Taxonomy and Evaluation for Microblog Popularity Prediction

15

XIAOFENG GAO and ZHENHAO CAO, Shanghai Jiao Tong University
SHA LI, University of Illinois at Urbana-Champaign
BIN YAO and GUIHAI CHEN, Shanghai Jiao Tong University
SHAOJIE TANG, University of Texas at Dallas

As social networks become a major source of information, predicting the outcome of information diffusion has appeared intriguing to both researchers and practitioners. By organizing and categorizing the joint efforts of numerous studies on popularity prediction, this article presents a hierarchical taxonomy and helps to establish a systematic overview of popularity prediction methods for microblog. Specifically, we uncover three lines of thoughts: the feature-based approach, time-series modelling, and the collaborative filtering approach and analyse them, respectively. Furthermore, we also categorize prediction methods based on their underlying rationale: whether they attempt to model the motivation of users or monitor the early responses. Finally, we put these prediction methods to test by performing experiments on real-life data collected from popular social networks Twitter and Weibo. We compare the methods in terms of accuracy, efficiency, timeliness, robustness, and bias.

As far as we are concerned, there is no precedented survey aimed at microblog popularity prediction at the time of submission. By establishing a taxonomy and evaluation for the first time, we hope to provide an in-depth review of state-of-the-art prediction methods and point out directions for further research. Our evaluations show that time-series modelling has the advantage of high accuracy and the ability to improve over time. The feature-based methods using only temporal features performs nearly as well as using all possible features, producing average results. This suggests that temporal features do have strong predictive power and that power is better exploited with time-series models. On the other hand, this implies that we know little about the future popularity of an item before it is posted, which may be the focus of further research.

CCS Concepts: • **General and reference** → **Evaluation**; • **Information systems** → *Social networks*; • **Applied computing** → *Sociology*;

Additional Key Words and Phrases: Social network, popularity prediction, evaluation, taxonomy

This work is supported by the National Key R&D Program of China 2018YFB1004703, the National Natural Science Foundation of China (61872238, 61672353, 61872235, 61729202, 61832017, and U1636210), the Shanghai Science and Technology Fund (17510740200), the Huawei Innovation Research Program (HO2018085286), the State Key Laboratory of Air Traffic Management System and Technology (SKLATM20180X), and Tencent Social Ads Rhino-Bird Focused Research Porgram.
Authors' addresses: X. Gao (Corresponding author), Shanghai Jiao Tong University, Shanghai 200240, China; email: gao-xf@cs.sjtu.edu.cn; Z. Cao, Shanghai Jiao Tong University, F1503023, Shanghai, 200240; email: hazelnut@sjtu.edu.cn; S. Li, University of Illinois at Urbana-Champaign, 201 N Goodwin Ave, Urbana, Illinois 61801, USA; email: shal2@illinois.edu; B. Yao, Shanghai Jiao Tong University, Room 3-505, SEIEE Building, 800 Dong Chuan Road, Shanghai, 200240, P.R. China; email: yaobin@cs.sjtu.edu.cn; G. Chen, Shanghai Jiao Tong University, SEIEE 3-422, 800 Dongchuan Road, Minhang District, Shanghai 200240, P.R. China; email: gchen@cs.sjtu.edu.cn; S. Tang, University of Texas at Dallas, 800 West Campbell Road, SM 33, Richardson, TX 75080, U.S; email: tangshaojie@gmail.com.

1556-4681/2019/03-ART15 $15.00
https://doi.org/10.1145/3301303

ACM Reference format:
Xiaofeng Gao, Zhenhao Cao, Sha Li, Bin Yao, Guihai Chen, and Shaojie Tang. 2019. Taxonomy and Evaluation for Microblog Popularity Prediction. *ACM Trans. Knowl. Discov. Data* 13, 2, Article 15 (March 2019), 40 pages.
https://doi.org/10.1145/3301303

1 INTRODUCTION

How does an idea or product gain popularity through social diffusion? This question has been intriguing to researchers as well as social activists and marketing personnel. Social network platforms give us the chance to track the diffusion of user-generated content (UGC), which we refer to as *items*, at a microscopic level and provide information about the participants and their relationships. With this newly obtained data, we can now build more precise models for the adoption of items, and even predict the popularity of an item before the actual adoption takes place. This prediction ability would allow for the discovery of potential hot items or improvements on viral marketing strategies.

The *popularity* of an item can be measured in many ways such as the number of views or the number of 'likes' it received. In this article, we measure an item's popularity by using the number of times it has been reposted. Reposting is the critical mechanism in information diffusion. By reposting from his/her friends, a user creates a new personal status that will appear on his/her followers feeds, enlarging the audience of the item, possibly evoking a chain of repost events referred to as *cascades*. Some studies choose to predict popularity at the time of posting, however, a more widely adopted approach includes peeking into the initial stages of diffusion, suggesting that early patterns are tightly connected to the eventual fate of the item.

The prediction problem itself can either be defined as predicting the final repost count or setting up a threshold for identifying the popular items. In choosing the threshold, one natural way is to set a certain percentile for distinguishing popular items. Some other works have formulated it as a balanced classification problem, predicting whether a cascade will double in the future [16]. Other variations of the popularity prediction problem are either microscopic by predicting whether a specific user will adopt an item [67, 99] or continuous in time by predicting the size of the cascade along the time domain [94]. Another closely related problem is outbreak time prediction which can be solved by recursively testing the size at different times. We put our focus on final count and two-way classification as they are the most widely used formulations of the popularity prediction problem in related literature.

In the past few years, a series of efforts has been devoted to predicting the popularity of an item in social networks. However, most papers describe new methods, which they show work well under their set of assumptions, but none of them explains how well their scheme can perform under different settings. Hence, in this article, we set out to classify the proposed models and perform a fair and rigid evaluation under a unified testing scheme. Our testing scheme covers accuracy, timeliness, robustness, efficiency, and bias.

We establish a taxonomy of prediction methods with the first level of the hierarchy being three general categories: *feature-based methods*, *time-series methods*, and *collaborative filtering methods*. Feature-based methods emphasize devising effective features and adopt classical machine-learning models for prediction. As social networks involve heterogeneous data, features are further divided into content-related, user-related, structural, and temporal. User responses can also be seen as a special type of feature as in [17]. Time-series methods borrow ideas from financial modelling and epidemiology, using mainly the repost times to model the growth of cascades. Based on whether the model is stochastic or deterministic, we identify two sub-categories: point process models and compartment models. Collaborative filtering methods represent cascades as a user-item repost matrix or tensor. We infer a user's repost actions from her history and the actions of similar users.

We also make an attempt to reveal why these prediction methods work. We categorize prediction methods from another dimension, namely *motivation-oriented* or *monitor-based*. Motivation can either be internal or external. Internal motivation cannot be directly observed from the network, thus we leverage the effect of homophily, the phenomenon that similar users perform similar actions, to model internal motivation. External influence, e.g., how our friends influence our decisions, is captured by features such as the number of followers or the authority score. Monitor-based prediction refers to the 'peeking' strategy, which uses the early response towards an item as the basis for prediction. They exploit patterns in the growth of a cascade to predict the final size. Time-series methods can be seen as a typical example of monitor-based prediction. Other methods tend to be a mixture of modelling motivation and monitoring response.

We perform a unified evaluation of 14 proposed prediction methods, including 9 feature-based methods, 4 time-series methods, as well as 1 collaborative filtering method, covering all categories in our taxonomy. Our dataset consists of two parts: a Twitter dataset with 2 million microblogs and a Weibo dataset with 0.3 million microblogs. We evaluate the accuracy of prediction methods at 24 time points using two sets of metrics for different formulations of the problem. Apart from accuracy, we also assess the methods in terms of timeliness, robustness, efficiency, and bias. We discover that while time-series models perform well in accuracy, they suffer from the existence of extreme values and their dependency on observation. Feature-based methods are the most robust ones and are not influenced by the time of prediction. Collaborative filtering methods have the tendency to underestimate the size of cascades due to the sparsity problem, but their performance improve as a larger proportion of the cascade is observed.

Our contribution to popularity prediction in social networks is three-fold as follows:

- We construct a taxonomy of proposed methods for popularity prediction, categorizing them into feature-based, time-series-based, and collaborative filtering methods. In the process, we uncover many correlations between these methods.
- We analyse the prediction methods according to the underlying rationale, especially the modelling of sociological concepts homophily and influence. This serves as a reference for the modelling and prediction of social phenomena.
- Apart from accuracy metrics such as absolute error and percentage error, we also evaluate prediction methods in terms of timeliness, robustness, efficiency, and bias. Based on such criteria, we discuss which prediction methods are recommendable and some possible improvements.

The rest of this article is organized as follows. We formally describe the prediction problem in Section 2. In Sections 3–5, we introduce feature-based methods, time-series-based methods, and collaborative filtering methods, respectively. In Section 6, we categorize the mentioned methods by the rationale of prediction into motivation-oriented methods or monitor-based methods. Sections 7–9 show our experiment setup and evaluation results. We conclude our findings in Section 10.

2 PROBLEM OVERVIEW

We measure *popularity* by the number of reposts an item receives. Through reposting, users spread the content to his or her friends, possibly evoking a chain of repost events referred to as *cascades*. We denote the number of reposts of an item i by time t as R_i^t. When no time is specified, we refer to the final popularity count R_i^∞ with R_i. We denote the popularity of an item i by P_i.

The popularity predication problem can be formulated in two ways: regression and classification. The regression formulation aims at predicting the final repost count R_i^∞ of an item. However,

Table 1. Notation List

Symbol	Description
P_i	The popularity of item i
R_i^t	The repost count of item i by time t
R_i^∞/R_i	The final repost count of the item i
t	Time of prediction
t_u	The repost time of the u-th user
N_u	Total number of the users (of the network)
G_f	The follow relationship graph
G_r	The mention relationship graph of reposter
$\lambda(t)$	Conditional intensity function
$N(t)$	Repost count

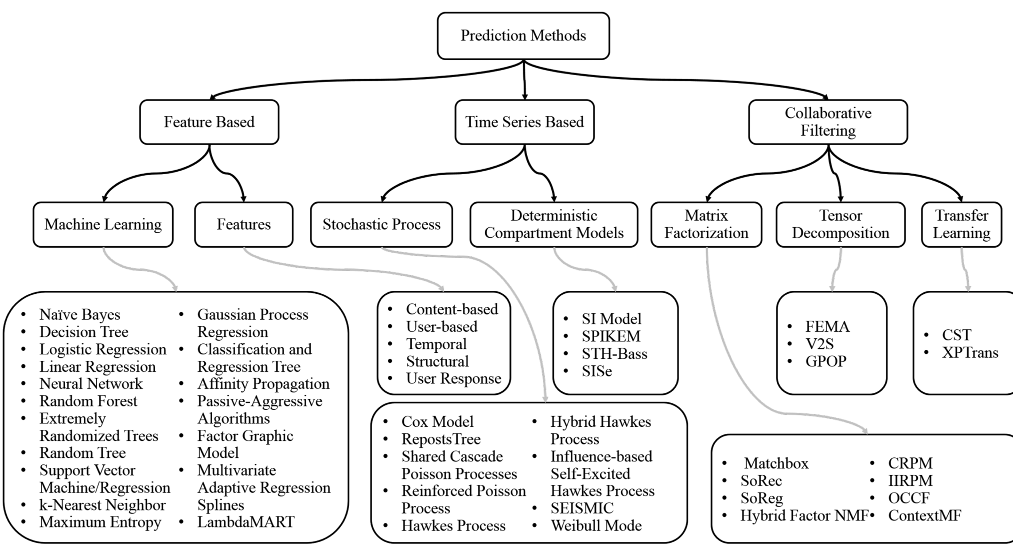

Fig. 1. Hierarchy of prediction methods.

in many applications, the exact count is unnecessary—we only need to distinguish the popular ones from the obscure. This leads to the classification formulation of the problem, which aims at predicting whether an item will receive more reposts than a certain threshold τ.

For the regression problem, we have

$$P_i = R_i.$$

For the classification problem, we have the following formula:

$$P_i = \begin{cases} 1, & \text{if } R_i > \tau, \\ 0, & \text{otherwise.} \end{cases}$$

Recurrent mathematical notations are listed in Table 1.

We show our taxonomy of prediction methods in Figure 1. We primarily divide prediction methods into feature-based methods, time-series-based methods, and collaborative filtering methods. For feature-based methods, as the focus is mainly on extracting features, multiple classifiers may be used for the same set of features. Time-series-based methods can be further divided into compartment models and point process models based on whether the model is stochastic or deterministic. Both can predict the size of a cascade at any time during diffusion. Collaborative filtering methods attempt to predict popularity on a microscopic level, from the prospective of predicting user actions.

Studies on microblog popularity prediction are shown in Table 2. Some of them generalized the task to popularity prediction of any UGC [1], or focus on citation popularity prediction [13, 102], but the methods are applicable to microblog problem. *Level* in Table 2 refers to the level of analysis, specifying whether the method solves the problem from the individualistic, the holistic, or the intermediate perspectives [11]. Single actor (actor level), actor-pair (dyadic level), actor-triplet (triadic level), and egocentric network [99] are typical analysis units in micro-level studies. Macro-level approaches, on the other hand, are concerned with the general outcomes over the entire population. Meso-level methods focus on the population size falling between micro- and macro-levels, usually entailing primary clustering or partitioning to utilize group-level attributes [28].

3 FEATURE-BASED METHODS

Many researchers have studied the factors affecting content popularity in social networks.

One of the pioneers is Suh et al. [78]. Characterizing a microblog with content features and user features, they employed PCA and generalized linear model to find out what might affect the repost probability of a microblog. Later researchers who targeted the prediction problem also adopted the combination of feature engineering and general machine-learning methods. This grew to be one of the paradigms.

The features used for popularity prediction can be divided into four categories: content-related, user-related, structural, and temporal. We will introduce these four types of features in order, paying special attention to those that have been reported to be effective in experimental evaluation. We can also treat user responses as a special type of features as in [17]. A complete category of features is shown in Figure 2.

Content-related. As the name suggests, content-related features are textual features derived from the original post. Some of these features directly come from the message, such as the number of hashtags, URLs, or mentions [78]. More sophisticated features, such as sentiment and topic [16, 47, 65], are extracted using natural language processing techniques.

Naveed et al. [65] were one of the first to use sentiment for popularity prediction. In 2013, Jenders et al. [34] brought up some other new sentiment features such as emotional divergence, and analysed their relationship with microblog popularity. Thelwall et al. [80] found that the negative sentiment has a higher correlation with item popularity. The topic can either be defined as the hashtags attached to the microblog or extracted using topic models. Apart from sentiment and topic, some other features like novelty [69] and degree of self-disclosure [95] have also been used as hyper-features.

Some works showed that content features are effective in the popularity prediction task. Jenders et al. [34] reported that the content features, especially the sentiment features, have predictive power, though they are less effective than the structural features. Petrovic et al. [69], Suh et al. [78], and Cheng et al. [16] agree that content features are weak predictors of how widely disseminated a piece of content would become. As a result, content features are usually seen as auxiliary in popularity prediction.

We use the following features in our implementation: individual sentiment value [34], emotional divergence [70], tweet length [69], number of hashtags [78], number of mentions, and total number of special signals [54].

Table 2. Literature on Popularity Prediction

Article	Type	Classification/ regression	Method	Level	Outcome	Feature type(s)				Dataset
						Cont.	User	Temp.	Struct.	
NIPS2010 [97]	Collaborative filtering	Classification	Matchbox	Micro	Probability	✓	✓	–	–	Twitter
SocialCom2010 [78]	Feature-based	Regression	Unnamed	Micro	Probability	✓	✓	–	–	Twitter
ICWSM2010 [89]	Feature-based	Regression	Unnamed	Macro	Value	✓	✓	✓	–	Twitter
CIKM2010 [92]	Feature-based	Classification	ARTs	Micro	Value	✓	✓	✓	✓	Twitter
WWW2011 [30]	Feature-based	Classification	Unnamed	Micro	Multiclass	✓	✓	✓	✓	Twitter
WebSci2011 [65]	Feature-based	Classification	Unnamed	Macro	Probability	✓	–	–	–	Twitter
AAAI2011 [69]	Feature-based	Classification	Unnamed	Macro	Binary	✓	✓	✓	–	Twitter
ICML2011 [82]	Time-series-based	Regression	Cox	Micro	Value	–	–	✓	–	Citation
KDD2012 [60]	Time-series-based	Regression	SPIKEM	Macro	Value	–	–	✓	–	Twitter
CIKM2012 [44]	Feature-based	Classification	Unnamed	Micro	Value	✓	✓	✓	✓	Twitter
SIGIR2012 [57]	Feature-based	Classification	Unnamed	Micro	Value	✓	✓	–	✓	Twitter
WSDM2012 [81]	Feature-based	Regression	Unnamed	Macro	Value	✓	–	✓	✓	Twitter
SIGKDD2013 [17]	Feature-based	Classification	OSLOR	Micro	Binary	–	✓	✓	–	Weibo
JASIST2013 [58]	Feature-based	Classification	Unnamed	Micro	Binary	✓	✓	–	✓	Twitter
WWW2013 [6]	Feature-based	Regression	Unnamed	Micro	Value	–	–	–	✓	Weibo
IJCAI2013 [98]	Feature-based	Classification	Unnamed	Micro	Binary	✓	✓	✓	✓	Weibo
WWW2013 [34]	Feature-based	Classification	Unnamed	Macro	Binary	✓	✓	–	–	Twitter
DASFFA2013 [49]	Time-series-based	Regression	SISe	Macro	Value	–	–	✓	–	Weibo
SIGKDD2013 [33]	Time-series-based	Regression	SCPP	Micro	Value	–	–	✓	–	Delicious
WSDM2013 [1]	Feature-based	Regression	Unnamed	Macro	Value	–	–	✓	–	Digg, YouTube
AAAI2014 [74]	Time-series-based	Regression	RPP	Macro	Value	–	–	✓	–	Citation
WWW2014 [16]	Feature-based	Classification	Unnamed	Meso	Binary	✓	✓	✓	✓	Twitter
SIGIR2014 [42]	Feature-based	Regression	Unnamed	Macro	Value/Binary	✓	✓	✓	✓	Twitter
WWW2014 [20]	Feature-based	Classification	Unnamed	Meso	Binary	–	–	✓	✓	Weibo
AOAS2014 [96]	Time-series-based	Regression	Unnamed	Micro	Probability	–	–	✓	✓	Twitter
ICODSE2014 [25]	Feature-based	Regression	Unnamed	Micro	Value	✓	✓	✓	✓	Twitter
WSDM2014 [86]	Feature-based	Classification	HybridBON	Micro	Rank	–	–	–	✓	Twitter
ICDM2014 [90]	Feature-based	Regression	IIBP etc.	Meso	Value	–	–	✓	✓	Weibo
KDD2014 [39]	Collaborative filtering	Regression	FEMA	Micro	Value	✓	✓	✓	✓	Weibo; MAS
ICDM2014 [103]	Feature-based	Classification	BCI	Micro	Value	✓	✓	–	✓	Weibo
APWeb2014 [21]	Feature-based	Classification	unnamed	Meso	Multiclass	–	–	✓	✓	Weibo
JNCA2014 [53]	Time-series-based	Regression	RepostsTree	Meso	Value	✓	–	✓	✓	Weibo
SBP-BRiMS2015 [2]	Feature-based	Classification	THF	Micro	Binary	✓	✓	–	✓	Twitter
WSDM2015 [22]	Time-series-based	Regression	PETM	Macro	Value	–	–	✓	–	Weibo
WWW2015 [7]	Time-series-based	Regression	SEHP	Macro	Value	–	–	✓	–	Weibo
SIGKDD2015 [104]	Time-series-based	Regression	SEISMIC	Micro	Value	–	–	✓	✓	Twitter

(Continued)

Table 2. Continued

Article	Type	Classification/ regression	Method	Level	Outcome	Feature type(s)				Dataset
						Cont.	User	Temp.	Struct.	
ICDM2015 [94]	Time-series-based	Regression	Weibull	Micro	Value	–	✓	✓	✓	Weibo
CIKM2015 [36]	Collaborative filtering	Classification	CRPM&IRPM	Macro	Value	✓	✓	–	–	Weibo
PAKDD2015 [54]	Feature-based	Classification	Unnamed	Micro	Binary	✓	✓	✓	✓	Weibo
JASIST2015 [43]	Feature-based	Classification	Unnamed	Macro	Binary	✓	✓	✓	✓	Twitter
ICWSM2015 [59]	Feature-based	Classification	Unnamed	Macro	Value	✓	✓	✓	–	Twitter
TKDD2015 [99]	Feature-based	Classification	Unnamed	Micro	Probability	✓	✓	✓	✓	Weibo
ICDMW2015 [45]	Feature-based	Regression	MARS	Meso	Value	✓	✓	–	–	Twitter
DASFAA2015 [48]	Collaborative filtering	Classification	Unnamed	Macro	Probability	✓	✓	–	–	Weibo
CCITSA2015 [8]	Feature-based	Regression	Unnamed	Micro	Rank	–	✓	✓	–	Weibo
DASFAA2016 [87]	Time-series-based	Regression	STH-Bass	Macro	Value	✓	✓	✓	–	Twitter
SIGIR2016 [35]	Collaborative filtering	Classification	OCCF	Micro	Probability	✓	✓	–	–	Weibo
CIKM2016 [100]	Feature-based	Classification	SUA-ACNN	Micro	Probability	✓	✓	–	–	Twitter
WWW2016 [5]	Time-series-based	Regression	ISEHP	Meso	Value	–	✓	✓	–	Weibo
CIKM2016 [63]	Time-series-based	Regression	HHP	Meso	Value	–	✓	✓	✓	Twitter
Inf.Sci.2016 [19]	Feature-based	Classification	TrendLearner	Macro	Multiclass	–	–	✓	–	YouTube
WSDM2016 [95]	Feature-based	Classification	Unnamed	Micro	Value	✓	✓	–	✓	Weibo
TKDE2016 [29]	Collaborative filtering	Classification	V2S	Micro	Value	✓	✓	–	–	Twitter
DEXA2016 [27]	Feature-based	Regression	FEP	Micro	Value	✓	✓	–	✓	Twitter
WWW2017 [28]	Collaborative filtering	Regression	GPOP	Meso	Value	–	✓	–	✓	Twitter, Behance
WWW2017 [77]	Feature-based	Classification	SansNet	Macro	Binary	–	–	✓	–	Facebook, Twitter
WWW2017 [14]	Feature-based	Classification	GroupPinto	Macro	Value	–	–	✓	–	Twitter
CIKM2017 [13]	Feature-based	Regression	DeepHawkes	Micro	Value	–	–	✓	–	Weibo, Citation
SIGIR2017 [102]	Feature-based	Classification	HCID	Meso	Value	✓	–	–	✓	Weibo, DBLP
CIKM2018 [85]	Feature-based	Regression	PreNets	Macro	Value	–	–	✓	–	Twitter, Amazon
DEXA2018 [88]	Time-Series-based	Classification	EPOC	Macro	Binary	–	✓	✓	–	Twitter, Weibo
ICCS2018 [37]	Collaborative filtering	Classification	Unnamed	Micro	Probability	✓	–	–	✓	Weibo
IJCAI2018 [84]	Feature-based	Regression	UMAN	Micro	Value	✓	–	✓	–	Twitter, Douban
Neu.comp.2018 [52]	Feature-based	Classification	C-RBFNN	Micro	Probability	–	✓	–	✓	Weibo

User-related. Here we refer to all users that have participated in the message diffusion. User-related features are either extracted from user profiles or the users' historical activity statistics. Historical statistics are the summary of the actions of the user before the time of prediction. These statistics show the past success of the user and reflect the influence of the user. The user profile includes the demographics and the social network status of the user.

The most straightforward user features include the number of followers and the age of an account. Other features extracted from historical statistics, such as tweet frequency and active days

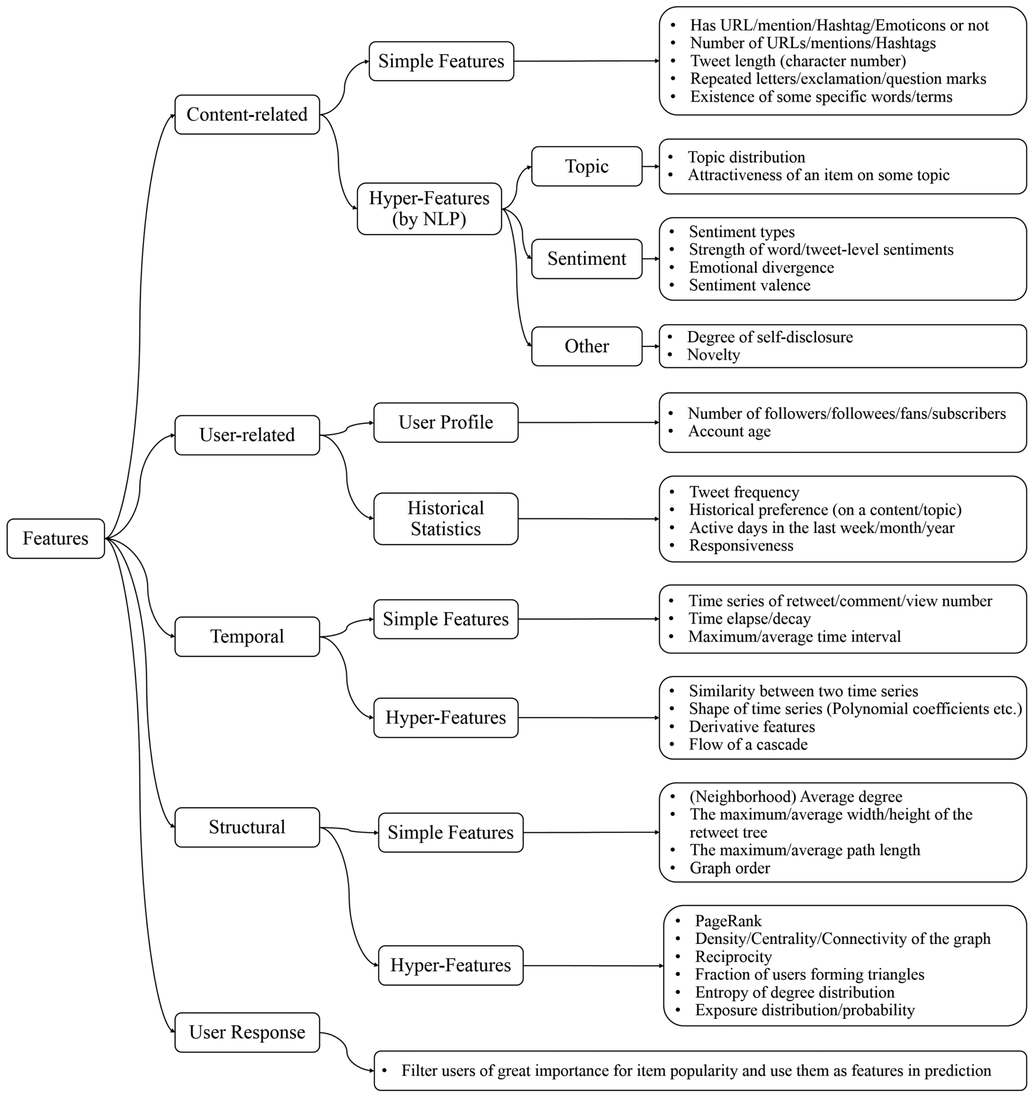

Fig. 2. Feature categories.

in the last period, are also useful to analyse a user's attribute in social network. Yang et al. [92] and Zhang et al. [100] proposed user interest (or preference) as a user-related feature, taking account of the contents of a user's historical tweets. Zhang et al. [98], Yuan et al. [95], and Hoang et al. [27] further extend this feature to the topic-specific level.

Petrovic et al. [69] pointed out the features of an author appear to be more important than features of the item itself. Multiple studies [4, 58] confirmed that the number of followers of the original author is an important predictor of popularity. Tsur et al. [81] also pointed out considering the fraction of past items that received reposts increases the accuracy of predictions. However, few studies report user-related features as effective as structural features or temporal features.

We use the following features in our implementation: number of reposts in the four weeks before t [89], mean number of reposts of the reposters in the four weeks before t, number of friends or followers of the author of the original post [78], mean number of friends or followers of the reposter [81], time the root user has been registered and mean time the reposters have been registered [75].

Structural. Structural features are most commonly extracted from users' friendship network, G_f. The term was coined in 2013 [6] though PageRank values were included for prediction in previous works [44]. Apart from using friendship networks, Gao et al. [20] used the network constructed by historical mention relationships. Some other studies directly focused on the retweet network and analysed its feature [20, 43, 96]. Furthermore, Yang et al. [92], Kupavskii et al. [44], and Lu et al. [53] used tree-based model to simplify the structure of retweet network, assuming retweet behaviour is triggered solely by a former user.

Apart from simple attributes of the networks such as the in-degree of the user, measures of connectivity and centrality are often used as well. Link density and the number of connected components are examples of metrics for connectivity. Intuitively, the more densely connected the network is, the higher probability messages are seen by other users. Authority scores obtained by PageRank [20, 58, 83] is a measure of the centrality, or in other words, the influence of the user. In Gao et al.'s experiment [20], maximum authority of authors appeared to be the best single feature.

Structural features were reported to improve prediction accuracy [16, 71]. When compared with content- and user-related features, structural features are more powerful in prediction. However, studies that include both structural and temporal features suggested that the structural features are not so effective as temporal features [16, 20]. It is also noteworthy that extracting structural features is generally time consuming and space consuming, due to the size of the network.

We use the following features in our implementation: the maximum authority of authors in G_r, reciprocity of network, number of connected component ($size > 2$) in the network, maximum size of connected component in the network, average authority of authors in the network [20], link density [101], clustering coefficient of network [30], authority of the author of the original post [58], number of connected component in the network [71], number of edges from early adopters to the entire graph, indegree of the ith reposter ($2 < i < k$) [4], number of nodes reachable in two steps from the early reposter [75], and indegree of the ith adopter on the subgraph [46].

Temporal. Temporal features are concerned with the repost time of a post. Researchers utilize the early repost series, from the time when the item was published to the time when we make the prediction t_p. The time series can be presented in two formats: one is the accumulated repost count at equal time intervals and the other is the timestamps of repost actions. Later, researchers brought up some derived temporal features such as the mean time interval.

Many experimental results, such as [20, 42, 75], showed that temporal features are the most effective type of features. Szabo et al. [79] also showed that temporal features alone can reliably predict future popularity. Shulman et al. [75] conducted a cross-domain prediction experiment and showed temporal features have the ability to generalize to other social network. Hong et al. [30] pointed out temporal features are more effective on small cascades rather than large cascades.

We use the following features in our implementation: time between the ith reposts and the $(i-1)$-th reposts ($2 < i < k$) [30], mean time between reposts for the first half (rounded down) of the reposts [75], mean time between reposts for the last half (rounded up) of the reposts, coefficients of polynomial curve fitting and symbolic sequences [42], dormant period, mean value and standard deviation of the time series, mean value and standard deviation of the absolute first-order derivative, k-dimension time vector and the maximum time interval [20].

User response. Many popular items are related with influential users—these users are experts at discovering content and have a large group of faithful fans. The response of these users are great indicators of potential popular cascades. The main difficulty is the selection of effective users since the number of users is too large to be directly used as features.

Table 3. Reported Effectiveness of Features

Paper	Year	Content-related	User-related	Structural	Temporal
[97]	2010	*	**	–	–
[78]	2010	*	**	–	–
[79]	2010	–	–	*	**
[92]	2010	–	–	*	–
[30]	2011	*	**	*	*
[65]	2011	*	–	–	–
[69]	2011	*	**	–	–
[44]	2012	*	*	**	**
[57]	2012	*	–	**	–
[81]	2012	*	–	**	**
[58]	2013	*	–	**	–
[6]	2013	*	–	**	–
[34]	2013	*	**	–	–
[42]	2013	*	**	**	***
[20]	2014	–	–	*	**
[16]	2014	*	*	**	**
[25]	2014	–	–	–	*
[103]	2014	–	*	–	–
[21]	2014	–	–	*	**
[43]	2015	*	*	*	**
[59]	2015	*	**	–	**
[94]	2015	–	–	*	**
[75]	2016	*	*	*	**
[100]	2016	–	*	–	–
[95]	2016	–	–	*	*
[27]	2016	*	*	–	–
[77]	2017	–	–	–	*

More $*$ means a stronger predictive power has been reported.

For every item, Cui et al. [17] aimed to select a subset of users that are tightly correlated with the popularity of the item. These users are used as features in a logistic regression model. Apart from maximizing prediction accuracy, two constraints are considered: limitation on the size of the user subset and minimal redundancy.

We list the categories of features used in previous literature and their reported effectiveness in Table 3. We can see that temporal features and structural features are most frequently shown to be effective. User-related features are less important when compared with temporal features and the structural features, though they have stronger predictive power than content-related features. Such comparison and conclusions are more likely to be found in earlier studies, since sufficient exploration has paved the way for later research, which focused more on learning approaches.

As for the machine-learning models, Ma et al. [57] were the first to try out multiple machine-learning models and evaluate them against each other. Following this lead, researchers began to compare the performance of different learning methods. However, most of the literature simply listed the results but did not give the principle for choosing models. Different methods were reported to perform best under different circumstances. This is somehow reasonable, since the performance of machine-learning models relies heavily on the features selected and the

Table 4. Machine-Learning Methods and Abbreviations

Notation	Description
GLM	Generalized linear model
LR	Linear regression
LogReg	Logistic regression
NB	Naive Bayes model
DT	Decision trees
NN	Neural network
CNN	Convolutional neural network
RNN	Recurrent neural network
LSTM	Long short-term memory
RF	Random forest
RT	Random tree
SVM	Support vector machine
KNN	k-nearest neighbours algorithm
MARS	Multivariate adaptive regression splines
CART	Classification and regression tree

evaluation criteria. We list some frequently used machine-learning models in Table 4, and collect all the machine-learning methods as well as hyper-features adopted by feature-based literature in Table 5.

4 TIME-SERIES-BASED METHODS

Time-series-based methods exploit the patterns in the sequence of repost times and model the process of repost arrival. In Table 6, we summarize the core functions of the models to be illustrated and provide an explanation of the symbols used. We also compare these models in terms of how a repost proves feedback to the model (excitation), how the time decay effect is captured, and whether the model has an explicit upper bound for the number of reposts.

Point processes. Every cascade can be broken up into single reposts that happen one by one, which can be described by a point process. Formally, a point process is a stochastic process $N(t)$ taking on integer values with $N(0) = 0$ and is a right-continuous step function with increments by 1 each time. $\mathcal{H}(u)$ is the history of arrivals up to time u. Another way to describe such a process is by recording the time of arrivals $T = \{t_1, t_2, \ldots\}$.

Point processes are characterized by a conditional intensity function $\lambda(t)$. Intuitively, the conditional intensity function refers to the expected rate of arrivals given history $\mathcal{H}$. This is also called the hazard function in survival analysis literature [12].

$$\lambda(t) = \lim_{h\to 0} \frac{E[N(t+h) - N(t)|\mathcal{H}(t)]}{h}.$$

The basic example of point processes is the Poisson process (PP) where the intensity function is constant. This means that the process is memoryless, which is counter-intuitive for items in social networks. Lu et al. [53] proposed a tree-based model, RepostsTree, which presents the whole repost process as a composite PP comprising individual ones. Featured by a distinct intensity λ_i, each single PP describes the generative scenario of a tree segmentation whose nodes constitute a sub-series.

The reinforced Poisson process (RPP) [74] is an example of inhomogeneous PP which takes the current state into account. Specifically, the amount of attention received $a_i(\cdot)$ is defined as a step

Table 5. Machine-Learning Methods and Hyper-Features Adopted by Feature-Based Literature

Article	Classification/ regression	ML methods	Hyper-features	
[78]	Regression	GLM	User-based	Tweet frequency
[89]	Regression	COX regression	Temporal	Stage (appearance time in the topic lifespan); mentioned rate of the tweeter
			Structural	The number of hops in a diffusion chain
[92]	Classification	Self-designed	Content-based	Frequencies of the terms; the sum of tf-idf values of keywords
			User-based	Users' historical preferences; the authority of users; the similarity between user u and message m
			Temporal	Time series of a message seen by a user; time delay
[30]	Classification	LogReg	Content-based	Topic distribution for every tweet
			User-based	How many times a user's tweets have been retweeted
			Temporal	Time difference between the current and the origin message; time difference between the current and the previous tweet; average time difference of consecutive messages; average time a user's messages get retweeted
			Structural	PageRank, degree distribution, local clustering coefficient, and reciprocal links
[65]	Classification	LogReg	Content	Sentiments; presence or absence of some terms; the probability of a term to occur in a retweeted or a non-retweeted message; topic feature
[69]	Classification	Passive-aggressive (PA) algorithm	Content-based	Novelty; is the tweet a reply; the actual words in the tweet
[44]	Classification	Two binary classifiers	Content-based	Valence, arousal, dominance (based on ANEW)
			User-based	Average global and local retweet ratio
			Temporal	For specified intervals: flow of a cascade; sum of average retweet ratios; sum of PageRank over users
			Structural	Weighted and not weighted PageRank
[57]	Classification	NB,DT (C4.5), KNN, SVM, maximum entropy (MaxEnt)	Content-based	Hashtag clarity; 20-dimension hashtag topic vector
			User-based	User authority
			Structural	User authority (PageRank); average authority of users; fraction of users forming triangles; density of graph; average edge weights; 15-dimension vector of exposure probability; ratio between number of connected components and number of nodes
[81]	Regression	LR, SVR	Content-based	Hashtag orthography (no caps, some caps, all caps, and contains digits); lexical items (the existence of five lexicon types); hashtag location; hashtag collocation; cognitive dimension (involving sentiment, sociality, behaviour etc.)
			Temporal	'Stickiness' and 'persistence': the ratio of change from the previous timestamp
			Structural	Retweets ratio; average number of followers
[17]	Classification	LogReg	–	–
[58]	Classification	NB, KNN, DT, SVM, LogReg	Content-based	Number of segment words; 3-Dimension semtiment vector; 20-Dimension topic distribution vector; Hashtag Clarity; segment words clarity;
			User-based	User authority
			Structural	User authority (PageRank); average authority of users; fraction of users forming triangles; density of graph; average edge weights; 15-dimension vector of exposure probability; ratio between number of connected components and number of nodes
[21]	Regression	LR	Structural	Link density; diffusion depth

(Continued)

Table 5. Continued

Article	Classification/ regression	ML methods	Hyper-features	
[98]	Classification	LogReg	Content-based	Topic distribution of microblog m
			User-based	Topic distribution of user v; topic propensity
			Temporal	Instantaneity
			Structural	Social influence locality; pairwise influence; structure influence; reciprocal
[34]	Classification	NB, LogReg	Content-based	Sentiment valence; emotional divergence; individual sentiments
[1]	Regression	Affinity propagation (AP)	Temporal	The attention a content attracts with respect to all other observed contents at some time interval; normalized rate of change in the attention attracted at some time interval; similarity between two time series
[16]	Classification	LogReg	Content-based	The probability of the photo having a specific feature (food, overlaid text, landmark, nature, etc.); word semtiment; word sociality
			User-based	Author's average active days in the last month; for the first k resharers: average friend/fan/subscriber count; average time since register; average active days in the past month; number of female
			Temporal	Time elapse; average time between reshares for the first and last half reshares, respectively; change in the time between reshares of the first k;
			Structural	For the first k resharers: number of author's direct friend/fan; Change in tree depth of the first k reshares; Average or 90th percentile tree depth
[42]	Regression	Weighted-SVM, LR, CART, Gaussian process regression (GPR), SVR, NN	Content-based	Strength of word-level sentiments; number of special signals; the count of the hashtag with different cases; co-occurrence frequency of hashtags; weighted average number and popularity of top-k similar historic hashtags;
			Temporal	Polynomial coefficients (shape of time series); symbolic sequences (shape of time series); derivative features
			Structural	Graph order; graph density; average degree; entropy of degree distribution
[20]	Classification	DT	Temporal	Time taken for first k retweet to arrive; max time interval
			Structural	For retweet network: density; diffusion depth; reciprocity; clustering coefficient; PageRank; connectivity; For border network: density; reciprocity; max and average authority; exposure distribution
[25]	Regression	SVR, LR	User-based	tweet frequency
			Temporal	Elapsed time; average time between responses; average number of interactions in a time window
			Structural	Retweet depth
[86]	Classification	RankingSVM	Structural	Correlation of two networks; hybrid potential influence
[90]	Regression	Self-designed	Temporal	Similarity between two time series
			Structural	Ratio of out-degree to in-degree; K-shell value
[103]	Classification	Self-designed	Content-based	Tweet latent features; content correlation between followers tweets and the target tweet
			User-based	User latent features
			Temporal	Followers' influence on retweeting with time decay
[21]	Classification	NB, KNN, SVM, LogReg, Bagging, DT	User-based	Authority score
			Temporal	Tweet time (a new notion); max and average time interval
			Structural	For retweet network: density; diffusion depth; reciprocity; clustering coefficient; PageRank ; connectivity; For border network: density; reciprocity; max and average authority; exposure distribution

(Continued)

Table 5. Continued

Article	Classification/ regression	ML methods	Hyper-features	
[2]	Classification	RF	Content-based	Link rate; distinct link rate
			Structural	PageRank; eigen vector centrality; closeness/betweeness/degree/indegree/outdegree centrality; #uninfected neighbors of early adopters; neighborhood average degree
[54]	Classification	LogReg, RF, DR, NB, SVM, KNN	Structural	The maximum width and height of the retweet tree; the average path length
[43]	Classification	SVM, LR, CART, GPR, SVR, NN	Content-based	Word-level sentiment strength; case-sensitive hashtag count; co-occurrence times with other hashtags; prototype feature: weighted average TBB or TRA and the number of top-k similar historic hashtags
			User-based	Passivity
			Temporal	Dormant period; polynomial coefficients (shape of time series); symbolic sequences (shape of time series); derivative features
			Structural	Density of the graph; average degree; entropy of degree distribution
[59]	Classification	SVR	Content-based	Parts-of-speech tag diversity; word clarity; hashtag clarity; Pagerank of the hashtags
			User-based	Mention multiplicity; retweet multiplicity
			Temporal	Early coinage; time gap between first few tweets
[99]	Classification	LogReg, factor graphic model	Content-based	Topic distribution of microblog m
			User-based	Topic distribution of user v; topic propensity
			Temporal	Instantaneity
			Structural	Social influence locality; pairwise influence; structure influence; recoprocal
[45]	Regression	MARS	–	–
[8]	Regression	Self-designed	User-based	Initial score (based on #follower)
			Temporal	Attention over time (based on comment and retweet count)
[100]	Classification	CNN	Content-based	Word vector; tweet matrix
			User-based	User interests; similarity of tweet and user interests
[19]	Classification	ExtraTree	Temporal	Similarity between two time series; probability of beloinging to a time-searies class; change rate of views/comments/favourites; peak fraction; time window size
[95]	Classification	SVM, LogReg, LambdaMART	Content-based	Sentiment; self-disclosure
			User-based	Profile affinity; topic affinity; responsiveness
			Structural	Reciprocity: Reciprocal Interaction Rank (RIR)
[27]	Regression	LogReg	Content-based	Attractivness of an item on topic k
			User-based	User reputation on topic k; user interest on topic k; sharing level; willingness of diffusion
			Structural	PageRank
[77]	Classification	Self-designed	Temporal	Survival probabilities
[85]	Regression	RNN	Temporal	{number of retweets, increment of retweets, mean time interval between retweets} in the first half of observing interval, maximum time interval between retweets, mean and standard deviation of {retweet number, time interval between retweets} per time unit etc.
[84]	Regression	LSTM, CNN	Content-based	Word embedding, attention
			Temporal	Sequential content correlation, sequential popularity correlation, user memory
[52]	Classification	NN	User-based	Historical activity, follower number, followee number, retweet rate, average inward influence
			Structural	Retweeter network, potential retweeter network

Table 6. Comparison of Time-Series-Based Methods

Name	Function	Interpretation	Excitation	Time decay	Bound
PP	$\lambda(t) = \lambda$	λ: constant intensity rate	None	None	No
RepTree [53]	$\lambda(i, t) = \lambda_i$	i: sub-serial triggered by user i	None	None	No
RPP[74]	$\lambda(t) = \lambda_i f(t; \theta) a_i(t)$	λ_i: item attractiveness; $f(\cdot)$: time decay function; $a_i(\cdot)$: amount of attention received	Accumulated	Log-normal	No
PETM[22]	$\lambda_m(t, k) = c_m r_m(k) t_k^{-\gamma_m}$	$\lambda_m(\cdot)$: rate function of item m; $r_m(\cdot)$: total number of retweets; t_k: the time delay when total retweet count reaches k; γ_m: decay rate	Accumulated	Power law	No
SCPP [33]	$\lambda_{(i,t',u')}(t, u) = \alpha_{u'} \theta_{u'u} e^{-\gamma(t-t')}$	λ: background intensity; $\alpha_{u'}$: influence of user u' on the other users; $\theta_{u'u}$: strength of the relation from user u' to user u	None	Exponential	No
HP	$\lambda(t) = \lambda + \int_0^t \mu(t-u) dN(u)$	λ: background intensity; $\mu(\cdot)$: excitation function	Single	Exponential	No
SEHP[7]	$\lambda(t) = v e^{-\beta t} + \alpha \sum_{j=1}^{jmax(t)} e^{-\beta(t-t_j)}$	v: attractiveness of the original microblog; α: the triggering strength of each subsequent forwarding; $j_{\max}(t)$: the index of the last forwarding before time t	Single	Exponential	No
ISEHP [5]	$\lambda(t) = v e^{-\beta t} + \sum_{j=1}^{jmax(t)} a_j e^{-\beta(t-t_j)} + \gamma \sin(\frac{2\pi}{p}(t+s))$	v: attractiveness of the original microblog; a_j: attractiveness of the subsequent microblog; s: the phase shift of periodicity; : strength of periodicity	Single	Exponential	No
HHP [63]	$\lambda(t) = \sum_{t_i<t} \kappa m^{\beta} (t - t_i + c)^{-(1+\theta)}$	$\lambda(t)$: arrival rate of new events; κ : virality of the tweet content and it scales the subsequent retweet rate; β: a warping effect for the user influence; c: temporal shift term	Accumulated	Power law	Yes
SEISMIC [104]	$\lambda(t) = p_t + \Sigma_{t_u<t} n_i \phi(t - t_u)$	n_i: node degree; $\phi(t)$: human reaction time; p_t: post infectiousness	Single	Power-law	No
Weibull [94]	$\lambda_i(t) = \frac{k_i}{\lambda_i}(\frac{t}{\lambda_i})^{k_i-1}$	λ_i: scale; k_i: shape	None	Exponential	Yes
Cox [82]	$\lambda_i(t) = \alpha_0(t) e^{(\beta^T s_i(t))}$	$\alpha_0(\cdot)$: baseline hazard function; $\mathbf{s_i}$: vector of features; β: vector of coefficients	Accumulated	Depends on $\alpha_0(\cdot)$	No
SI	$\frac{dI(t)}{dt} = \beta \times (N_u - I(t)) I(t)$	N_u: total nodes; $I(t)$: number of infected nodes; β: the infectiousness	Accumulated	None	Yes
SPIKEM [60]	$\Delta R(n+1) = U(n) \Sigma_{t=n_0}^{n} (\Delta R(t) + S(t)) f(n+1-t) + \epsilon$	$R(t)$: the number of users that have reposted; $U(t)$: the number of users that have not reposted; $S(t)$: shock; $f(\cdot)$: time decay	Single	Power law	Yes
Bass [87]	$\frac{dH(t)}{dt} = (p + qH(t))(1 - H(t))$	$H(t)$: fraction of adoption; p: external influence; q: internal influence	Accumulated	None	Yes
SISe [49]	$\Delta I(t+1) = \beta I(t) S(t) + \eta I(t) + \gamma E(t) - \alpha I(t)$	$I(t)$: the number of message retweeters; β: retweeting transmission rate; α: retweeter-to-follower change rate; η: the multi-retweeting rate; γ: the externally spontaneous retweeting rate; ω: proportion of external spontaneous visitor	Single	None	Yes

function with prior m which is proportional to the current repost count. This implies that the more popular an item currently is, the more likely that a repost action will happen in the near future. In [22], the reinforcement function $a_i(\cdot)$ was generalized to be any piecewise constant function which is defined on number of retweets, i.e., during the time interval between the $(k-1)$-th and the kth retweet, $a_i(\cdot)$ stays unchanged. In particular, they observed that as the number of reposts grow, the amount of attention received grows slower than linearly. Hence, they modelled the effect of reposts as a geometric sequence.

Another method based on inhomogeneous PP [33] is the shared cascade Poisson processes proposed by Iwata et al. In this method, user-related and structural features are introduced into the intensity function, where a'_u and $\theta_{u'u}$ are parameters representing a user's influence on others or a specific follower. Decay of the influence is modelled as exponential form with the parameter $\gamma \geq 0$. Note that an event may occur without influence from preceding events (i.e., emergence of the first post) or preceding users (retweet without following first), then the intensity function degenerates into $\lambda_{(i,0)}(t, u) = \alpha_i \theta_{0u}$. In this case, $\alpha_i \geq 0$ is the general popularity of item i, and $\theta_{0u} \geq 0$ represents the probability that user u adopts an item without any trigger.

The Hawkes process (HP) is a point process that describes self-exciting properties. Self-exciting point processes are frequently used to model the 'rich get richer' phenomena [60, 64]. Such phenomena is present in social networks because when a post is reshared, it will be exposed to a larger audience, who may reshare the post later and influence their followers. When an arrival occurs, the condition intensity function increases, causing temporal clustering.

Note that if the excitation function $\mu(\cdot) = 0$, the HP degrades into the homogeneous PP. The HP is flexible in the sense that the excitation function can be tailored for the specific problem.

Bao et al. [7] proposed the self-excited Hawkes process (SEHP) and use the exponential excitation function to model a single tweet cascade as an HP. Different parameters are used for the initial post and further reposts, implying that they are of different importance to the final popularity. Bao [5] extended SEHP into the influence-based SEHP by introducing a user-specific triggering effect into the self-excitation term. Different forwarding strength values were allocated to different subseries with respect to each user and his/her direct followers in followship network. Besides, an exogenous effect term was added to the rate function to model the external influence from the platform itself. In [91], the authors treated the activities of a single user as an HP and considered the interaction between users. This is a multi-dimension extension from the previous case.

On the basis of previous work, Zhao et al. [104] extended the Hawkes model to a doubly stochastic self-exciting point process SEISMIC. The post infectiousness p_t measures of its probability of being reposted at time t. When the post infectiousness is constant, the model is the standard HP. p_t is not modelled in a parametric form, but estimated from the observations using a triangular kernel. As for the reaction time factor $\phi(t)$, experimental analysis on Twitter data shows that it remains constant for a short period and then follows a power-law decay. Other works show that the tail follows a log-normal distribution [96]. The exact distribution may vary between different social networks and thus should be confirmed against real data.

Mishra et al. [63] used a feature-based predictive layer on top of the Hawkes self-exciting point process. The content-quality κ and the event-magnitude m_i together account for the magnitude of influence. The exponent β works as a warping effect for the user influence, related to the long-tailed distribution of user influence in social network. Time decay was represented as a power-law triggering kernel. In particular, a branch factor n^* is modelled using the aforementioned components and acts as a common ratio of the predictive layer, where the repost cascade was deemed as a geometric progression over reposter generations. Note that both the branch factor and expected number in each generation contribute to the self-excitation, so the idea of doubly exciting point process is actually adopted here.

Yu et al. [94] adopted the Weibull model for the intensity function. Both parameters k and λ are estimated using the maximum likelihood. The homogeneous PP is a special case of the Weibull model when $k_i = 1$.

Vu et al. [82] proposed to use the proportional hazards model (Cox model) in. This model combines features with point processes. A similar idea was proposed in [24], however, the latter focused on the task of network inference.

Compartment models. Compartment models were first applied to model the spread of epidemic diseases. These models have some common characteristics: being defined on a finite population and dividing the population according to their state. Compartment models are inherently self-excited in the initial phase since the susceptible are transformed into the infected which then attempt to infect others. However, due to the limits on the population, compartment models often reach an equilibrium later on while non-terminating point processes have the risk of exploding to infinity.

The most basic epidemic model consists of two states: susceptible (S) and infected (I), hence the name 'susceptible-infected' (SI) model. Every infected node attempts to infect each of its neighbours independently with a probability β. Once infected, the node will remain in an infected state.

In [49], traditional SI model was modified into a SI-susceptible extended version, considering more complex states in the context of popularity predicting. Four processes were considered in total: (1) a retweeter manages to infect his/her direct followers with rate β; (2) a retweeter returns to the susceptible state at rate α if he/she is a direct follower of some current retweeter; (3) a retweeter remains infected by retweeting the same tweet multiple times at rate η; (4) an external visitor (that is, he has not followed any current retweeter) spontaneously repost the tweet at rate γ. The number of message retweeters, number of direct followers of each retweeter at t, and number of external spontaneous retweeters were constantly tracked over time series. They were presented by difference equations combined with the aforementioned rates. Hence, values of these rates can be estimated through time series and short-term predictions can be accomplished.

Matsubara et al. [60] also pointed out that the classical SI model does not capture the patterns of real-life information cascades. Under the assumptions of SI, the solution for $I(t)$ is sigmoid and the rise and fall of the derivative are both exponential. However, the fall of the intensity in social networks follows the power-law or log-normal distribution. Building on the SI model, they devised the SPIKEM model. The SPIKEM model has two states: the unposted or uninformed state (U) and the reposted state (R). Compared to the HP, the SPIKEM model allows multiple nodes to be activated (multiple arrivals in point processes) at a single time. Both models consider the excitation of each activated node separately and do a summation over the nodes. $f(\cdot)$ explicitly reflects the power-law time decay. ϵ is a noise term.

Observing that posting behaviour is often periodical in time, a periodicity factor $p(\cdot)$ is introduced into an improved version of SPIKEM. In essence, the periodical pattern is captured by a generalized trigonometric function.

The Bass model [9] for describing the adoption of new products in the marketing field can be seen as a derivation of the SI model that considers external factors. Yan et al. [87] extended the Bass model for tweet prediction by incorporating user features x and content features y. This is yet another example of combining features with models.

In Figure 3, we show how different time-series models fit to an actual cascade. The PP is obviously too simple to capture the trend pattern. Bass seems to fit the factual curve pretty well, while Weibull tends to overestimate the cascade size. Similarly, HP also overrates the popularity, probably due to its rudimentary self-excitation mechanism.

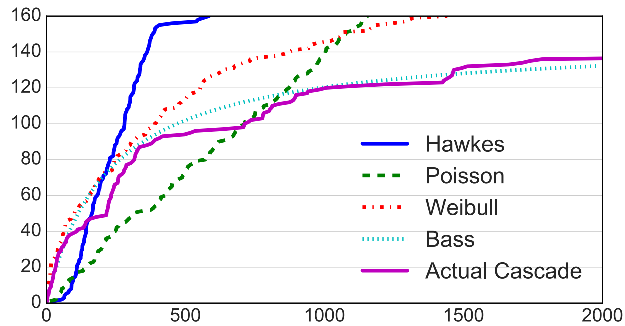

Fig. 3. Fitting time-series models to an actual cascade.

Table 7. Symbols for Collaborative Filtering Methods

Symbol	Description
R	Repost matrix
U	User latent factor matrix
V	Item latent factor matrix
Y	Indicator matrix or weight matrix
$\odot$	Hadamard product
S	User–user relationship/influence matrix
N_i	Neighbors of user u_i
T	Content topic matrix
C	Content similarity matrix
A	User–user similarity matrix
F	User–user interaction matrix

5 COLLABORATIVE FILTERING METHODS

Collaborative filtering prediction methods use interaction history of users as the basis for prediction. Instead of extracting features from users' historical reactions as in feature-based methods, here we only focus on whether a user has participated in a cascade or not. Commonly used symbols for collaborative filtering methods are shown in Table 7.

Matrix Factorization. For this category of prediction methods, we use an alternative data representation: user-item repost matrix R.

$$R_{ij} = \begin{cases} 1, & u_i \text{ reposted } m_j \\ 0, & \text{otherwise} . \end{cases}$$

By employing collaborative filtering, our goal is to predict the entries of the final repost matrix. It is noteworthy that another motivation that leads to this goal is social recommendation, where we actively present the predicted items in the users' feed. Nevertheless, in both cases, we evaluate the system by standard metrics such as accuracy and recall.

Collaborative filtering has been proved to be very successful in recommendation systems. The assumption behind it is that users who consume similar items share the same tastes and interests, thus they will continue to behave alike. Applied to the problem of popularity prediction, this assumes that users who repost the same items will continue to do so for future cascades.

The basic matrix factorization model aims to minimize:

$$\min_{U,V} \mathcal{J}(R,U,V) = \|Y \odot (R - UV^T)\|_F^2 + \alpha\|U\|_F^2 + \beta\|U\|_F^2.$$

U is latent user feature matrix and V is the latent item feature matrix. Y is an indicator matrix showing whether user i has reposted item j. The popularity prediction problem is transformed into a matrix completion problem of R.

There are two shortcomings with directly applying matrix factorization: the first is that it completely ignores available information about users or items other than the repost action. Leveraging item features adds a content-based flavour and making use of social relations brings a complementary source of information. The second is that it is hard to distinguish missing data with negative data. A user may have not reposted an item because he/she simply has not seen it or because he/she does not find it interesting. We do not explicitly observe negative responses, only positive responses.

SoRec [55] assumes that the users should share the same user preference vector u_i in the repost space and the social space, which is to say that the repost matrix and the social matrix should be coherent. Thus, it performs a co-factorization between the two. The optimization problem is as follows:

$$\min_{U,V,Z} \mathcal{J}(R,U,V,Z,S) = \|Y \odot (R - U^T V)\|_F^2 + \alpha\Sigma_{i=1}^n \Sigma_{u_k \in \mathcal{N}_i} \left(S_{ik} - u_i^T z_k\right)^2$$
$$+ \beta\|U\|_F^2 + \gamma\|V\|_F^2 + \lambda\|Z\|_F^2.$$

S is the user–user relationship matrix, $S_{ij} = 1$ when u_i and u_j are friends. Z is a learned latent factor matrix. $\mathcal{N}_i$ is the set of users directed connected with u_i. Y is the repost action indicator matrix.

SoReg [56] is based on the phenonemon of homophily, assuming that the preferences of a user would be similar to that of his friends. We either perform regularization on the user preference matrix U by using the average of the friends' tastes or by the sum of individual friends.

Jiang et al. attempted to tackle the first problem by introducing message clustering as a regularization factor into their centroid-based regularization prediction model (CRPM) [36]. Items were represented by content-based feature vectors and then clustered using the k-means algorithm. The underlying rational is that items that are within the same cluster in the observed space should also be close to the cluster centroid in the latent space. The cluster centroid regularizer factor is as follows:

$$\Sigma_{j=1}^N \Sigma_{k=1}^K T_{jk} \|V_j - \frac{1}{|C_k|}\Sigma_{m_x \in C_k} V_x\|_F^2.$$

C_k represents the kth cluster and T_{jk} is an indicator function showing if the message m_j belongs to the cluster C_k. At the same time, they proposed an Individual-based Regularization Prediction Model (IRPM) to try and make items from the same cluster close to each other. The regularization factor for IRPM is

$$\Sigma_{j=1}^N \Sigma_{k=1}^K T_{jk} \Sigma_{m_x \in C_k} S(j,x) \|V_j - V_x\|_F^2.$$

The ideas behind CRPM and IRPM are the same, only the optimization objective differs. The clustering constrains matrix factorization with the observed features of the items.

HF-NMF [18] jointly incorporates social factors and content factors for recommendation. A minor difference is that the authors believe that the product of the latent user matrix and the latent item matrix should be the item-level social influence instead of the repost matrix directly. We define the strength of user u_i's influence on item j as f_{ij}. Then the influence matrix X is defined as the regularized influence matrix:

$$X_{ij} = \frac{f_i j}{\mathcal{N}_i}.$$

Then we construct a user–user influence matrix S by a weight sum of the percentage of active friends and the average friend tie strength. For the post content, we use LDA to obtain an item-topic matrix C. Summing up, the optimization problem is consisted by a basic matrix factorization, a user-level regularization and an item-level regularization.

$$\min_{U,V,G} \mathcal{J}(X,U,V,G,S,C) = \|Y \odot (X - UV^T)\|_F^2 + \alpha\|S - UU^T\|_F^2 + \beta\|C - VG^T\|_F^2 + \gamma\|U\|_F^2 + \delta\|V\|_F^2.$$

ContextMF [38] gives a diffusion-based adoption model that considers the sender and receiver of items. An item can only be adopted if it matches the user's preference and it is reposted by his friends. G is the item sender matrix, in other words, the current repost matrix. W is the user–user preference similarity matrix, C is the item–item content similarity matrix, and F is the user–user interaction matrix. S is the latent interpersonal influence matrix which needs to be learned along with latent user matrix U and latent item matrix V. The optimization function is regularized in three different aspects: users that are similar in latent space U should also share preferences in the observed matrix W; items that are similar in the latent space V should also be similar in content C; high interpersonal influence in S should lead to frequent interactions in F.

$$\begin{aligned}&\min_{U,V,S} \mathcal{J}(U,V,S,R,G,W,C,F)\\ &\quad= \|R - SG^T \odot U^TV\|_F^2 + \alpha\|W - U^TU\|_F^2 + \beta\|C - V^TV\|_F^2 + \gamma\|F - S\|_F^2\\ &\qquad+ \delta\|S\|_F^2 + \eta\|U\|_F^2 + \lambda\|V\|_F^2.\end{aligned}$$

Li et al. [48] transformed item space into keyword space, for it is much likely that a user will repost tweets containing the same words other than repost the same tweet more than one time. Moreover, we take account of the author's (or publisher's) influence on both the tweet itself and the readers. We define the preference of user u towards tweet i as follows:

$$y_{u,i} = p_u^T\left(\frac{1}{Z}\Sigma_{w\in T_i} q_w + \alpha d_{p(i)}\right) + \frac{1}{Z}\Sigma_{w\in T_i} q_w^T \beta d_{p(i)},$$

where p_u is the feature vector extracted from the latent user feature matrix, q_w from the latent word (in tweet T_i) feature matrix, and $d_{p(i)}$ from the matrix of publishers $p(i)$ and latent factors. Note that by taking user, publisher, and tweet keyword all into consideration, both the two shortcomings mentioned above are mitigated.

Pan et al. [67] adopted one-class collaborative filtering (OCCF) to solve the second problem. The OCCF problem is defined under the setting of implicit feedback, which in our case, is implicit negative response. All missing values are still treated as negative, but by adding weights to the examples we distinguish between interactions that are likely to happen and those that are not. The objective function is the basic matrix factorization function with the indicator matrix Y replaced by a user-item repost probability matrix.

$$Y_{ij} = \begin{cases} 1, & R_{ij} = 1\\ \eta S_{\text{interest}}(u_i, m_j) + (1-\eta) S_{\text{influence}}(u_i, v_{m_j}), & R_{ij} = 0\end{cases}$$

v_{m_j} is the author of m_j. S_{interest} captures the degree that the interest of u_i aligns with the topic of m_j. This is calculated by the consine similarity between the topic vectors of the user and the item. Topics are extracted from the original text by LDA. $S_{\text{influence}}$ captures the influence between a pair of users. It is consisted of two factors: the repost probability $P(u,v)$ and the repost $H(u,v)$

frequency. The larger the repost probability S_{ij} is, the greater penalty for $U_i V_j$ not being equal to R_{ij}.

Tensor decomposition. In [39], flexible evolutionary multi-faceted analysis (FEMA) extends this problem for multiple types of context and temporal evolution through tensor analysis. User behaviour is represented as a sequence of tensors over time. Regularization is also used to tackle the sparsity problem as with the above matrix factorization approaches. The behaviour tensor is decomposed into a core tensor and projection matrices and then the product of the latter is used for prediction. The algorithm is based on tensor perturbation theory and performs incremental updates as the behaviour tensor evolves.

It is noteworthy that FEMA is a very flexible framework and could be applied to various pattern discovery and prediction tasks. One example may be predicting tweet with mentions. Here we represent a tweet by a tuple (user s, mention d, word w, time t). Then the original data would be a three-order tensor sequence $\mathcal{X}_t$. We seek to factorize this tensor sequence into the product of a core tensor and three projection matrices.

$$\mathcal{X}_t = \mathcal{Y}_t \times S_t \times D_t \times W_t.$$

User's social relations and word semantics may be encoded in Laplacian matrices $L_{(S)}, L_{(w)}$ for regularization.

V2S framework proposed by Hoang and Lim [29] also builds a three-order tensor δ_{uvm} for each time window, and carries out retweet predictions through decomposition with four factor matrices. In this framework, users' adoption behaviour in social network is attributed to three topic-specific behavioural factors, including topic virality, topic-specific user virality, and topic-specific user susceptibility. Observed data are represented by a tuple (u, v, m) where m is a content item generated by user u, and exposed to user v. δ_{uvm} is a binary variable indicating whether v adopts m or otherwise. A topic distribution matrix as well as the three factors mentioned above constitutes the factor matrices of our tensor-decomposition model, that is,

$$l(\delta_{uvm}) = \Sigma_{k=1}^{K} [D_{m,k} \cdot V_{u,k} \cdot I_k \cdot S_{v,k}],$$

where $l(\cdot)$ represents the likelihood, $D_{m,k}$ represents the topic distribution of a content item m, $V_{u,k}$ is the virality of user u for topic k, I_k is topic k's endogenous virality, and $S_{v,k}$ is the susceptibility of user v for topic k. Note that *topic* serves as a latent feature or dimension in V2S. For optimization, both a numerical model and a probabilistic model are proposed based on the above factorization method.

Hoang et al. [28] also adopted tensor analysis and proposed a hierarchical framework, Group-level POpularity Prediction (GPOP), which adapts itself specially to the context of social network. Instead of representing users' historical behaviour as a tensor sequence over time as in FEMA, GPOP explicitly introduces time series into the tensor as a dimension. Moreover, GPOP performs a user clustering based on graph and thus reaches a two-level hierarchy, group-level (or meso-level) and population-level (or macro-level). Hence, adoption behaviours (i.e., retweeting) can be represented as a triplet (group c_i, content p_j, time t_k) at group level, and (1, content p'_j, time t'_k) at population level. Note that the first number in the latter triplet is always 1, because we do not care about which user exactly contributes to the popularity of some item at population level.

Therefrom, original data are actually transformed into two three-order tensors, $\mathcal{T} \in \mathbb{R}^{k \times q \times l}$ and $\mathcal{Y} \in \mathbb{R}^{k \times q \times 1}$, each of whom is then decomposed into three matrices using PARAFAC [41] such that, for all observed entries:

$$\mathcal{T}_{itj} = \Sigma_{r=1}^{R} D_{ir} J_{tr} F_{jr},$$
$$\mathcal{Y}_{itj} = \Sigma_{r=1}^{R} H_{ir} J_{tr} (\vec{1}F)_r.$$

It is worth mentioning that another two low-order tensors are also used in GPOP, specially storing the information on the target item. By sharing a same factor matrix F, this hierarchical model can be efficiently learned.

Transfer learning. It is also possible to exploit auxiliary information from other social platforms through transfer learning. Assuming that users and items are fully overlapped with another domain, Pan et al. [68] proposed coordinate system transfer. We use $R^{(1)}$ to represent the repost matrix (or rating matrix) that shares the same users and $R^{(2)}$ to represent the repost matrix that shares the same items. Given these two observed matrices, we first perform sparse SVD on $R^{(1)}$, $R^{(2)}$.

$$\min_{U^{(i)}, V^{(i)}, B^{(i)}} \|Y^{(i)} \odot U^{(i)} B^{(i)} V^{(i)T}\|_F^2.$$

$B^{(i)}$ is a diagonal matrix filled with eigenvalues. This is to scale $U^{(i)}$ and $V^{(i)}$ so that $U^{(i)}U^{(i)T} = I$, $V^{(i)}V^{(i)T} = I$. U, and V can be seen as coordinates in the latent space, and since they overlap in either users or items, the corresponding coordinates might as well stay unchanged for the target domain. Thus, we let $U_0 = U^{(1)}$ and $V_0 = V^{(2)}$ and regularize the factorization of the target domain by trying to keep the coordinates the same.

$$\min_{U,V,B} \mathcal{J}(U, V, B, R, U_0, V_0) = \|Y \odot (R - UBV^T)\|_F^2 + \alpha\|U - U_0\|_F^2 + \beta\|V - V_0\|_F^2.$$

However, in reality, finding two auxiliary sources that perfectly overlap in users and items may be a difficult task on its own. As users commonly engage in two or more social network platforms that are of different focus, Jiang et al. [40] proposed XPTrans to exploit this portion of overlapping users to perform cross-platform recommendation. The key finding is that even with a small overlapping crowd, we are able to cover the majority of items in the domain. Hence, this overlapping crowd can be used to regularize the transfer process.

$$\begin{aligned}&\min_{U,V,U',V'} \mathcal{J}(R, R', U, U', V, V', Y, Y', A, A')\\ &\quad = \|Y \odot (R - UV^T)\|_F^2 + \alpha\|Y' \odot (R' - U'V'^T)\|_F^2 + \beta\Sigma_{i_1,i_2,j_1,j_2} Y^0_{i_1,j_1} Y^0_{i_2,j_2} (A_{i_1,i_2} - A_{j_1,j_2})^2.\end{aligned}$$

A is the user-based similarity matrix, and $Y^0 = Y' \odot Y$ which indicates whether the user belongs to the overlapping crowd.

We summarize the factors considered by the above models in Table 8. It is noteworthy that compared with time-series-based methods, collaborative filtering approaches commonly pay more attention to the attributes of users and item contents. Only a few of them explicitly take temporal factors into account. In addition, as higher level form of information, cross-domain knowledge is used in the last two methods.

6 CATEGORIZING METHODS BY RATIONALE

In this section, we explore the rationale behind popularity prediction and categorize them according to this dimension. We find that prediction methods either model the motivation behind repost actions or monitor early responses. We show the rationale behind different popularity prediction methods in Table 9.

6.1 Motivation-Oriented Methods

Items gain popularity on social networks through the sharing of users, thus the motivation beneath such an action is critical in understanding information cascades. We divide motivation into internal motivation and external motivation, referring to them as the homophily effect and the influence effect below.

Table 8. Factors Considered for Collaborative Filtering

Method	Content	User	Temporal	Cross-domain
SoRec [55]	✓	–	–	–
SoReg [56]	✓	–	–	–
CRPM [36]	–	✓	–	–
IRPM [36]	✓	✓	–	–
HF-NMF [18]	✓	✓	–	–
ContextMF [38]	✓	✓	–	–
OCCF [67]	✓	✓	–	–
PRP [48]	✓	✓	–	–
FEMA [39]	✓	✓	✓	–
GPOP [28]	–	✓	✓	–
V2S [29]	✓	✓	–	–
CST [68]	–	–	–	✓
XPTrans [40]	–	–	–	✓

Table 9. Comparison of Rationale

Name	Motivation-oriented		Monitor-based
	Homophily	Influence	
Feature-based	✓	✓	✓
Time-series-based	×	×	✓
Collaborative filtering	✓	×	✓
OSLOR	×	✓	✓

The influence of friends have been seen as an important factor. It has been observed that influence can affect our choice of information consumption, preferences for cultural items, adoption of innovations, and even political votes [73]. Many believe that influence is the driving force behind viral diffusion. However, we cannot neglect the impact of homophily. As 'birds of a feather flock together', users that are similar naturally adopt the same items [62]. Studies have been dedicated to distinguishing one from another but here we focus on how prediction methods incorporate these effects to explain popularity and more importantly, predict popularity.

Homophily. Homophily can either be explicitly observed by user demographic features or implicitly captured by historical reposting behaviour.

An important indicator of homophily is user's interest towards content. To extract such a feature, many methods summarize the content of the item by topics, with LDA being the most commonly used topic model. Such a model received success in analysing articles, but has been reported to perform not as well on microblog text, due to the irregular grammar and self-created phrases. Bian et al. [10] proposed to classify microblogs by transfer learning. Both text and images were accounted for by comparing microblogs with new articles published in the same time period. Some feature-based methods also extract content features such as the number of URLs, images, or punctuation marks.

Collaborative filtering methods predict popularity mainly by the effect of homophily. In the basic CF setting, latent user features and item features are found solely from the user repost records. Pan et al. [67] also computed similarity scores of the user interest and item content by the use of topic models.

Influence. In social networks, users exert influence on their followers and as a result, followers will tend to adopt items that are created or reposted by the former. This is particularly evident in the rise of online celebrities.

The most direct indicator of user influence is perhaps user features such as the number of followers. Another commonly used indicator is user authority, mostly obtained by PageRank.

When using users as sensors, [17] attached weights to users as a measure of their correlation to item popularity. From another point of view, these weights also reflect the influence users have—the amount of attention they draw once they are participants of the diffusion. On micro-level, Zhang et al. [99] pays attention to ego network and studies how friends in a user's ego network influence his/her retweet behaviours. This angle of view differs from other research by investigating inward influence exerted on a user, rather than outward influence of someone.

6.2 Monitor-Based Methods

Apart from explaining popularity with motivation, another effective method is to observe popularity.

Temporal features and time-series methods take great advantage of the early response towards an item as a sign of future popularity. According to the findings of [75], temporal features are the most effective in improving prediction accuracy. Time-series modelling use only the time of reposts, achieving parsimony and competitive accuracy at the same time.

Although accounting for the motivation behind reposting is a more fundamental approach, temporal data is quite valuable in the sense that it reflects the result of interplay between latent factors, many of which are not sufficiently captured in the former approach.

A chronological tree of microblog popularity prediction is shown in Figure 4. The basic categorization still follows the three lines of thoughts, i.e., feature-based, time-series modelling, and collaborative filtering. More detailed categories vary over these three types for their own properties and methods, which have been mostly illuminated above. Note that feature-based approach is a relatively flexible type of methods, due to the diversity of feature set options. Some feature-based studies apply the idea of time-series modelling and collaborative filtering by constructing hyper-features or modifying machine-learning methods. With respect to these studies, we use dashed lines to indicate their latent relationships.

7 EXPERIMENT SETUP

7.1 Data Description

Two different datasets were used for evaluation, one from Twitter[1] and the other from its Chinese equivalent Weibo.[2] Both are very popular social networks, with reported active users of 310 million per month on Twitter and 260 million on Weibo. Although the dominant language is different, the information diffusion mechanism is the same: items need to be reposted by friends to be visible in the feed. The friend relationship on both networks are asymmetric—user A is allowed to follow any user B without his/her permission. This unidirectional relationship allows user A to see user B's actions without having his/her own actions appearing in user B's feed.

7.1.1 Twitter. The Twitter dataset is sampled from the Twitter API ranging from July 24, 2015 to Jan 31, 2016. The tweet id, post time, content, and user id are collected for every tweet and retweet. In addition, the friend network was crawled. The dataset contains 2,206,219 microblogs and 2,165,863 users with 4,965,964,514 following relationships and 420,833 reposts. This dataset is a random sample of all cascades on Twitter in the time duration.

[1] www.twitter.com.
[2] www.weibo.com.

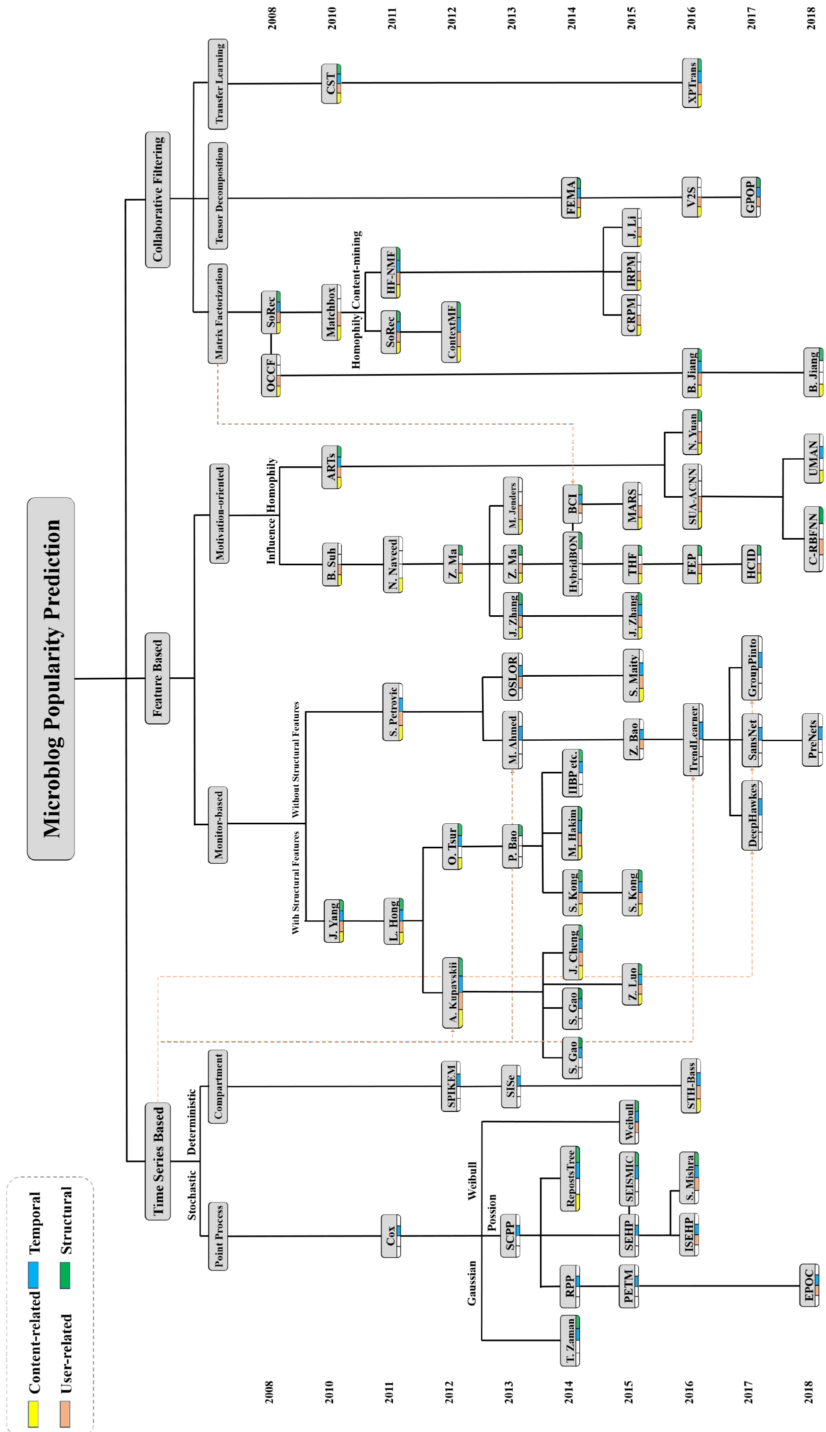

Fig. 4. A chronological tree of microblog popularity prediction.

Table 10. Data Statistics

Dataset	#Microblog	#User	#Following relationships	#Repost
Twitter	2,206,219	2,165,863	4,965,964,514	420,833
Weibo	300,000	1,776,950	308,489,739	23,755,810

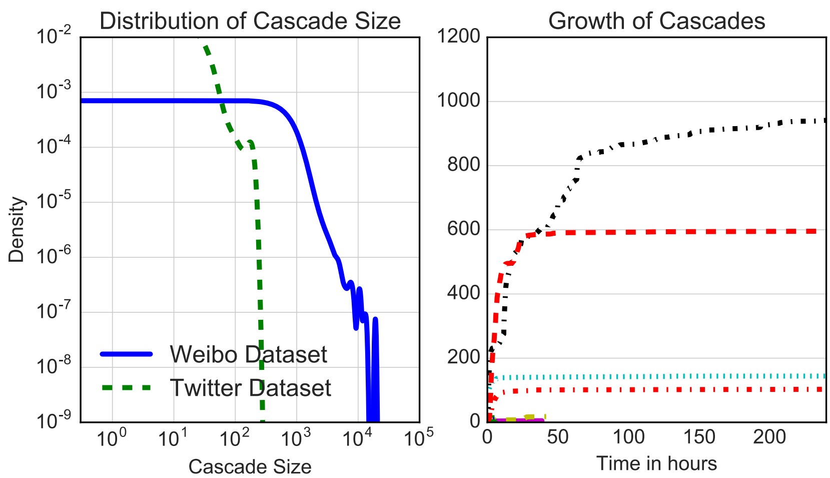

Fig. 5. Distribution of cascades and their growth.

7.1.2 Weibo. The Weibo dataset is from Zhang et al. [98] and is publicly available online.[3] The dataset contains 1.7 million users, 0.3 billion following relationships, and 300,000 microblogs (including tweets and retweets). All user profiles including name, gender, verification status, #bi-following, #followers, #followees, and #microblogs were crawled. The 300,000 microblogs are the most reposted microblogs involving this set of users. The basic statistics of the two datasets are listed in Table 10.

In accordance with many previous works, we observe a power law in the distribution of the size of cascades as shown in the left of Figure 5. Social networks are highly uneven and few users are under the spotlight of the crowd. There are also notable distinctions between the two datasets due to the different data collection procedure: the Weibo dataset contains large cascades up to the size of 10,000 and cascade sizes of 100 are quite common; on the other hand, the Twitter dataset rarely has cascades that grow over 100 and most cascades have under 10 reposts. The right of Figure 5 shows the growth patterns of some cascades. Every line represents an item in Weibo and their repost count is shown up to 10 days. We can see that the growth patterns vary greatly, but generally the growth slows down over time.

7.2 Prediction Methods and Data Input

We list the prediction methods used in the experimental evaluation and compare the types of data needed for each method in Table 11.

A total of eight feature-based methods were implemented with traditional machine-learning frameworks, and two of the best were selected to represent this category in comparison with others, since the primary focus of this type of methods is not on the machine-learning model,

[3]arnetminer.org/Influencelocality.

Table 11. Need for Data of Prediction Methods

Method	User activity history	User profile	Friend network	Item content	Repost time	Repost count
Feature-based	∘	∘	∘	∘	∘	∘
RPP[74]	×	×	×	×	✓	✓
Weibull[94]	×	×	×	×	✓	✓
SEISMIC[104]	×	✓	×	×	✓	✓
STH-Bass[87]	×	✓	×	✓	×	✓
OCCF[67]	✓	×	✓	✓	×	✓
OSLOR[17]	✓	×	×	×	×	✓

but the types of features used. For each method, we performed classification by using either all types of features or temporal features only. The reason for differentiating temporal features is two-fold: in previous literature, temporal features have been reported to be the most effective; using only temporal features gives us the common ground to compare with time-series-based methods. For time-series-based methods, four models were implemented, among them three point process models and one epidemic-based model. The Hawkes model was not selected as SEISMIC is derived from it. For matrix factorization methods, collaborative filtering methods formulate the prediction problem as the prediction of the repost action of individual users. Preliminary attempts show that when used for cascade size prediction, they tend to be conservative and greatly underestimate the size of cascades. Treating users as sensors is an unique idea displayed in the OSLOR model and we use this model for comparison. We parallel analyse the performance of OSLOR though it is theoretically classified as a feature-based method.

As seen in the Table 11, feature-based methods are quite flexible in terms of the data needed. In general, time-series models are based on temporal data. However, STH-Bass incorporates user data and content data, while SEISMIC needs the user's follower number. This provides them with additional information about the context of the cascade. Collaborative filtering methods need the user-item repost matrix as the basis for prediction. This matrix, in essence, is the activity history of the users. They rely on the similarity between users and posts which is hindered by the sparseness of the user-item interaction matrix.

7.3 Experiment Methodology

Prediction involves peeking into the initial stages of the cascade diffusion. Compared to setting a certain percentage of the data or a threshold for repost count as the peeking stage, using the time elapsed since posting is a more natural measure and also more practical in real life. We select the observation interval to be 1 hour, starting from the first hour after posting and up to 24 hours. Smaller cascades easily reach their final repost count in hours after posting, specifically, in the Twitter dataset, 75% of the cascades reach their final repost count after 1 day.

For time-series methods, the prediction of cascades are independent. We use the repost timestamps up to the observation time as input and predict the final number of reposts. As for the Bass model, the input is the accumulated number of reposts at these timestamps.

For feature-based methods and collaborative filtering methods, we randomly select 80% of the cascades to serve as training data in order to predict the remaining 20% unseen cascades. The peeking strategy is then applied to the unseen cascades.

For the classification methods in our comparison study, we adopt the most common way of labelling a post as popular or not: selecting a repost count threshold τ. In our implementation, τ is selected as the minimum number of reposts needed to reach the top 1% percentile. Given this

Table 12. Accuracy of Regression—Weibo Dataset

Time	6h		12h		18h		24h	
Methods	MSE	MAPE	MSE	MAPE	MSE	MAPE	MSE	MAPE
SEISMIC	**19.488**	**0.163**	**14.053**	**0.118**	**11.558**	**0.098**	11.251	**0.095**
RPP	389.087	0.821	328.423	0.794	275.442	0.763	263.235	0.751
Weibull model	80.787	0.375	36.000	0.259	18.195	0.195	**10.930**	0.140
STH-Bass	203.626	0.378	124.689	0.286	88.957	0.238	67.472	0.208
OCCF	660.038	0.969	692.630	0.964	692.559	0.961	692.442	0.956

Table 13. Accuracy of Regression—Twitter Dataset

Time	6h		12h		18h		24h	
Methods	MSE	MAPE	MSE	MAPE	MSE	MAPE	MSE	MAPE
SEISMIC	1.000	0.393	1.000	0.304	1.000	0.216	0.630	0.203
RPP	1.000	0.306	1.000	0.216	0.924	0.165	0.724	0.129
Weibull model	0.339	0.582	0.339	0.582	0.339	0.582	0.339	0.582
STH-Bass	**0.000793**	**0.0257**	**0.000660**	**0.0218**	**0.000601**	**0.0206**	**0.000542**	**0.0199**
OCCF	4.000	0.997	4.000	0.996	4.000	0.996	4.000	0.996

threshold, the methods that are proposed for the regression task can also be applied to classification problems by comparing the predicted final repost count R_i and the threshold τ.

8 EVALUATION RESULTS

8.1 Accuracy

Two sets of evaluation metrics are used for the regression problem and the classification problem. We use median APE and median SE to evaluate the accuracy of the regression task. Compared to mean APE and R^2, the median is known to be stable in the existence of extreme values. We will discuss these extreme values in the robustness section.

$$\mathrm{SE}(t) = (\hat{R}_i - R_i)^2,$$
$$\mathrm{APE}(t) = \frac{|\hat{R}_i - R_i|}{R_i}.$$

The accuracy of regression on Weibo and Twitter dataset is listed in Tables 12 and 13, respectively. Bold face values represent the optimal result w.r.t. the corresponding metric. The results of OCCF on the two metrics are also recorded and shown because we can obtain the final repost counts (R_i) from OCCF model by counting the predicted reposters at each prediction time. The median SE-time curves and median APE-time curves of different methods are shown in Figures 6 and 7.

For the classification task, we rank the cascades by repost count and select the top 1% percentile as popular cascades. Under this skewed classification setting, it is easy to achieve a high accuracy by making negative predictions, so we employ precision, recall, and F_1 score as evaluation metrics.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$
$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$
$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}.$$

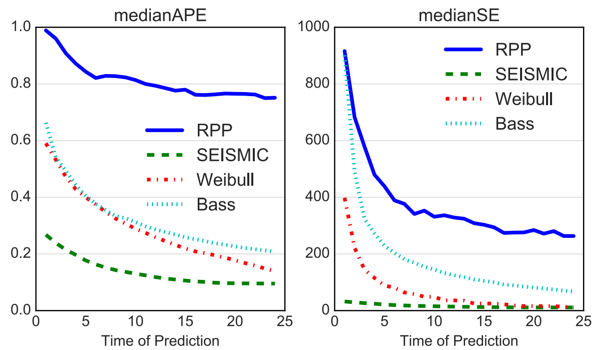

Fig. 6. Regression metrics on Weibo dataset.

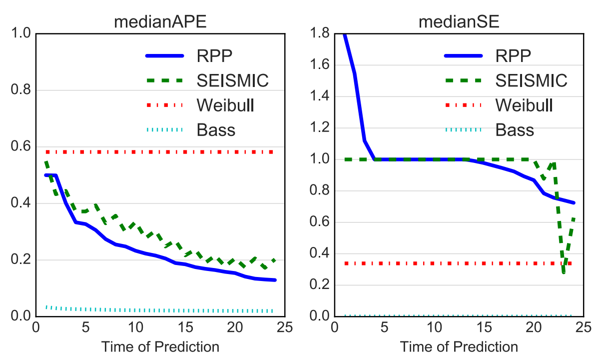

Fig. 7. Regression metrics on Twitter dataset.

Table 14. Accuracy of Classification on Weibo Dataset

Time	6h			12h			18h			24h		
Methods	Precision	Recall	F1	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1
SEISIC	**0.748**	**0.771**	**0.759**	**0.768**	**0.805**	**0.786**	0.787	**0.848**	**0.817**	0.774	**0.899**	**0.832**
RPP	0.574	0.443	0.530	0.652	0.464	0.549	**0.812**	0.455	0.579	**0.810**	0.432	0.598
Weibull model	0.579	0.222	0.321	0.636	0.350	0.452	0.658	0.520	0.581	0.699	0.65	0.674
STH-Bass	0.100	0.424	0.162	0.111	0.490	0.181	0.140	0.580	0.225	0.217	0.68	0.329
OSLOR	0.707	0.182	0.289	0.731	0.247	0.369	0.736	0.291	0.417	0.736	0.334	0.459
Feature-based(All): DT	0.434	0.421	0.427	0.468	0.441	0.452	0.483	0.476	0.480	0.445	0.434	0.437
Feature-based(All): RF	0.607	0.314	0.413	0.577	0.308	0.401	0.646	0.378	476	0.617	0.315	0.417
Feature-based(Temporal): DT	0.415	0.420	0.417	0.425	0.423	0.424	0.457	0.460	0.458	0.444	0.420	0.431
Feature-based(Temporal): RF	0.537	0.244	0.334	0.579	0.260	0.358	0.624	0.295	0.400	0.592	0.268	0.367

TP, FP, and FN are number of true positive instances, number of false-positive instances, and number of false-negatives instances, respectively. It is noteworthy that this set of metrics is also applicable to regression methods (e.g., time-series-based methods) by converting the regression outcomes into binary values, that is, comparing the final count R_i and popularity threshold τ. This makes the two types of methods comparable and facilitates the general comparison. The accuracy of classification is listed in Tables 14 and 15. The precision curves, recall curves, and F_1 curves of different methods are shown in Figures 8 and 9.

Table 15. Accuracy of Classification on Twitter Dataset

Time	6h			12h			18h			24h		
Methods	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1
SEISMIC	0.785	**0.516**	**0.623**	0.766	**0.595**	0.670	0.802	0.639	0.711	0.742	0.683	0.711
RPP	0.574	0.443	0.500	0.652	0.464	0.542	0.812	0.455	0.583	0.810	0.432	0.564
Weibull model	**0.921**	0.385	0.543	**0.927**	0.560	**0.699**	**0.937**	**0.648**	**0.766**	**0.940**	**0.692**	**0.797**
STH-Bass	0.061	0.396	0.106	0.077	0.516	0.133	0.086	0.582	0.151	0.095	0.648	0.165
OSLOR	0.523	0.211	0.301	0.531	0.220	0.311	0.553	0.218	0.313	0.630	0.222	0.328
Feature-based(All): DT	0.472	0.452	0.461	0.475	0.457	0.465	0.526	0.521	0.523	0.486	0.470	0.477
Feature-based(All): RF	0.609	0.361	0.453	0.628	0.375	0.469	0.645	0.405	0.497	0.620	0.345	0.442
Feature-based(Temporal): DT	0.453	0.453	0.452	0.461	0.475	0.468	0.489	0.491	0.490	0.465	0.436	0.449
Feature-based(Temporal): RF	0.605	0.302	0.402	0.643	0.355	0.457	0.680	0.390	0.495	0.628	0.313	0.416

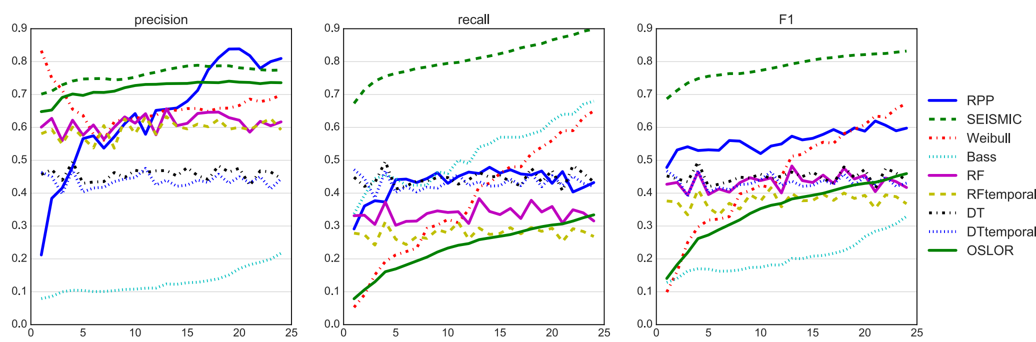

Fig. 8. Classification metrics on Weibo dataset.

Regression task. For the regression task, SEISMIC achieves the best accuracy on the Weibo dataset and next to best accuracy on Twitter. For the Weibo dataset, the performance of the Weibull model and the Bass model improve greatly with time and by 24 hours, the accuracy of the Weibull model is comparable to that of SEISMIC. The Weibull model and Bass model generally remains steady over time on the Twitter dataset which is consisted mostly of small cascades. The Bass model achieves very impressive accuracy on the Twitter dataset but only average performance on the Weibo dataset. RPP also performs better on the Twitter dataset. This suggests that Bass and RPP might be more suitable for the prediction of small cascades.

Classification task. For the classification task, feature-based methods have a moderate performance and do not see significant improvement over time on both datasets. The advantage of including all types of features over using only temporal features is not significant. As Decision Trees and Random Forest outperform the rest in terms of *F*1 score, we choose these two methods as representatives when comparing with other types of prediction methods in Table 15.

As for other methods in classification task, SEISMIC performs very well on the Weibo dataset, but displays fluctuation in recall on the Twitter dataset. After inspecting the results, we believe this is due to the estimate of p_t being near the critical threshold. Once this threshold is exceeded, SEISMIC will predict the cascade to grow into infinity and such values are considered as invalid. The accuracy of time-series models improve with time as with the regression task. RPP improves quickly in terms of precision but has a moderate recall value. Bass and Weibull improve in terms of recall. However, Bass maintains low precision, while Weibull's precision improve with time,

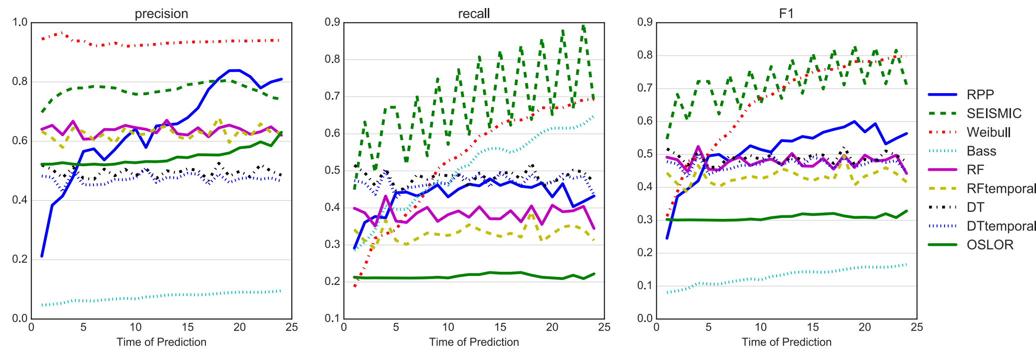

Fig. 9. Classification metrics on Twitter dataset.

Table 16. Accuracy of Classification Feature Based on Weibo Data (All Features)

Time	6h			12h			18h			24h		
Method	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1
DT	0.434	0.421	**0.427**	0.468	0.441	**0.452**	0.483	0.476	**0.48**	0.445	0.434	**0.437**
LR	0.679	0.142	0.232	0.636	0.203	0.308	0.644	0.223	0.331	0.628	0.194	0.296
NB	0.26	**0.654**	0.37	0.295	**0.547**	0.38	0.249	**0.745**	0.373	0.289	**0.614**	0.388
GLR	0.554	0.107	0.179	0.601	0.121	0.2	0.591	0.148	0.236	0.579	0.133	0.217
RF	0.607	0.314	0.413	0.577	0.308	0.401	**0.646**	0.378	0.476	0.617	0.315	0.417
KNN	0.297	0.257	0.275	0.321	0.292	0.305	0.327	0.297	0.311	0.3	0.273	0.286
NN	**0.706**	0.061	0.112	0.51	0.048	0.088	0.554	0.138	0.221	0.692	0.124	0.21
SVM	0.325	0.546	0.407	**0.684**	0.197	0.306	0.58	0.262	0.36	**0.806**	0.142	0.242

Table 17. Accuracy of Classification Feature Based on Weibo Data (Temporal Features)

Time	6h			12h			18h			24h		
Method	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1
DT	0.416	**0.42**	**0.417**	0.425	**0.423**	**0.424**	0.457	**0.46**	**0.458**	0.444	**0.42**	**0.431**
LR	**0.685**	0.133	0.221	0.64	0.195	0.299	0.668	0.224	0.335	0.632	0.18	0.278
NB	0.667	0.053	0.099	0.561	0.041	0.076	0.65	0.049	0.091	0.722	0.051	0.094
GLR	0.597	0.112	0.188	**0.587**	0.109	0.184	0.595	0.147	0.236	0.631	0.121	0.202
RF	0.537	0.244	0.334	0.579	0.26	0.358	0.624	0.295	0.4	0.593	0.268	0.368
KNN	0.163	0.161	0.161	0.161	0.163	0.162	0.194	0.187	0.19	0.192	0.199	0.195
NN	0.625	0.101	0.174	0.6	0.086	0.15	0.579	0.147	0.235	0.619	0.06	0.109
SVM	0.462	0.232	0.309	0.568	0.106	0.179	**0.696**	0.164	0.266	**0.864**	0.093	0.168

making Weibull more preferable for the classification task. OSLOR performs better on the Weibo dataset with high precision but has the lowest recall on both datasets.

Other than Decision Trees and Random Forest, we have also conducted experiments on other six types of machine-learning methods. We list the accuracy of these feature-based methods in Tables 16–19. From these tables, we can see that in most cases, the *F*1 score of Decision Trees and Random Forest are the highest. Although some machine-learning models may be able to achieve high scores for one metric, they often fail to get high scores for the other, leading to a mediocre *F*1 score.

Table 18. Accuracy of Classification Feature Based on Twitter Data (All Features)

Time	6h			12h			18h			24h		
Method	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1
DT	0.472	0.452	**0.461**	0.475	0.457	0.465	0.526	0.521	**0.523**	0.486	0.47	**0.477**
LR	**0.697**	0.172	0.272	0.638	0.254	0.363	**0.651**	0.264	0.376	**0.628**	0.233	0.339
NB	0.289	**0.658**	0.4	0.331	**0.557**	0.413	0.275	**0.773**	0.405	0.325	**0.59**	0.417
GLR	0.585	0.148	0.235	0.631	0.171	0.268	0.622	0.181	0.28	0.609	0.164	0.258
RF	0.609	0.361	0.453	0.628	0.375	**0.469**	0.645	0.405	0.497	0.62	0.345	0.442
KNN	0.328	0.289	0.307	0.351	0.322	0.335	0.362	0.324	0.342	0.347	0.316	0.331
NN	0.615	0.175	0.272	0.556	0.133	0.215	0.573	0.232	0.331	0.523	0.091	0.155
SVM	0.328	0.657	0.437	**0.688**	0.231	0.346	0.56	0.385	0.456	0.545	0.312	0.397

Table 19. Accuracy of Classification Feature Based on Twitter Data (Temporal Features)

Time	6			12			18			24		
Method	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1	Precision	Recall	*F*1
DT	0.453	**0.453**	**0.452**	0.461	**0.475**	**0.468**	0.489	**0.491**	0.49	0.465	**0.436**	**0.448**
LR	0.71	0.17	0.27	**0.661**	0.257	0.37	0.678	0.264	0.38	0.627	0.225	0.33
NB	**0.715**	0.054	0.1	0.641	0.044	0.082	**0.681**	0.046	0.086	**0.743**	0.051	0.096
GLR	0.656	0.142	0.233	0.642	0.164	0.261	0.645	0.176	0.277	0.63	0.153	0.246
RF	0.605	0.302	0.402	0.643	0.355	0.457	0.68	0.39	**0.495**	0.628	0.313	0.416
KNN	0.197	0.199	0.197	0.204	0.204	0.203	0.221	0.215	0.217	0.216	0.222	0.219
NN	0.597	0.162	0.254	0.494	0.178	0.261	0.593	0.264	0.365	0.585	0.151	0.24
SVM	0.436	0.36	0.394	0.63	0.148	0.24	0.653	0.277	0.389	0.75	0.143	0.24

8.2 Timeliness

Using the time elapsed since posting for the peeking stage makes it possible for us to compare the timeliness of prediction. The time-series methods and user-based method improve their accuracy with time, but the effect seems to be most evident with the Weibull model, Bass model, and RPP model. These methods need more observation of the cascade to achieve a competitive predictive result. The accuracy of feature-based methods is nearly time-invariant.

In the regression task, SEISMIC displays its advantage for the Weibo dataset from the beginning and Bass for Twitter. For the classification task though, SEISMIC has early advantage on both of the datasets. The accuracy of the Weibull models grows very fast on both datasets, and might be preferred over SEISMIC in the Twitter dataset due to its steady performance. It is noteworthy that Weibull, Bass, and OSLOR do not perform well in the first few hours, so feature-based methods can be used for early prediction.

8.3 Robustness and Extreme Values

In this section, we discuss the stability of prediction. During our empirical evaluation, we find that some of the prediction values are unreasonably large, greatly affecting the mean APE and mean SE. In application, we tend to choose prediction methods that not only perform well on average, but also do not make serious misjudgements.

We measure the robustness of a prediction method with respect to the distribution of its PE. We adopt an intuitive metric—the percentage of predictions that have APE > 200% and a statistic

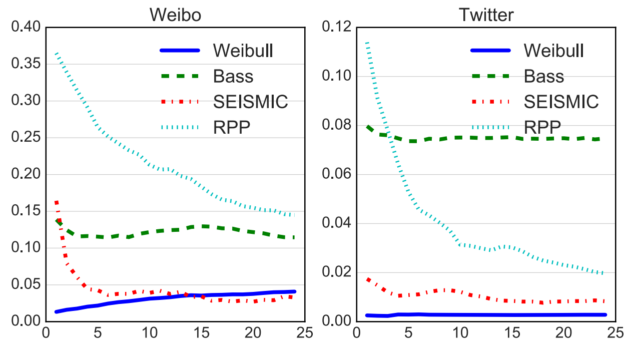

Fig. 10. APE>200%.

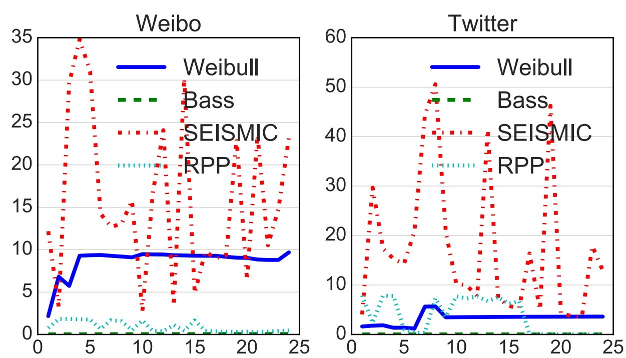

Fig. 11. Kurtosis/10^3.

metric—kurtosis. Kurtosis measures the heavy tail of the distribution. A large kurtosis means that the variance is mainly contributed by few extreme values.

$$\text{Kurtosis } \kappa(X) = \frac{\frac{1}{n}\Sigma(X-\mu)^4}{(\frac{1}{n}\Sigma(X-\mu)^2)^2} - 3.$$

It is noteworthy that SEISMIC and RPP may produce infinite prediction values, such values are included in the APE>200% metric but not in the kurtosis metric.

SEISMIC and RPP, respectively, have a mean of 2.25% and 0.0980% of infinite values over all prediction time points.

By looking at Figure 10, we observe that RPP has a high amount of inaccurate predictions. The proportion drops rapidly with time but is still larger than that of other methods. The Bass model produces a substantial amount of inaccurate predictions as well. The Weibull model and SEISMIC model do not produce many extreme values.

Referring to Figure 11, we see that SEISMIC has a much higher kurtosis than the rest, which implies that the variance of SEISMIC is due to the existence of extreme values.

9 ANALYSIS AND DISCUSSION

9.1 Efficiency

We divide the process of popularity prediction into three stages: the data preprocessing stage, the training stage and the prediction stage. For data preprocessing, we rank the time efficiency into two categories: high complexity (may take hours or days to get the result) and low complexity (takes

Table 20. Efficiency of Methods

Methods	Data preprocessing	Model training	Predicting
SEISMIC	High	Single	Single
RPP	High	Single	Single
Weibull model	High	Single	Single
STH-Bass	High	Single	Single
OCCF	Low	Batch	Batch
OSLOR	High	Batch	Single
Feature-based: Temporal	High	Batch	Single
Collaborative filtering	Low	Batch	Single

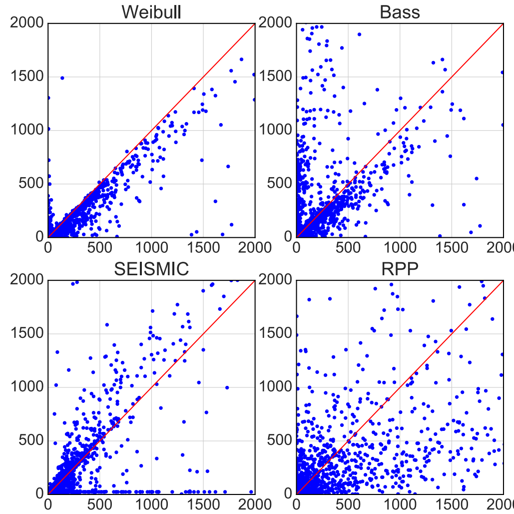

Fig. 12. Comparing the predicted size to actual cascade size on Weibo dataset.

minutes to get the result). For the training phase and prediction phase, we consider whether items are processed in batch or individual fashion.

Results of the comparison are listed in Table 20. For OCCF, all stages are performed batchwise. It is both high in time complexity and space complexity and requires pre-trained content topics as input. The feature-based method that use all features is time-consuming due to network features and content features extraction. Feature-based methods including OSLOR model train the model in batch and predict cascade sizes individually. Time-series methods need nearly no preprocessing and parameters are learned and used for prediction on an individual basis.

For the preprocessing stage, we can possibly take advantage of parallel computing since features are generally calculated on an user basis or item basis. For the training stage, many machine-learning models can use stochastic gradient descent for optimization instead of batch gradient descent to achieve speedup.

9.2 Bias

By plotting the predicted size of cascades (y-axis) against the actual size (x-axis) as in Figures 12 and 13, we discover that some of the prediction methods display bias. Specifically, the Weibull model produces pessimisic predictions. The Bass model, on the other hand, tends to overestimate

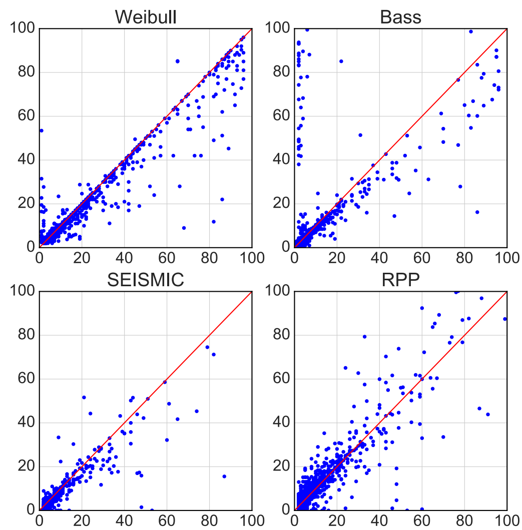

Fig. 13. Comparing the predicted size to actual cascade size on Twitter dataset.

the final size of small cascades. The reason why Bass tends to predict optimistically may be that Bass model does not have a explicit time decay factor and the slow down in diffusion speed is actually due to the bound. RPP does not display any tendency but shows high variance. SEISMIC also produces optimistic results, possibly due to its self-exciting property.

9.3 Recommendation

SEISMIC can achieve relatively high accuracy most of the time. However, it may produce a considerable amount of extreme values. If an adequate proportion of the cascades is already observed, the Weibull model and RPP may perform equally well or slightly better than SEISMIC on the classification task. The Bass model and Weibull model are suitable for regression at a later stage. OSLOR is not recommended for its low recall and its low space efficiency, especially the popularity prediction in large community since it requires more space to train the model. Features-based methods have a moderate performance and may be used in the absence of temporal data. However, when we use structural features to predict the popularity, space efficiency and time efficiency are important factors that need to be considered.

Summarizing, we recommend using SEISMIC as the major prediction method and use Weibull and Bass as a reference to avoid extreme values.

9.4 Extended Discussion on Predictions in Social Network

The prediction power of social network data is not limited to predicting the future popularity of user-generated items. Single items can be aggregated into topic and events, and prediction can be applied to these topics and events as a whole. For example, [51], [93], and [42] were originally designed to track the popularity of topics. Moreover, aggregate perspective enables cross-platform popularity study and prediction. Gao et al. [23] focused on the data correlation between different platforms and provided an analytical method. Liao et al. [50], likewise, proposed a cross-platform model to predict the popularity of events on one platform based on the information of another one. Users can also be aggregated into communities. Hu et al. [31] takes on the perspective of community level diffusion, based on the 'Strength of Weak Ties' Theory which suggests that inter-communities interactions play a critical role in diffusion.

Chatter from social networks has also proved to be able to predict real-world outcomes. Asur Huberman [3] used the rate of Twitter mentions to predict the box office revenue of movies. Although only a linear model was built, the results outperformed market-based predictors. As social network posts can reflect the attitude of the investors, many works have attempted to predict financial trends by using sentiment or mood extracting from social networks. Huang et al. [32] ultilized Granger causality analysis to select the moods that were most significantly correlated with stock indexes, then used the selective hidden Markov model to predict the stock trend. Nguyen and Shirai [66] extended the LDA model to account for sentiment and topic at the same time, acknowledging that the same word could imply different sentiment under different topics. After extracting the sentiment, the SVM model is used to predict the trend. Social network data has been applied to the prediction of user activities outside the network, such as volunteer tendency in [76].

Under the setting of location-based social networks, user's locations can be predicted given the locations of his/her friends [61] or the user's profile [15]. Social network activity has also been applied to traffic prediction. Traffic indicators were extracted from tweets and served as input for a linear regression model in [26]. Social networks also made it possible to predict the transmission of epidemic diseases on a microscopic level. Using the states of Twitter friends as features, Sadilek et al. [72] designed a conditional random field model to predict whether and when a user would get sick.

Summing up, we can categorize prediction in social networks according to the target. The popularity of single items, hashtags, and topics can be seen as prediction within the network, as oppose to user location, traffic, disease, or stock market trends which exist outside the social network.

10 CONCLUSION

In this article, we set out to compare methods for popularity prediction on social networks, specifically the prediction of single microblogs, by establishing a taxonomy and evaluating the performance under a unified testing scheme. For the taxonomy, we divide these prediction methods into three categories—feature-based, time-series-based, and collaborative filtering—and analyse them, respectively. We also take on another angle and categorize these methods into motivation-oriented or monitor-based. Motivation-oriented methods attempt to model the motivation behind reposting actions, which is either the internal motivation or external influence. Monitor-based methods focus on observing the initial responses of users once the item is posted.

We conduct experiments on two real-world datasets. Our results show that temporal data has the most predictive power. This is further amplified by the use of time-series models. Although feature-based methods do not achieve the best performance and have quite heavy overhead, this is the only type of method that can be used in absence of temporal data and their performance is stable when the prediction horizons changes. User-based methods are not suitable for very sparse user-item interactions, which is unfortunately the case for social networks.

From an empirical point of view, using only temporal data with time-series models might be the most effective and efficient way of predicting popularity at the moment. However, if we wish to deepen our understanding of popular items on social networks, we encourage future researchers to combine features with time series to account for the contextual differences of cascades. For feature-based approaches, as they take into consideration all the types of data available in social networks, we believe that the key may lie in integrating heterogeneous data and expressing their connections.

ACKNOWLEDGMENTS

X. Gao, Z. Cao, B. Yao, and G. Chen are with the Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University.

The authors would like to thank the reviewers for their insightful comments. The authors would also like to thank Zexin Chen, Cheng Chang and Yan Yan for their contributions to the early versions of this article. S. Li completed this work when she was an undergraduate student of Shanghai Jiao Tong University, China.

REFERENCES

[1] Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. 2013. A peek into the future: Predicting the evolution of popularity in user generated content. In *WSDM*. 607–616.

[2] Sultan Alzahrani, Saud Alashri, Anvesh Koppela, Hasan Davulcu, and Ismail Toroslu. 2015. A network-based model for predicting hashtag breakouts in Twitter. In *SBP*. 3–12.

[3] Sitaram Asur and Bernardo A. Huberman. 2010. Predicting the future with social media. In *Web Intelligence*. 492–499.

[4] Eytan Bakshy, Jake Hofman, Winter Mason, and Duncan Watts. 2011. Everyone's an influencer: Quantifying influence on Twitter. In *WSDM*. 65–74.

[5] Peng Bao. 2016. Modeling and predicting popularity dynamics via an influence-based self-excited Hawkes process. In *CIKM*. 1897–1900.

[6] Peng Bao, HuaWei Shen, Junming Huang, and XueQi Cheng. 2013. Popularity prediction in microblogging network: A case study on sina weibo. In *WWW*. 177–178.

[7] Peng Bao, Hua-Wei Shen, Xiaolong Jin, and Xue-Qi Cheng. 2015. Modeling and predicting popularity dynamics of microblogs using self-excited Hawkes processes. In *WWW*. 9–10.

[8] Z. M. Bao, Y. Liu, F. Xiong, and Y. X. Zhu. 2015. A new method to predict the pupularity of the microblog. In *CCITSA*. 119–123.

[9] Frank M Bass. 1969. A new product growth for model consumer durables. *Management Science* 15, 5 (1969), 215–227.

[10] Jingwen Bian, Yang Yang, and Tat-Seng Chua. 2014. Predicting trending messages and diffusion participants in microblogging network. In *SIGIR*. 537–546.

[11] Mireia Bolíbar. 2016. Macro, meso, micro: Broadening the 'social' of social network analysis with a mixed methods approach. *Quality & Quantity* 50, 5 (2016), 2217–2236.

[12] George E. P. Box and David R Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society* 26, 2 (1964), 211–252.

[13] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. 2017. DeepHawkes: Bridging the gap between prediction and understanding of information cascades. In *CIKM*. 1149–1158.

[14] Qi Cao, Huawei Shen, Hao Gao, Jinhua Gao, and Xueqi Cheng. 2017. Predicting the popularity of online content with group-specific models. In *WWW*. 765–766.

[15] Wipada Chanthaweethip, Xiao Han, Noël Crespi, Yuanfang Chen, Reza Farahbakhsh, and Ángel Cuevas. 2013. "Current city" prediction for coarse location based applications on Facebook. In *GLOBECOM*. 3188–3193.

[16] Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. 2014. Can cascades be predicted?. In *WWW*. 925–936.

[17] Peng Cui, Shifei Jin, Linyun Yu, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2013. Cascading outbreak prediction in networks: A data-driven approach. In *SIGKDD*. 901–909.

[18] Peng Cui, Fei Wang, Shaowei Liu, Mingdong Ou, Shiqiang Yang, and Lifeng Sun. 2011. Who should share what?: Item-level social influence prediction for users and posts ranking. In *SIGIR*. 185–194.

[19] Flavio Figueiredo, Jussara Almeida, Marcos Gonçalves, and Fabrício Benevenuto. 2014. TrendLearner: Early prediction of popularity trends of user generated content. In *Information Sciences*. 172–187.

[20] Shuai Gao, Jun Ma, and Zhumin Chen. 2014. Effective and effortless features for popularity prediction in microblogging network. In *WWW*. 269–270.

[21] Shuai Gao, Jun Ma, and Zhumin Chen. 2014. Popularity prediction in microblogging network. In *APWeb*. 379–390.

[22] Shuai Gao, Jun Ma, and Zhumin Chen. 2015. Modeling and predicting retweeting dynamics on microblogging platforms. In *WSDM*. 107–116.

[23] Tianxiang Gao, Weiming Bao, Jinning Li, Xiaofeng Gao, Boyuan Kong, Yan Tang, Guihai Chen, and Xuan Li. 2018. DancingLines: An analytical scheme to depict cross-platform event popularity. In *International Conference on Database and Expert Systems Applications*. Springer, 283–299.

[24] Manuel Gomez-Rodriguez, Jure Leskovec, and Bernhard Schölkopf. 2013. Modeling information propagation with survival theory. In *ICML*. 666–674.

[25] M. A. N. Hakim and M. L. Khodra. 2014. Predicting information cascade on Twitter using support vector regression. In *ICODSE*. 1–6.

[26] Jingrui He, Wei Shen, Phani Divakaruni, Laura Wynter, and Rick Lawrence. 2013. Improving traffic prediction with tweet semantics. In *IJCAI*. 1387–1393.

[27] Bao-Thien Hoang, Kamel Chelghoum, and Imed Kacem. 2016. Modeling information diffusion via reputation estimation. In *DEXA*. 136–150.
[28] Minh X. Hoang, Xuan-Hong Dang, Xiang Wu, Zhenyu Yan, and Ambuj K. Singh. 2017. GPOP: Scalable group-level popularity prediction for online content in social networks. In *WWW*. 725–733.
[29] T. A. Hoang and E. P. Lim. 2016. Microblogging content propagation modeling using topic-specific behavioral factors. *IEEE Transactions on Knowledge and Data Engineering* 28, 9 (2016), 2407–2422.
[30] Liangjie Hong, Ovidiu Dan, and Brian D. Davison. 2011. Predicting popular messages in Twitter. In *WWW*. 57–58.
[31] Zhiting Hu, Junjie Yao, Bin Cui, and Eric P. Xing. 2015. Community level diffusion extraction. In *SIGMOD*. 1555–1569.
[32] Yifu Huang, Shuigeng Zhou, Kai Huang, and Jihong Guan. 2015. Boosting financial trend prediction with Twitter mood based on selective hidden Markov models. In *DASFAA*. 435–451.
[33] Tomoharu Iwata, Amar Shah, and Zoubin Ghahramani. 2013. Discovering latent influence in online social activities via shared cascade Poisson processes. In *SIGKDD*. 266–274.
[34] Maximilian Jenders, Gjergji Kasneci, and Felix Naumann. 2013. Analyzing and predicting viral tweets. In *WWW*. 657–664.
[35] Bo Jiang, Jiguang Liang, Ying Sha, Rui Li, Wei Liu, Hongyuan Ma, and Lihong Wang. 2016. Retweeting behavior prediction based on one-class collaborative filtering in social networks. In *SIGIR*. 977–980.
[36] Bo Jiang, Jiguang Liang, Ying Sha, and Lihong Wang. 2015. Message clustering based matrix factorization model for retweeting behavior prediction. In *CIKM*. 1843–1846.
[37] Bo Jiang, Zhigang Lu, Ning Li, Jianjun Wu, and Zhengwei Jiang. 2018. Retweet prediction using social-aware probabilistic matrix factorization. In *ICCS*. 316–327.
[38] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2012. Social contextual recommendation. In *CIKM*. 45–54.
[39] Meng Jiang, Peng Cui, Fei Wang, Xinran Xu, Wenwu Zhu, and Shiqiang Yang. 2014. FEMA: Flexible evolutionary multi-faceted analysis for dynamic behavioral pattern discovery. In *SIGKDD*. 1186–1195.
[40] Meng Jiang, Peng Cui, Nicholas Jing Yuan, Xing Xie, and Shiqiang Yang. 2016. Little is much: Bridging cross-platform behaviors through overlapped crowds. In *AAAI*. 13–19.
[41] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM Review* 51, 3 (2009), 455–500.
[42] Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye, and Zhe Zhao. 2014. Predicting bursts and popularity of hashtags in real-time. In *SIGIR*. 927–930.
[43] Shoubin Kong, Fei Ye, Ling Feng, and Zhe Zhao. 2015. Towards the prediction problems of bursting hashtags on Twitter. *Journal of the Association for Information Science and Technology* 66, 12 (2015), 2566–2579.
[44] Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, and Andrey Kustarev. 2012. Prediction of retweet cascade size over time. In *CIKM*. 2335–2338.
[45] Rupert Lemahieu, Steven Van Canneyt, Cedric De Boom, and Bart Dhoedt. 2015. Optimizing the popularity of Twitter messages through user categories. In *ICDMW*. 1396–1401.
[46] Kristina Lerman and Aram Galstyan. 2008. Analysis of social voting patterns on digg. In *WOSN*. 7–12.
[47] Chenliang Li, Aixin Sun, and Anwitaman Datta. 2012. Twevent: Segment-based event detection from tweets. In *CIKM*. 155–164.
[48] Jun Li, Jiamin Qin, Tao Wang, Yi Cai, and Huaqing Min. 2015. A collaborative filtering model for personalized retweeting prediction. In *DASFFA*. 122–134.
[49] Yiping Li, Zhuonan Feng, Hao Wang, Shoubin Kong, and Ling Feng. 2013. ReTweetp: Modeling and predicting tweets spread using an extended susceptible-infected- susceptible epidemic model. In *DASFFA*. 454–457.
[50] Mingding Liao, Xiaofeng Gao, Xuezheng Peng, and Guihai Chen. 2018. CROP: An efficient cross-platform event popularity prediction model for online media. In *DEXA*. 35–49.
[51] Weiwei Liu, Zhi-Hong Deng, Xiuwen Gong, Frank Jiang, and Ivor W. Tsang. 2015. Effectively predicting whether and when a topic will become prevalent in a social network. In *AAAI*. 210–216.
[52] Yanbing Liu, Jinzhe Zhao, and Yunpeng Xiao. 2018. C-RBFNN: A user retweet behavior prediction method for hotspot topics based on improved RBF neural network. *Neurocomputing* 275 (2018), 733–746.
[53] Xinjiang Lu, Zhiwen Yu, Bin Guo, and Xingshe Zhou. 2014. Predicting the content dissemination trends by repost behavior modeling in mobile social networks. *Journal of Network and Computer Applications* 42 (2014), 197–207.
[54] Zhilin Luo, Yue Wang, Xintao Wu, Wandong Cai, and Ting Chen. 2015. On burst detection and prediction in retweeting sequence. In *PAKDD*. 96–107.
[55] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: Social recommendation using probabilistic matrix factorization. In *CIKM*. 931–940.
[56] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*. 287–296.

[57] Zongyang Ma, Aixin Sun, and Gao Cong. 2012. Will this# hashtag be popular tomorrow? In *SIGIR*. 1173–1174.

[58] Zongyang Ma, Aixin Sun, and Gao Cong. 2013. On predicting the popularity of newly emerging hashtags in Twitter. *Journal of the Association for Information Science and Technology* 64, 7 (2013), 1399–1410.

[59] Suman Maity, Abhishek Gupta, Pawan Goyal, and Animesh Mukherjee. 2015. A stratified learning approach for predicting the popularity of Twitter idioms. In *ICWSM*. 642–645.

[60] Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, and Christos Faloutsos. 2012. Rise and fall patterns of information diffusion: Model and implications. In *SIGKDD*. 6–14.

[61] Jeffrey McGee, James Caverlee, and Zhiyuan Cheng. 2013. Location prediction in social media based on tie strength. In *CIKM*. 459–468.

[62] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 27, 1 (2001), 415–444.

[63] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. 2016. Feature driven and point process approaches for popularity prediction. In *CIKM*. 1069–1078.

[64] George O. Mohler, Martin B. Short, P. Jeffrey Brantingham, Frederic Paik Schoenberg, and George E. Tita. 2012. Self-exciting point process modeling of crime. *Journal of the American Statistical Association* 106, 493 (2012), 100–108.

[65] Nasir Naveed, Thomas Gottron, Jérôme Kunegis, and Arifah Che Alhadi. 2011. Bad news travel fast: A content-based analysis of interestingness on Twitter. In *WebSci*. 1–7.

[66] Thien Hai Nguyen and Kiyoaki Shirai. 2015. Topic modeling based sentiment analysis on social media for stock market prediction. In *ACL*. 1354–1364.

[67] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *ICDM*. 502–511.

[68] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, vol. 10, 230–235.

[69] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2011. RT to win! predicting message propagation in Twitter. In *ICWSM*. 586–589.

[70] Rene Pfitzner, Antonios Garas, and Frank Schweitzer. 2012. Emotional divergence influences information spreading in Twitter. In *ICWSM*. 2–5.

[71] Daniel M. Romero, Chenhao Tan, and Johan Ugander. 2013. On the interplay between social and topical structure. In *Seventh International AAAI Conference on Weblogs and Social Media*.

[72] Adam Sadilek, Henry A. Kautz, and Vincent Silenzio. 2012. Predicting disease transmission from geo-tagged microblog data. In *AAAI*. 136–142.

[73] Amit Sharma and Dan Cosley. 2016. Distinguishing between personal preferences and social influence in online activity feeds. In *CSCW*. 1091–1103.

[74] Hua-Wei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and predicting popularity dynamics via reinforced Poisson processes. In *AAAI*. 291–297.

[75] Benjamin Shulman, Amit Sharma, and Dan Cosley. 2016. Predictability of popularity: Gaps between prediction and understanding. In *ICWSM*. 348–357.

[76] Xuemeng Song, Liqiang Nie, Luming Zhang, Mohammad Akbari, and Tat-Seng Chua. 2015. Multiple social network learning and its application in volunteerism tendency prediction. In *SIGIR*. 213–222.

[77] Karthik Subbian, B. Aditya Prakash, and Lada Adamic. 2017. Detecting large reshare cascades in social networks. In *WWW*. 597–605.

[78] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H. Chi. 2010. Want to be retweeted? Large scale analytics on factors impacting retweet in Twitter network. In *SOCIALCOM*. 177–184.

[79] Gábor Szabó and Bernardo A. Huberman. 2010. Predicting the popularity of online content. *Communications of the ACM* 53, 8 (2010), 80–88.

[80] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2011. Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology* 62, 2 (2011), 406–418.

[81] Oren Tsur and Ari Rappoport. 2012. What's in a hashtag? Content based prediction of the spread of ideas in microblogging communities. In *WSDM*. 643–652.

[82] Duy Quang Vu, Arthur U. Asuncion, David R. Hunter, and Padhraic Smyth. 2011. Dynamic egocentric models for citation networks. In *ICML*. 857–864.

[83] Senzhang Wang, Zhao Yan, Xia Hu, Philip S. Yu, and Zhoujun Li. 2015. Burst time prediction in cascades. In *AAAI*. 325–331.

[84] Wen Wang, Wei Zhang, Jun Wang, Junchi Yan, and Hongyuan Zha. 2018. Learning sequential correlation for user generated textual content popularity prediction. In *IJCAI*. 1625–1631.

[85] Qitian Wu, Chaoqi Yang, Hengrui Zhang, Xiaofeng Gao, Paul Weng, and Guihai Chen. 2018. Adversarial training model unifying feature driven and point process perspectives for event popularity prediction. In *CIKM*. 517–526.

[86] Shanchan Wu and Louiqa Raschid. 2014. Prediction in a microblog hybrid network using bonacich potential. In *WSDM*. 383–392.
[87] Yan Yan, Zhaowei Tan, Xiaofeng Gao, Shaojie Tang, and Guihai Chen. 2016. STH-Bass: A Spatial-temporal heterogeneous bass model to predict single-tweet popularity. In *DASFAA*. 18–32.
[88] Chaoqi Yang, Qitian Wu, Xiaofeng Gao, and Guihai Chen. 2018. EPOC: A survival perspective early pattern detection model for outbreak cascades. In *DEXA*. 336–351.
[89] Jiang Yang and Scott Counts. 2010. Predicting the speed, scale, and range of information diffusion in Twitter. In *ICWSM*. 355–358.
[90] M. Yang, K. Chen, Z. Miao, and X. Yang. 2014. Cost-effective user monitoring for popularity prediction of online user-generated content. In *IEEE ICDM*. 944–951.
[91] Shuang-Hong Yang and Hongyuan Zha. 2013. Mixture of mutually exciting processes for viral diffusion. In *ICML*. vol. 28, 1–9.
[92] Zi Yang, Jingyi Guo, Keke Cai, Jie Tang, Juanzi Li, Li Zhang, and Zhong Su. 2010. Understanding retweeting behaviors in social networks. In *CIKM*. 1633–1636.
[93] Hongzhi Yin, Bin Cui, Hua Lu, Yuxin Huang, and Junjie Yao. 2013. A unified model for stable and temporal topic detection from social media data. In *ICDE*. 661–672.
[94] Linyun Yu, Peng Cui, Fei Wang, Chaoming Song, and Shiqiang Yang. 2015. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In *ICDM*. 559–568.
[95] Nicholas Jing Yuan, Yuan Zhong, Fuzheng Zhang, Xing Xie, Chin-Yew Lin, and Yong Rui. 2016. Who will reply to/retweet this tweet? The dynamics of intimacy from online social interactions. In *WSDM*. 3–12.
[96] Tauhid Zaman, Emily Fox, Eric Bradlow, et al. 2014. A Bayesian approach for predicting the popularity of tweets. *Annals of Applied Statistics* 8, 3 (2014), 1583–1611.
[97] Tauhid Zaman, Ralf Herbrich, Jurgen Gael, and David Stern. 2010. Predicting information spreading in Twitter. In *NIPS*. 599–601.
[98] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. 2013. Social influence locality for modeling retweeting behaviors. In *IJCAI*. 2761–2767.
[99] Jing Zhang, Jie Tang, Juanzi Li, Yang Liu, and Chunxiao Xing. 2015. Who influenced you? Predicting retweet via social influence locality. *ACM Transactions on Knowledge Discovery From Data* 9, 3 (2015), 25:1–25:26.
[100] Qi Zhang, Yeyun Gong, Jindou Wu, Haoran Huang, and Xuanjing Huang. 2016. Retweet prediction with attention-based deep neural network. In *CIKM*. 75–84.
[101] Xiaoming Zhang, Zhoujun Li, Wenhan Chao, and Jiali Xia. 2014. Popularity prediction of burst event in microblogging. In *WAIM*. 484–487.
[102] Yuan Zhang, Tianshu Lyu, and Yan Zhang. 2017. Hierarchical community-level information diffusion modeling in social networks. In *SIGIR*. 753–762.
[103] H. Zhao, G. Liu, C. Shi, and B. Wu. 2014. A retweet number prediction model based on followers' retweet intention and influence. In *ICDM*. 952–959.
[104] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A self-exciting point process model for predicting tweet popularity. In *SIGKDD*. 1513–1522.

Received March 2018; revised October 2018; accepted November 2018