that favorite count reaches its stationary much faster than the other two, which makes sense. Retweet or reply action includes something like chatting. Therefore, it takes less time to click your mouse to favorite a tweet than retweet or reply a tweet. For example, user A replies to user B's tweet, and B replies with something that A is interested in. Then A will reply again. Finally, the reply count will keep increasing. The same to retweet count.

Variety Characteristic. Tweets have different and multitudinous trends in real, which can be classified into four kinds as Fig. 2b (the mini box represents the relationship at the early time): gets high popular count immediately after posting, and stays popular (Example 1); slowly increases and finally gets high popular count (Example 2); increases at a high rate at the beginning, but are not popular in the end (Example 3); slowly increases and stays slow in the end (Example 4).

From another perspective, the variety characteristic of tweets reflects on their uncertainty. Fig. 2c shows two typical examples of the tweets' trends. Example 5 and Example 6 have different trend at the beginning and both get popular in the end. Otherwise, Example 6 and Example 7 have a similar trend in the observation period during the first 24 hours. However, due to different factors which we are exploring for, one gets popular and another stays unpopular. Our model is able to predict their difference.

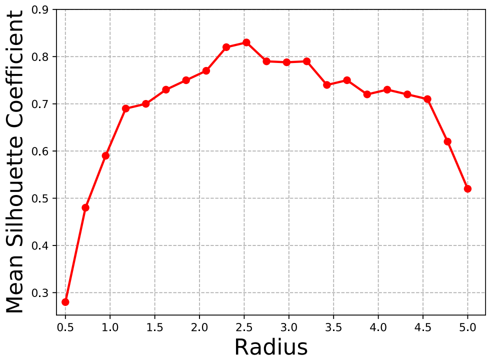


Fig. 3. The impact of radius in DBSCAN.

TABLE 3
The Relationship of User-Tweet

Table	Detail	Table	Detail	Table	Detail
Tweet	tweet id post time text catch time author id	User	user id number of followers number of followings number of tweets number of favorites register time	User-Tweet	user id tweet id

TABLE 4
Tweets' Popular Count Distribution

PopularCount	0	1-9	10-99	100-999	1000+
Potion(%)	70.23%	26.52%	2.11%	0.85%	0.29%

6.3 Parameter Settings

After we get the popularity weight of favorite count, retweet count and reply count from the Section 5.1, we can obtain the popular threshold naturally from Algorithm 1. When $T = 24h$, we have $\mathbf{w} = (0.28, 0.34, 0.38)$ and $\gamma = 140$. In addition, when $T = 240h$, we have $\mathbf{w} = (0.30, 0.34, 0.36)$ and $\gamma = 172$. With the threshold γ, we can smoothly classify the tweets as popular or unpopular in RANDOMTWEET, which is as Table 5.

Based on the above characteristics in Section 6.2, we can set the Popular Determined Time $T(hours)$ at the end of early stage $24h$ and stable stage $240h$ after a tweet posted.

6.4 Feature Extraction

In this section, we dig deeper about the correlation between features and popularity of tweets. We will analyze those relationships in detail as follows.

TABLE 5
The Description of RANDOMTWEET

T	Threshold	Popular Tweets	Unpopular Tweets
$24h$	140	26522	2489918
$240h$	172	22603	2493837

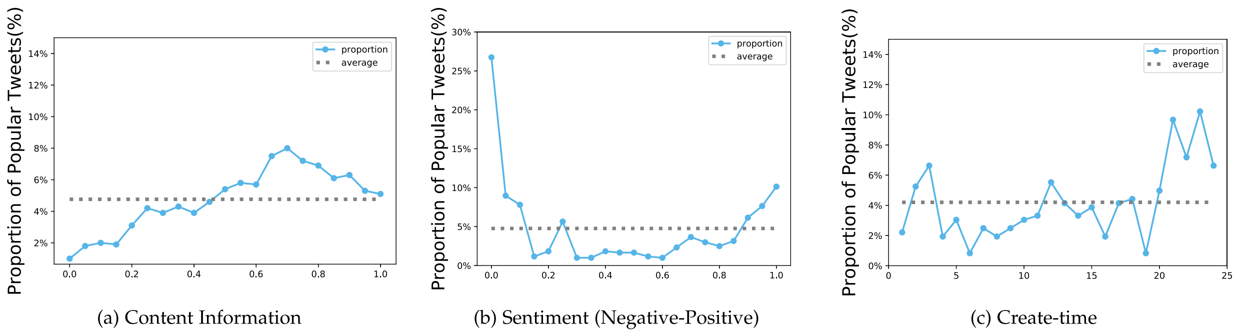


Fig. 4. Single-tweet features analysis part 1.

6.4.1 User Features

User features reflect the impact, of whom posted the tweet, on other users in social networks.

Follower Number. It well reflects the user's influence on the social network. That is to say, the larger the follower number is, the more users can be influenced.

Tweet Number. It reveals the active activity of the user. Generally, the active user will attract more users.

Favorite Number. It reflects the interests of users. Naturally, the interests in turn would attract more users.

Following Number. It is a good indicator for measuring the degree of concern for society. Respectively, the more following numbers the users have, the more attention the users can receive in social network.

Timestamp of Create Time. It is an indicator to distinguish the difference between old users and new users. In general, new users always get less focus than the old users since old users have a long time to get enough resources.

6.4.2 Single-Tweet Features

Single-Tweet Features indicate the characters of the tweet which are of great importance for its popularity.

Content Information. The long tweets do not always contain more information than short tweets due to the meaningless words. Thus, we use the content information rather than character number to measure the importance of tweets, which is calculated by LDA.

In detail, first, we process word tokenization and filter stop words. The number of all topics in datasets can be determined by Hashtags. Then, we can use LDA to calculate the topic probabilities $P = (p_1, p_2, \ldots, p_{n_p})$ of each tweet in a collection. The content information of each tweet is measured by $P \cdot N_p$, where $N_p = (n_p^1, n_p^2, \ldots, n_p^n)$ and n_p^i denotes the number of ith topic in the tweets set. After normalization, the value is between 0 and 1.

Fig. 4a shows the relationship between the content information of tweets and the proportion of popular tweets. It shows that the content information of a middle or larger level has a higher probability to get popular, especially when it comes to the middle level. When it is too small, it has a small probability of becoming popular.

Sentiment Feature. Researchers have analyzed the relationship between sentiment and the final popularity of tweet [50]. Most of them found that the negative sentiment has a higher correlation with the final popularity.

In order to extract the sentiment feature of tweets accurately, we need to do some preprocessing [51], such as removing all the URLs, targets (@username), special twitter words (e.g., 'RT'). Then we adopt the Dynamic Convolutional Neural Network (DCNN) [52] to model the sentiment in twitter, which has a pretty high accuracy. We train the DCNN based on millions of tweet with sentiment labels, and the output of DCNN is between 0 and 1, from extremely negative to extremely positive.

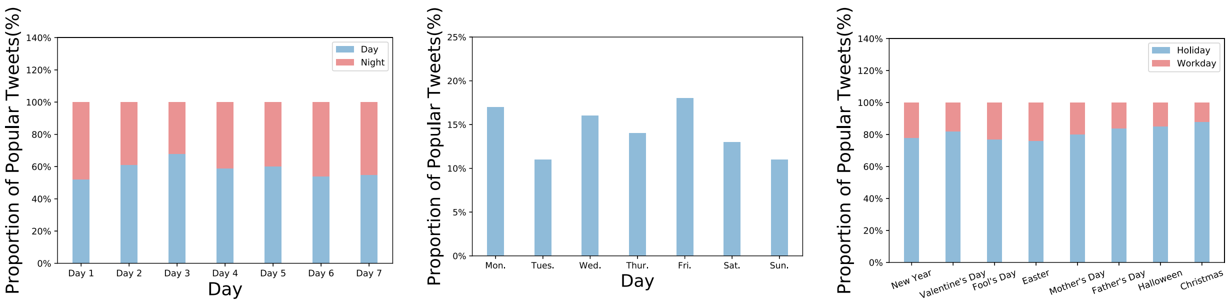
We randomly choose the popular tweets and calculate their sentiment with DCNN. The proportion of popular tweet in each sentiment is as Fig. 4b, which shows that extreme sentiment in tweet will get higher possibility to get more popularity, especially extremely negative sentiment.

Create-Time Weight. Social networks are dominated by users and adapt to users' daily routine, so tweets in different time have different popularities. We randomly choose the popular tweet to calculate the proportion of popular tweets in different creating time, summarized as Fig. 4c. It is obvious that $2\ a.m.{-}3\ a.m.$, $12\ a.m.{-}1\ p.m.$ and $8\ p.m.{-}12\ p.m.$ are the best time periods that have a large proportion of popular tweets. However, there are obviously some periods which are not good time to post a tweet, such as $6\ a.m.{-}10\ a.m$, and $6\ p.m.{-}7\ p.m.$. It is closely related to users' daily schedule. Hence, posting tweets at a suitable time will gain more popularity.

In addition, we consider the temporal layers more specifically in order to detect the disorders. In detail, first, we randomly choose 7 days in dataset to observe the variations of popular tweets in day and night. As shown in Fig. 5a, we find that tweets posted in day have more chance to become popular. Then, we randomly choose one week and detect the trend of popular tweets in the week, summarized as Fig. 5b, that we can observe proportion of popular tweets are higher on Monday and Friday. Lastly, we select 8 holidays to find variations of popular tweets between holiday and the day after holidays, depicted in Fig. 5c. It is intuitive that there are more popular tweets in holidays.

URL Number. Uniform Resource Locator (URL) in tweets is an excellent method that contains more information on the limitation of 140 characters. It always contains three kinds of messages, pictures, videos, and web resource, which can include a wealth of information.

Hashtag Number. With an interesting hashtag, a tweet will attract more users who are interested in this topic. In this way, a tweet with the hashtag may be propagated better.



(a) Day-night (b) Week (c) Holiday

Fig. 5. Single-tweet features analysis part 2.

Atuser Number. A tweet with the atuser will notice mentioned users and the users who are concerned about them, which lead a better spread.

To figure out the exhaustive effect that features make, we use Principal Component Analysis (PCA) and Index Correlation Matrix to obtain the main impact features and remove the slight features. In all, Timestamp of Create Time and URL Number are filtered out due to their low contribution. The main influence factors are as Table 6.

6.5 Baselines

We compare the performance of the proposed methods with some baselines. These methods are as follows:

SVR: Support Vector Regression [53] is a widely used discriminative model, which gets a great performance in various regression tasks including popularity prediction.

LLR: The Logarithmic Linear Regression Model [19] considers strong correlations between early and later times of the logarithmic transformed popularity: $\ln Y(t_r) = \ln r(t_i, t_r) + \ln Y(t_i) + \xi(t_i, t_r)$, where $Y(t_r)$ is the popularity of t_r, $Y(t_i)$ is the popularity of t_i, $r(t_i, t_r)$ represents the popularity from t_i to t_r, and $\xi(t_i, t_r)$ is the bias.

SEISMIC: The stochastic self-exciting point process model [20] uses the information cascades as self-exciting point process and it requires no feature engineering and scales linearly with the number of observed shares of a post. The intensity function is $Y_\infty(t) = Y_t + \alpha_t \frac{p_t(N_t - N_t^e)}{1-\gamma_t p_t n^*}, 0 < \alpha_t, \gamma_t < 1$, where α_t and γ_t are two scaling constants that adjust the final prediction and p_t is the infectiousness parameter. Y_t represents the observed popularity of time t.

PSEISMIC: The personalized self-exciting point process [37] is to model the types of tweet posts via clustering analysis so that the personalized memory kernels for different types can be derived and used for popularity prediction.

Hybrid: The Hybrid [21] uses each social cascade with a marked Hawkes self-exciting point process, and then estimates the content virality. Then the model learns a predictive layer for popularity prediction using a collection of cascade history. The final popularity is computed as $Y_\infty = n + \omega(\frac{A_1}{1-n^*})$, where n is the number of observed popularity and ω is the scaling factor of the training Random Forest for the expected number of future events in a retweet cascade. A_1 is the expected number of total popularity.

6.6 Evaluation Metrics

Two sets of evaluation metrics are used for our task, containing the regression task which is to predict the trends and the classification task which is to predict the popular status.

For the quantitative prediction, we measure the performance of the proposed models using the following metrics:

- *MdAPE* is the Median Absolute Percentage Error, which is the median value of *APE* in this sample. For each tweet w at time t, the formulation of APE is $APE(t) = \frac{|Y(t)-G(t)|}{G(t)}$
- *MAPE* is the Mean Absolute Percentage Error. The size of the sample is n, the *MAPE* of this sample at t can be defined as $MAPE(t) = \frac{1}{n}\sum_{i=1}^{n} \frac{|Y(t)-G(t)|}{G(t)}$
- *RMSE* is the Root Mean Square Error. The size of the sample is n, the *RMSE* of this sample can be represented as $RMSE(t) = \sqrt{\frac{1}{n}(Y(t) - G(t))^2}$

Where $G(t)$ is the real popularity of time t. Specifically, if t is equal to T $(0 < t \leq T)$, the calculated popularity count $Y(t)$ is the real popularity $G(t)$. Thus we can use $G(t)$ to represent the final count of a single tweet, and compare it with the approximate final count $Y(t)$.

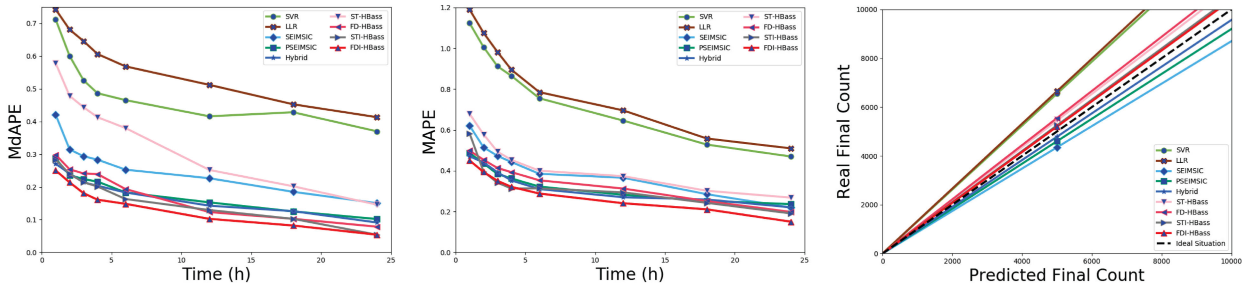
For the qualitative prediction, after calculating the popular threshold γ, then a tweet with $Y(T) \geq \gamma$ is regarded as popular and labeled positive. Otherwise, it is labeled negative. To measure the performance of the qualitative prediction, we use the metrics as follows:

- *Precision* is defined as $Precision = \frac{TP}{TP+FP}$
- *Recall* is defined as $Recall = \frac{TP}{TP+FN}$
- *F-score* is defined as the harmonic mean of the Precision and Recall. A high value of F-score indicates that both Precision and Recall are reasonably high. It represents as $F{-}score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$

The *TP*, *FN* and *FP* represent the number of true positive instances, the number of false negative instances and the number of false positive instances, respectively.

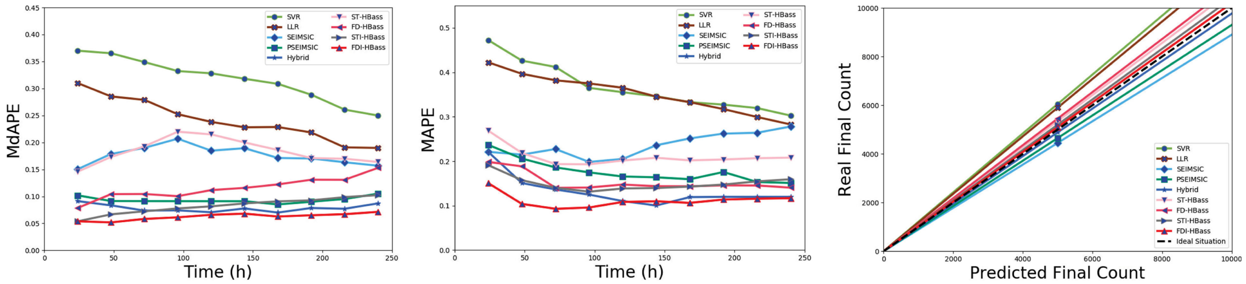
TABLE 6
The Main Features

Features	Detail features	Features	Detail features
User Feature	Follower Number Tweet Number Favorite Number Following Number	Single -tweet Feature	Create-time Weight Content Information Sentiment Feature Hashtag Number Atuser Number



(a) MdAPE (b) MAPE (c) The predicted final count VS real final count

Fig. 6. The results for the early stage.



(a) MdAPE (b) MAPE (c) The predicted final count VS real final count

Fig. 7. The results for the stable stage.

6.7 Quantitative Prediction

In contrast to traditional machine learning methods, which usually have a training set to train the parameters and a test set to evaluate trained models, our HBass divides the whole life cycle of each tweet into training part and predicting part. We set the earlier time data as the training set to train the parameters and each time stage data as the test set to evaluate trained models. This is to say, at the early stage, we use the features before $1h$, and to evaluate the performance of our model at the early stage. At the stable stage, we adopt the data before $24h$ to evaluate the performance at the stable stage. Therefore, our sample points can well test the whole life cycle of a tweet.

Figs. 6a and 6b respectively show the Median APE and Mean APE in the early stage. The results for the stable stage are as Figs. 7a and 7b. We test 8 time points at the early stage and 10 time points at the stable stage. At the early stage, ST-HBass and FD-HBass perform absolutely better than SVR and LLR, but worse than PSEISMIC and Hybird as a whole. FD-HBass performs better than SEISMIC, but ST-HBass performs worse than SEISMIC. When they combine the Interaction Enhancement, their performance is improved. The FDI-HBass completely performs better than PSEISMIC and Hybird, and STI-HBass performs better in some test points but worse than PSEISMIC and Hybird as a whole. Stable stage has the similar performance with early stage. Particularly, FDI-HBass gets the best result in both Median and Mean APE at early and stable stage. Our four HBass change little during the whole period, which indicates our model is good at predicting the trend of tweets that has continuous and big changes, while these tweets are usually more popular.

Moreover, we can get roughly similar results from *RMSE*. In detail, *RMSE* can enlarge those gaps between different models. We list the *RMSE* of different models in two stages at $t = 1h$, $t = 2h$, $t = 4h$, $t = 12h$, $t = 48h$, $t = 72h$, $t = 120h$ and $t = 240h$ as Table 7. The FDI-HBass gets the lowest error.

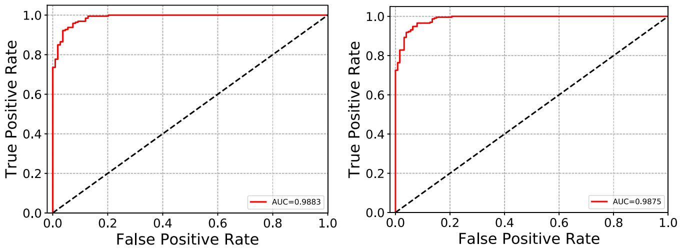
In order to measure the performance of those models, we select some representative points and plot the scatter plot with the predicted final popularity count and the real final popularity count. We further use linear regression to fit their relationship with different stages, which are illustrated as in Figs. 6c and 7c. The dash line indicates the ideal situation with the slope 1. The distribution of those fitting lines is mostly above on the dash line, which represents that the predicting results are smaller than the real situation. On the other hand, the results of SEISMIC, Hybrid and PSEISMIC are larger than real one overall, which are below the dash line. The fitting line which is closer to the dash line can represent a better performance in predicting the trends as a whole. Generally speaking, the Hybrid and FDI-HBass are the most closely model to the ideal situation. However, they have different trends. Hybrid model always gets a more popular situation with the slope lower than 1, while FDI-HBass usually obtains a more conservative result with the slope greater than 1. As for our HBass, most of predicted values are close to real values. When the final popular count is less than γ, our model performs best. As the real

TABLE 7
RMSE of Different Methods

Model	The early stage				The stable stage			
	1h	2h	4h	12h	48h	72h	120h	240h
SVR	71.73	146.59	184.55	214.82	223.31	234.27	253.54	275.66
LLR	74.53	149.11	189.77	201.57	215.47	227.64	248.74	268.82
SEISMIC	19.47	31.91	42.89	54.45	58.76	63.53	67.23	69.45
PSEISMIC	12.27	16.32	18.74	22.33	27.33	31.22	36.75	42.89
Hybrid	10.23	15.39	17.52	20.36	24.22	26.55	**27.57**	32.43
ST-HBass	28.67	39.62	49.06	57.79	62.23	65.44	69.31	72.89
FD-HBass	14.23	20.91	22.35	25.56	29.33	34.37	37.31	41.67
STI-HBass	12.11	17.17	19.43	24.15	29.22	32.72	34.42	36.41
FDI-HBass	**8.54**	**12.49**	**14.96**	**17.56**	**22.67**	**24.79**	28.33	**29.56**

TABLE 8
The Performances of Different Methods

Model	T = 24h			T = 240h		
	Precision	Recall	F-score	Precision	Recall	F-score
SVR	0.7432	0.3128	0.4403	0.7763	0.3642	0.4958
LLR	0.2863	0.8364	0.4266	0.4632	0.8726	0.6052
SEISMIC	0.8773	0.8631	0.8701	0.9321	0.9193	0.9257
PSEISMIC	0.9231	0.9318	0.9274	0.9357	0.9369	0.9363
Hybrid	0.9314	**0.9588**	0.9449	0.9473	**0.9518**	0.9495
ST-HBass	0.8564	0.8233	0.8395	0.9316	0.8936	0.9122
FD-HBass	0.9419	0.7864	0.8572	0.9621	0.8012	0.8743
STI-HBass	0.9286	0.8756	0.9013	0.9554	0.8382	0.8930
FDI-HBass	**0.9683**	0.9403	**0.9541**	**0.9788**	0.9244	**0.9508**



(a) T=24h (b) T=240h

Fig. 8. The ROC curve of FDI-HBass in two stages.

popularity count becomes larger, our prediction emerges its conservative. There is a high probability that our predicted value is smaller than the real ones. However, it does not affect the precision of predicting popularity.

In all, PSEIMSIC and Hybird can get a pretty good result. However, it uses hard-collected features that is difficult for both users and enterprises to capture and track. Generally, our HBass model only needs the easy-collected features and few earlier features, then we can get the popularity at a high accuracy at any time $t \geq 1h$ in all tweet's life cycle.

6.8 Qualitative Prediction

With the suitable threshold γ and the final popularity count $Y(T)$, we can smoothly get the popular status. Table 8 shows the performance of qualitative prediction by different methods. Whenever popular determined time is $T = 24h$ or $T = 240h$, our HBass has a pretty good performance in classification. Specially, FDI-HBass gets the best *Precision*, 3.5 percent better than Hybrid and almost 5.6 percent better than PSEISMIC, and best *F-score*, 0.5 percent better than Hybrid and almost 3.0 percent better than PSEISMIC. However, in terms of *Recall*, the performance of those HBass are not desired, which represents some popular tweets may be mistaken as unpopular tweets. In addition, we illustrate the Receiver Operating Characteristic (ROC) curve and AUC of FDI-HBass in Fig. 8, which shows the effectiveness of this classifier. As a whole, our HBass has the excellent performance to achieve qualitative prediction without additional models.

6.9 Case Study: Popularity Prediction

We conduct a case study of popularity prediction with FDI-HBass model, which performs better than other methods in most cases as we discussed above. We set $T = 240h$ and threshold $\gamma = 172$ according to the Parameter Settings. As

TABLE 9
The Result of Prediction Cases With FDI-HBass

Topic	Tweet	Real popularity	Predicted popularity
Gas explosion in Oklahoma	Example 1	282 (+)	271 (+)
	Example 2	151 (−)	149 (−)
School shooting in Kentucky	Example 3	176 (+)	162 (−)
	Example 4	105 (−)	101 (−)

(+ denotes popular status while − denotes unpopular status.)

shown in Table 9, we select two topics in our datasets, each contains two example tweets. One of them is popular tweet in the end while another is not.

As a result, the predicted popular status of Example 1, 2, and 4 are correct. However, the model fails to predict the final status of Example 3. It should be popular tweet rather than unpopular tweet. After data analysis, we find that the popular count of Example 3 increases slowly over time. It makes difficult for prediction because FDI-HBass focuses on the effect of features rather than temporal influence.

7 Conclusion

In this paper, we design the Heterogeneous Bass model (*HBass*) which contains two varieties, namely Spatial-Temporal Heterogeneous Bass Model (*ST-HBass*) and Feature-Driven Heterogeneous Bass Model (*FD-HBass*), to predict the popularity of a single tweet. We also propose an Interaction Enhancement to measure the competition and cooperation relationship of different tweets with the common topic. We further design an clustering method to bound the popularity threshold based on real world dataset.

Our experiments use real-world Twitter data to validate the efficiency and accuracy of our model in quantitative prediction, with less absolute percent error. For qualitative prediction, our model gets the best *Precision* and *F-score*, which shows that our models have a better classification detection.

Acknowledgments

This work was supported by the National Key R&D Program of China [2018YFB1004703]; the National Natural Science Foundation of China [61872238, 61972254, 61672353]; the Shanghai Science and Technology Fund [17510740200]; the CCF-Huawei Database System Innovation Research Plan [CCF-Huawei DBIR2019002A]; the Huawei Innovation Research Program [HO2018085286]; the State Key Laboratory of Air Traffic Management System and Technology [SKLATM20180X]; and the Tencent Social Ads Rhino-Bird Focused Research Program.

References

[1] J. Golbeck, C. Robles, and K. Turner, "Predicting personality with social media," in *Proc. ACM CHI Conf. Human Factors Comput. Syst.*, 2011, pp. 253–262.
[2] K. Lerman and T. Hogg, "Using a model of social dynamics to predict popularity of news," in *Proc. ACM Int. Conf. World Wide Web*, 2010, pp. 621–630.
[3] X. Zhang, X. Chen, Y. Chen, S. Wang, Z. Li, and J. Xia, "Event detection and popularity prediction in microblogging," *Elsevier Neurocomputing*, vol. 149, pp. 1469–1480, 2015.

[4] S. Asur and B. Huberman, "Predicting the future with social media," in *Proc. IEEE Int. Conf. Web Intell. Intell. Agent Technol.*, 2010, pp. 492–499.
[5] Y. Huang, S. Zhou, K. Huang, and J. Guan, "Boosting financial trend prediction with twitter mood based on selective hidden Markov models," in *Proc. Springer Int. Conf. Database Syst. Advanced Appl.*, 2015, pp. 435–451.
[6] Q. Zhang, Y. Gong, Y. Guo, and X. Huang, "Retweet behavior prediction using hierarchical dirichlet process," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 403–409.
[7] Q. Zhang, Y. Gong, J. Wu, H. Huang, and X. Huang, "Retweet prediction with attention-based deep neural network," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 75–84.
[8] J. Bian, Y. Yang, and T. S. Chua, "Predicting trending messages and diffusion participants in microblogging network," in *Proc. ACM Int. Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 537–546.
[9] P. Cui, S. Jin, L. Yu, F. Wang, W. Zhu, and S. Yang, "Cascading outbreak prediction in networks:a data-driven approach," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 901–909.
[10] 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 *Proc. IEEE Int. Conf. Social Comput.*, 2010, pp. 177–184.
[11] M. Jenders, G. Kasneci, and F. Naumann, "Analyzing and predicting viral tweets," in *Proc. ACM Int. Conf. World Wide Web*, 2013, pp. 657–664.
[12] R. Bandari, S. Asur, and B. A. Huberman, "The pulse of news in social media: Forecasting popularity," in *Proc. Assoc. Advancement Artif. Intell. Int. AAAI Conf. Web Social Media*, 2012, pp. 26–33.
[13] A. Kupavskii *et al.*, "Prediction of retweet cascade size over time," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 2335–2338.
[14] J. Li, J. Qin, T. Wang, Y. Cai, and H. Min, "A collaborative filtering model for personalized retweeting prediction," in *Proc. Springer Int. Conf. Database Syst. Advanced Appl.*, 2015, pp. 122–134.
[15] A. N. H. Man and M. L. Khodra, "Predicting information cascade on twitter using support vector regression," in *Proc. Int. Conf. Data Softw. Eng.*, 2014, pp. 1–6.
[16] M. Yang, K. Chen, Z. Miao, and X. Yang, "Cost-effective user monitoring for popularity prediction of online user-generated content," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 944–951.
[17] H. Zhao, G. Liu, C. Shi, and B. Wu, "A retweet number prediction model based on followers' retweet intention and influence," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 952–959.
[18] R. Lemahieu, S. Van Canneyt, C. De Boom, and B. Dhoedt, "Optimizing the popularity of twitter messages through user categories," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 1396–1401.
[19] P. Bao, H.-W. Shen, J. Huang, and X.-Q. Cheng, "Popularity prediction in microblogging network: A case study on sina weibo," in *Proc. ACM Int. Conf. World Wide Web*, 2013, pp. 177–178.
[20] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, "SEISMIC: A self-exciting point process model for predicting tweet popularity," in *Proc. ACM Knowl. Discovery Data Mining*, 2015, pp. 1513–1522.
[21] S. Mishra, M.-A. Rizoiu, and L. Xie, "Feature driven and point process approaches for popularity prediction," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1069–1078.
[22] T. Zaman, E. B. Fox, and E. T. Bradlow, "A Bayesian approach for predicting the popularity of tweets," *Ann. Appl. Statistics*, vol. 8, no. 3, pp. 1583–1611, 2014.
[23] B. Peng, "Modeling and predicting popularity dynamics via an influence-based self-excited Hawkes process," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1897–1900.
[24] Y. Li, Z. Feng, H. Wang, S. Kong, and L. Feng, "*ReTweetP*: Modeling and predicting tweets spread using an extended susceptible-infected-susceptible epidemic model," in *Proc. Springer Int. Conf. Database Syst. Advanced Appl.*, 2013, pp. 454–457.
[25] L. Yu, P. Cui, F. Wang, C. Song, and S. Yang, "From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 559–568.
[26] J. Y. Chin, S. S. Bhowmick, and A. Jatowt, "TOTEM: Personal tweets summarization on mobile devices," in *Proc. ACM Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 1305–1308.
[27] X. Jin, A. Gallagher, L. Cao, J. Luo, and J. Han, "The wisdom of social multimedia: Using flickr for prediction and forecast," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 1235–1244.
[28] F. M. Bass, "A new product growth for model consumer durables," *Manage. Sci.*, vol. 15, no. 5, pp. 215–227, 1969.
[29] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding topic-sensitive influential twitterers," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2010, pp. 261–270.
[30] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang, "TopicSketch: Real-time bursty topic detection from twitter," in *Proc. IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2216–2229, Aug. 2016.
[31] T.-A. N. Pham, X. Li, G. Cong, and Z. Zhang, "A general recommendation model for heterogeneous networks," in *Proc. IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3140–3153, Dec. 2016.
[32] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical approximate k nearest neighbor queries with location and query privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1546–1559, Jun. 2016.
[33] X. Zhou, X. Liang, H. Zhang, and Y. Ma, "Cross-platform identification of anonymous identical users in multiple social media networks," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 411–424, Feb. 2016.
[34] B. Jiang *et al.*, "Retweeting behavior prediction based on one-class collaborative filtering in social networks," in *Proc. ACM Int. Conf. Res. Develop. Inf. Retrieval*, 2016, pp. 977–980.
[35] Z.-H. Deng, X. Gong, F. Jiang, and I. W. Tsang, "Effectively predicting whether and when a topic will become prevalent in a social network," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 210–216.
[36] P. J. McParlane, Y. Moshfeghi, and J. M. Jose, "Nobody comes here anymore, it's too crowded; predicting image popularity on flickr," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2014, pp. 385–392.
[37] H.-Y. Chen and C.-T. Li, "PSEISMIC: A personalized self-exciting point process model for predicting tweet popularity," in *Proc. ACM Int. Conf. Big Data*, 2017, pp. 2710–2713.
[38] W. Ding, L. Guo, L. Guo, X. Hu, R. Yan, and T. He, "Video popularity prediction by sentiment propagation via implicit network," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 1621–1630.
[39] M. He, Y. Ge, L. Wu, E. Chen, and C. Tan, "Predicting the popularity of *DanMu*-enabled videos: A multi-factor view," in *Proc. Springer Int. Conf. Database Syst. Advanced Appl.*, 2016, pp. 351–366.
[40] X. He, M. Gao, M. Y. Kan, Y. Liu, and K. Sugiyama, "Predicting the popularity of web 2.0 items based on user comments," in *Proc. ACM Int. Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 233–242.
[41] H. Pinto and J. M. Almeida, "Using early view patterns to predict the popularity of youtube videos," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2013, pp. 365–374.
[42] W. Chaokun, X. Xin, and S. Jingwen, "When to make a topic popular again? a temporal model for topic rehotting prediction in online social networks," in *Proc. IEEE Trans. Signal Inf. Process. Over Netw.*, 2018, pp. 202–216.
[43] P. Bao, H.-W. Shen, X. Jin, and X.-Q. Cheng, "Modeling and predicting popularity dynamics of microblogs using self-excited Hawkes processes," in *Proc. ACM Int. Conf. World Wide Web*, 2015, pp. 9–10.
[44] M.-A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. Van Hentenryck, "Expecting to be hip: Hawkes intensity processes for social media popularity," in *Proc. ACM Int. Conf. World Wide Web*, 2017, pp. 735–744.
[45] H. Wenjian, S. K. Kumar, X. Fanyi, H. Jinyoung, C. C.-Nee, and L. Y. Jae, "Using early view patterns to predict the popularity of youtube videos," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2018, pp. 1–9.
[46] M. X. Hoang, X. H. Dang, X. Wu, Z. Yan, and A. K. Singh, "GPOP: Scalable group-level popularity prediction for online content in social networks," in *Proc. ACM Int. Conf. World Wide Web*, 2017, pp. 725–733.
[47] S. Kong, Q. Mei, L. Feng, F. Ye, and Z. Zhao, "Predicting bursts and popularity of hashtags in real-time," in *Proc. ACM Int. Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 927–930.
[48] D. Strang and N. B. Tuma, "Spatial and temporal heterogeneity in diffusion," *Amer. J. Sociology*, vol. 99, pp. 614–639, 1993.
[49] S. Wang, Z. Yan, X. Hu, S. Y. Philip, and Z. Li, "Burst time prediction in cascades," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 325–331.
[50] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in twitter events," *J. Assoc. Inf. Sci. Technol.*, vol. 62, no. 2, pp. 406–418, 2011.
[51] M. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *Proc. IEEE Int. Conf. Comput. Commun. Netw. Technol.*, 2013, pp. 1–5.
[52] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 1–11.

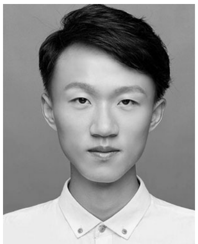
[53] N. Rosenfeld, M. Nitzan, and A. Globerson, "Discriminative learning of infection models," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2016, pp. 563–572.

[54] X. L. Dong, A. Kementsietsidis, and W.-C. Tan, "A time machine for information: Looking back to look forward," *ACM SIGMOD Record*, vol. 45, no. 2, pp. 23–32, 2016.

[55] S. Hosseini, H. Yin, X. Zhou, S. Sadiq, M. R. Kangavari, and N.-M. Cheung, "Leveraging multi-aspect time-related influence in location recommendation," *World Wide Web*, vol. 22, pp. 1001–1028, 2019.

[56] Y. Shen, "Forecasting twitter topic popularity using bass diffusion model and machine learning," Ph.D. dissertation, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology (MIT), Cambridge, MA, 2015.

Xiaofeng Gao received the BS degree in information and computational science from Nankai University, China, in 2004, the MS degree in operations research and control theory from Tsinghua University, China, in 2006, and the PhD degree in computer science from the University of Texas at Dallas, in 2010. She is currently an associate professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include wireless communications, data engineering, and combinatorial optimizations. She has published more than 160 peer-reviewed papers in the related area, including well-archived international journals such as the *IEEE Transactions on Computers*, *IEEE Transactions on Parallel Distributed Systems*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Knowledge Data Engineering*, and also in well-known conference proceedings such as IJCAI, SIGKDD, and INFOCOM. She has served on the editorial board of *Discrete Mathematics, Algorithms and Applications*. She is a member of the IEEE.



Zuowu Zheng received the BE degree in computer science and technology from the Ocean University of China, in 2018. He is currently working toward the doctor degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include social network analysis, event prediction, and data science.

Quanquan Chu received the BS degree in computer science and technology from Anhui University, in 2016, and the MS degree in computer science and engineering from Shanghai Jiao Tong University, in 2019. He is currently a software engineer at Qutoutiao. His research interests include data communication and engineering and social network analysis.

Shaojie Tang received the PhD degree in computer science from the Illinois Institute of Technology, in 2012. He is currently an assistant professor of the Naveen Jindal School of Management, University of Texas at Dallas. His research interest includes social networks, mobile commerce, game theory, e-business and optimization. He received the best paper awards in ACM MobiHoc 2014 and IEEE MASS 2013. He also received the ACM SIGMobile Service Award in 2014. He served in various positions (as chairs and TPC members) at numerous conferences, including ACM MobiHoc, IEEE INFOCOM, IEEE ICDCS, and IEEE ICNP. He is a member of the IEEE.

Guihai Chen received the BS degree from Nanjing University, in 1984, the ME degree from Southeast University, in 1987, and the PhD degree from the University of Hong Kong, in 1997. He is a distinguished professor of Shanghai Jiaotong University, China. He had been invited as a visiting professor by many universities including the Kyushu Institute of Technology, Japan, in 1998, University of Queensland, Australia, in 2000, and Wayne State University, USA, during September 2001 to August 2003. He has a wide range of research interests with focus on sensor network, peer-to-peer computing, high performance computer architecture, and combinatorics. He has published more than 200 peer-reviewed papers, and more than 120 of them are in well-archived international journals such as the *IEEE Transactions on Knowledge Data Engineering*, *IEEE Transactions on Parallel Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Mobile Computing*, and also in well-known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS, and ICDCS. He is a member of the IEEE.

Qianni Deng received the BS degree in computer technology from Shanghai Jiao Tong University, China, and the MS degree and PhD degrees in computer science from Shanghai Jiao Tong University. She is an associate professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include parallel computing and data intensive super computing.