Taxonomy and Testing for Microblog Popularity Prediction

ABSTRACT

As social networks have become a major source of information, predicting the outcome of information diffusion appears intriguing to both researchers and practitioners. In this paper we organize the joint effort of numerous studies into a hierarchical taxonomy. Specifically, we uncover three lines of thoughts: the *feature based approach*, *time series based approach* and the *user based approach*. 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 and robustness.

By establishing a taxonomy and testing scheme, 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 approaches have the advantage of achieving high accuracy and even further improving their accuracy when more of the cascade is seen. The feature based methods using only temporal features performs nearly as well as using all possible features, both having moderate performance. This suggests that temporal features do have strong predictive power and that power is better exploited with time series models. As social networks contain heterogeneous data, we encourage future researchers to devise new ways to incorporate various types of data.

CCS Concepts

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

Keywords

social network; popularity prediction; evaluation; taxonomy

1. INTRODUCTION

SIGMOD '17 Raleigh, North Carolina USA

© 2016 ACM. ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

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, 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 paper, 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, an 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 [11]. Other variations of the popularity prediction problem are either microscopic by predicting whether a specific user will adopt an item [64], [41] or continuous in time by predicting the size of the cascade along the time domain [60]. 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 2-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 have 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 paper we set out to classify

the purposed 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 3 general categories: *feature based methods*, *time series methods* and *user based 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. Time series methods borrow ideas from financial modelling and epidemiology, using only 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 epidemic models. User-based methods rely on the user-item repost matrix. Most of the methods follow the idea of collaborative filtering, inferring an user's repost actions from his/her history and the actions of similar users. Another line of thought is to treat users as sensors for popularity prediction and use their responses as features.

We also 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 can not 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 explicitly modelled in the independent cascade model and implicitly 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 8 feature based methods, 4 time series methods and 2 user based methods, 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. User-based methods have the tendency to underestimate the size of cascades, but their performance improve as a larger proportion of the cascade is observed.

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

- We construct a taxonomy of proposed methods for popularity prediction, categorizing them into feature based, time series based and user based methods. In the process we uncover many correlations between these methods.
- We analyze 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 square 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 paper is organized as follows. We formally describe the prediction problem in Section 2. In Section 3, Section 4 and Section 5 we introduce feature based methods, time series based methods and user based methods respectively. In Section 6, we categorize the mentioned methods by the rationale of prediction into motivation-oriented methods or monitor-based methods. Section 7, Section 8 and Section 9 shows 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.

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, 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.

Table 1: Notation list

Symbol	Description
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

We show our taxonomy of prediction methods in Figure 1. We primarily divide prediction methods into feature based methods, time series based methods and user based methods. For feature based methods, as the focus is mainly on

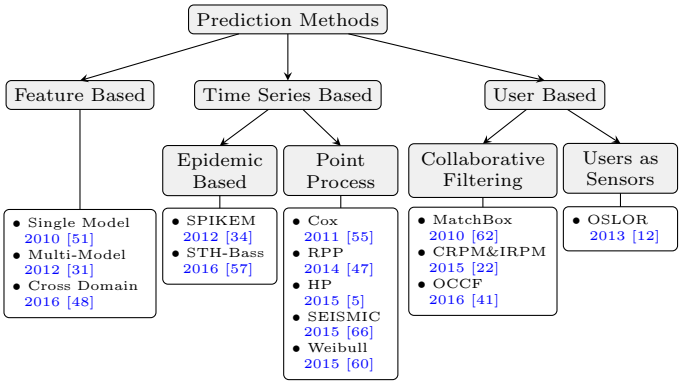

Figure 1: Category of prediction methods [1]

extracting features, the number of machine learning methods used for the task grew over time and recently there have been attempts to perform cross domain prediction. Time series based methods can be further divided into epidemic-based methods and point process methods based on whether the model is stochastic or deterministic. From the chronological tree, it is easy to see that point process model have received quite a lot amount of attention recently. User based methods can be categorized as collaborative filtering or using users as sensors. The collaborative filtering methods attempt to predict popularity on a microscopic level, from the prospective of predicting user actions. Using users as sensors is an unique idea proposed in the OSLOR model in which a selected set of users plays a role similar to features.

We list the related literature on popularity prediction in Table 2. In this table we show the type of prediction method used, how the prediction problem was formulated and which social network the dataset was from.

Table 2: Literature on popularity prediction

Article	Published in	Year	Type	Classification/ Regression	Dataset
[62]	NIPS	2010	User Based	Classification	Twitter
[51]	SocialCom	2010	Feature Based	Regression	Twitter
[58]	ICWSM	2010	Feature Based	Regression	Twitter
[17]	WWW	2011	Feature Based	Classification	Twitter
[38]	WebSci	2011	Feature Based	Classification	Twitter
[42]	AAAI	2011	Feature Based	Classification	Twitter
[55]	ICML	2011	time series Based	Regression	Citation
[34]	KDD	2012	Time Series Based	Regression	Twitter
[25]	CIKM	2012	Feature Based	Classification	Twitter
[31]	SIGIR	2012	Feature Based	Classification	Twitter
[54]	WSDM	2012	Feature Based	Regression	Twitter
[10]	SIGIR	2013	Feature Based	Classification	Twitter
[12]	SIGKDD	2013	User Based	Classification	Weibo
[32]	JASIST	2013	Feature Based	Classification	Twitter
[4]	WWW	2013	Feature Based	Regression	Weibo
[63]	IJCAI	2013	Feature Based	Classification	Weibo
[20]	WWW	2013	Feature Based	Classification	Twitter
[47]	AAAI	2014	Time Series Based	Regression	Citation
[11]	WWW	2014	Feature Based	Classification	Twitter
[23]	SIGIR	2014	Feature Based	Regression	Twitter
[13]	WWW	2014	Feature Based	Classification	Weibo
[1]	SBP-BRiMS	2015	Feature Based	Classification	Twitter
[5]	WWW	2015	Time Series Based	Regression	Weibo
[66]	SIGKDD	2015	Time Series Based	Regression	Twitter
[60]	ICDM	2015	Time Series Based	Regression	Weibo
[22]	CIKM	2015	User Based	Classification	Weibo
[30]	PAKDD	2015	Feature Based	Classification	Weibo
[24]	JASIST	2015	Feature Based	Classification	Twitter
[33]	ICWSM	2015	Feature Based	Classification	Twitter
[64]	TKDD	2015	Feature Based	Classification	Weibo
[26]	ICDMW	2015	Feature Based	Regression	Twitter
[57]	DASFAA	2016	Time Series Based	Regression	Twitter
[21]	SIGIR	2016	User Based	Classification	Weibo

[1]Only the first paper to use the model is cited in this figure.

3. FEATURE BASED METHODS

Many researchers have studied the factors affecting content popularity in social networks. One of the pioneers is Suh et al. [51]. Characterizing a microblog with content features and user features, they employed PCA and a Generalized Linear Model (GLM) 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.

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 [51]. More complicated features, such as sentiment and topic [28, 38, 11], are extracted using natural language processing techniques.

Naveed et al. [38] were one of the first to use content sentiment for popularity prediction. In 2013, Jenders et al. [20] brought up some other new sentiment features such as emotional divergence, and analyzed their relationship with microblog popularity. Some other researchers have also analyzed the association between the sentiment and the popularity of an item [53]. Most of them found that the negative sentiment has a higher correlation with item popularity.

f Some works showed that content features are effective in the popularity prediction task. Jenders et al. [20] reported that the content features, especially the sentiment features, have predictive power, though they are less effective than the structural features. [42], [51] and [11] 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 auxilliary in popularity prediction.

We use the following features in our implementation: individual sentiment value[20], emotional divergence [43], tweet length [42], number of hashtags [51], number of mentions and total number of special signals [30].

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 user's historical activity statistics, such as the number of reposts in the four weeks before the time of prediction t.

Petrovic et al. [42] pointed out the features of an author appear to be more important than features of the item itself. Multiple studies [3, 32] confirmed that the number of followers of the original author is an important predictor of popularity. Tsur et al. [54] also pointed out the feature —the fraction of past items that received reposts increases the accuracy of predictions. However, few studies report user based feature are 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 [58], 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 [51], mean number of friends or followers of the reposter [54], time the root user has been registered [48] and mean time the reposters have been registered.

Table 3: Effectiveness of the features

Paper	Year	Content-related	User-related	Structural	Temporal
[62]	2010	*	**	-	-
[51]	2010	*	**	-	-
[52]	2010	-	-	*	**
[17]	2011	*	**	*	*
[38]	2011	*	-	-	-
[42]	2011	*	**	-	-
[25]	2012	*	*	**	**
[31]	2012	*	-	**	-
[54]	2012	*	-	**	**
[32]	2013	*	-	**	-
[4]	2013	*	-	**	-
[20]	2013	*	**	-	-
[23]	2013	*	**	**	***
[13]	2014	-	-	*	**
[11]	2014	*	*	**	**
[24]	2015	*	*	*	**
[33]	2015	*	**	-	**
[60]	2015	-	-	*	**
[48]	2016	*	*	*	**

Structural Structural features are extracted from the user's friendship network, G_f. The term was coined in 2013 [4] though PageRank [40] values were included for prediction in previous works [25]. Gao et al. [13] used features extracted from network conducted with historical mention relationships.

When it comes to structural features, the PageRank algorithm is frequently used for quantifying the authorithy or influence of an user [13, 32, 56]. In Gao et al.'s experiment [13], maximum authority of authors in G_r appeared to be the best single feature in prediction.

Some research showed that using structural features improves prediction accuracy [44, 11]. When compared with content features and user related features, structural features are more powerful in prediction. However, works that include both structural features and temporal features suggest that the structural features are not so effective as temporal features [13, 11]. 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 [13], 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, link density [65], clustering coefficient of network [17], authority of the author of the original post [32], number of connected component in the network [44], number of edges from early adopters to the entire graph, indegree of the ith reposter ($2 < i < k$) [3], number of nodes reachable in two steps from the early reposter [48] and indegree of the ith adopter on the subgraph [27].

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 and coefficients of the fitted polynomial curve [25].

Many experimental results, such as [23], [48], [13], showed that temporal features are the most effective type of features. Szabo et al. [52] also showed that temporal features alone can reliably predict future popularity. Shulman et al. [48] conducted a cross-domain prediction experiment and the results showed temporal features have the ability to generalize to other social network. Hong et al. [17] 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 ($2 < i < k$) [17], mean time between reposts for the first half (rounded down) of the reposts [48], mean time between reposts for the last half (rounded up) of the reposts, coefficients of polynomial curve fitting and symbolic sequences [23], 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 [13].

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.

As for the machine learning models, Ma et al. [31] were the first to try out different 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 be the best under different circumstances. This is somehow reasonable, since the performance of machine learning models relies heavily on the features selected and the evaluation criteria. We list some frequently used machine learning models in Table 4.

Table 4: Machine learning methods

Notation	Description
GLM	Generalized linear model
NB	Naive Bayes model
DT	Decision Trees
LR	Logistic Regression
NN	Neural Network
RF	Random Forest
RT	Random Tree
SVM	Support Vector Machine

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.

Counting Processes/Point Processes. Every cascade can be broken up into single reposts that happen one by one, which can be described by a counting process. Formally, a counting 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 $T = \{t_1, t_2, \ldots\}$. This is known as a point process, but the two terms *counting process* and *point process* are interchangeable in our setting.

Point processes are characterized by a conditional intensity function $\lambda(t)$. Intuitively, the conditional intensity func-

Table 5: Comparison of time series based methods

Name	Function	Interpretation	Excitation	Time Decay	Bound
PP	$\lambda(t) = \lambda$	λ: constant intensity rate	-	-	No
RPP[47]	$\lambda(t) = \lambda_i f(t;\theta) a_i(t)$	λ_i: item attractiveness; $f(\cdot)$: time decay function; $a_i(\cdot)$: amount of attention received	Accumulated	Log-normal	No
HP [5]	$\lambda(t) = \lambda + \int_0^t \mu(t-u) dN(u)$	λ: background intensity; $\mu(\cdot)$: excitation function	Single	Exponential	No
SEISMIC [66]	$\lambda(t) = p_t + \Sigma_{t_u<t} n_i \phi(t-t_u)$	n_i: node degree; $\phi(t)$: human reaction time; p_t: post infectiousness	Single	Power-Law	No
Weibull [60]	$\lambda_i(t) = \frac{k_i}{\lambda_i}(\frac{t}{\lambda_i})^{k_i-1}$	λ_i: scale; k_i: shape	-	Exponential	Yes
Cox [55]	$\lambda_i(t) = \alpha_0(t) e^{(\beta^T s_i(t))}$	$\alpha_0(\cdot)$: baseline hazard function; $\mathbf{s_i}$: vector of features; β: vector of coefficients	Accumulated	Flexible	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	-	Yes
SPIKEM [34]	$\Delta R(n+1) = U(n)\Sigma_{t=n_0}^{n}(\Delta R(t)+ S(t))f(n+1-t)+\epsilon$	$R(t)$: 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 [57]	$\frac{dH(t)}{dt} = (p + qH(t))(1 - H(t))$	$H(t)$: fraction of adoption; p: external influence; q: internal influence	Accumulated	-	Yes

tion refers to the expected rate of arrivals given history $\mathcal{H}$. This is also called the hazard function in survival analysis literature [8].

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

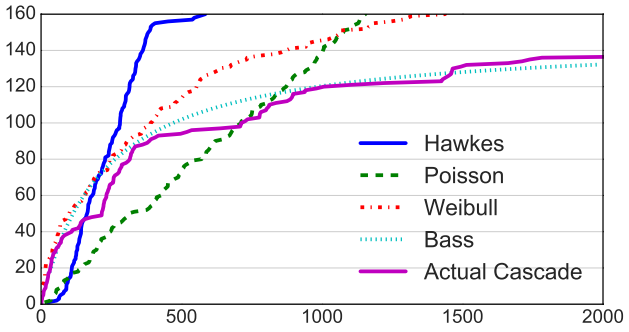

Figure 2: Fitting time series models to an actual cascade

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.

The Reinforced Poisson Process (RPP) [47] is an example of inhomogeneous Poisson Process which takes the current state into account. Specifically, the amount of attention received $a_i(\cdot)$ is defined as a step 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 [14], 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 an geometric sequence.

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 [34], [37]. 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. The total excitation is the sum of single excitations.

Note that if the excitation function $\mu(\cdot) = 0$, the Hawkes Process degrades into the homogeneous Poisson Process. The Hawkes Process is flexible in the sense that the excitation function can be tailored for the specific problem. In [5], Bao et al. proposed to use the exponential excitation function to model a single tweet cascade as a Hawkes Process. Different parameters are used for the initial post and further reposts, implying that they are of different importance to the final popularity.

On the basis of previous work, [66] extended the Hawkes model to a doubly stochastic self-exciting point process SEISMIC. The post infectiousness p_t measures of the probability of it being reposted at time t. When the post infectiousness is constant, the model is the standard Hawkes Process. p_t is not modelled in a parametric form, but estimated from the observations. Experimental analysis on Twitter data shows that the reaction time remains constant for a short period and then follows a power-law decay. Other works show that the tail follows a log-normal distribution [61]. The exact distribution may vary between different social networks and thus should be confirmed against real data.

Yu et al. adopted the Weibull model for the intensity function in [60]. Both parameters k and λ are estimated using the maximum likelihood. The homogeneous Poisson Process is a special case of the Weibull model when $k_i = 1$.

Vu et al. proposed to use the proportional hazards model (Cox model) in [55]. This model combines features with point processes. A similar idea was proposed in [15], however, the latter focuses on the task of network inference.

Epidemic Models Another family of models is derived from epidemic models. These models have some common characteristics: being defined on a finite population and labeling the population according to stages of disease. Epidemic models are inherently self-excited in the initial phrase since the susceptible is transformed into the infected which then attempt to infect others. However, due to the limits on the population, epidemic 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 2 states: susceptible (S) and infected (I), hence the name 'Susceptible-Infected' SI model. Every infected node attempts to infect each of its neighbors independently with a probability β. Once infected, the node will stay infected forever.

Matsubara et al. pointed out in [34] that the classical SI model does not capture the patterns of real-life information cascades. The solution for $I(t)$ is sigmoid, the rise and fall of the derivative are both exponential while 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 2 states: the unposted or uninformed state (U) and the reposted state (R). Compared to the Hawkes Process, the SPIKEM model allows multiple nodes to be activated (multiple arrivals in point processes) at a single time point. 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. The noise term ϵ is used to model the background rate.

Observing that posting behavior 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 [6] 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. extended the Bass model for tweet prediction by incorporating user features x and content features y [57]. This is yet another example of combining features with models. In Table 5 we show the core function of the above model and provide an explanation of the symbols. 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.

5. USER BASED METHODS

User based prediction methods use the interaction history of the users as the basis for prediction. The important distinction between user-based methods and user features mentioned in Section 3 is that user features are attributes of users, while in user based methods we focus on whether an user has participated in a cascade or not.

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

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

We observe that user based methods can be further divided into collaborative filtering methods and methods that exploit users as sensors.

Collaborative Filtering 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.

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; the second is that it is hard to distinguish missing data with negative data. An 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.

In an attempt to tackle the first problem, Zaman et al. adopted the Matchbox prediction model [50] in [62] to incorporate user features and content features. Jiang et al. introduced item clustering as a regularization factor to utilize item features [22]. This ensured that items similar in features would be also be close in latent space.

Pan et al. [41] 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.

Users as Sensors Many popular items are related with influential users—these users are experts at discovering content and have a large group of faithful fans. In a sense, these users are sensors of potential popular cascades.

For every item, Cui et al. [12] aims 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. The former is achieved by adding an L1 regularization term T_3 and the latter by adding an orthogonality regularization term T_2.

X_i is the current cascade vector, which is a column from the user-item repost matrix. The Orthogonal Sparse Logistic Regression (OSLOR) model is formulated as follows:

$$F(\theta) = T_1(\theta) + T_2(\theta) + T_3(\theta)$$

$$T_1(\theta) = -\log L(\theta) = -\Sigma_{i=1}^{m}(\log(1 + e^{X_i\theta}) + y^T X\theta$$

$$T_2(\theta) = \frac{\beta}{4}\Sigma_{i,j}(\theta_i X_i^T X_j \theta_j)^2$$

$$T_3(\theta) = \gamma\|\theta\|_1$$

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 6.

Table 6: Comparison of rationale

Name	Motivation-oriented		Monitor-based
	Homophily	Influence	
Feature based	✓	✓	✓
Time Series based	×	×	✓
Collaborative Filtering	✓	×	✓
OSLOR	×	✓	✓

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.

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 [46]. 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 [36]. 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 behavior.

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 analyzing 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. proposed to classify microblogs by transfer learning [7]. 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. [41] 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 or the fraction of tweets that have received reposts. Structural features like PageRank scores also measure the authority of an user. [63] went even further to devise novel features such as pairwise influence from random walk with restart and structural influence from the number of connected components in active neighbors.

When using users as sensors, [12] 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.

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 [48], 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.

7. EXPERIMENT SETUP

7.1 Soundness of Data

Here we list the criteria that datasets used for evaluation should meet.

The set of users should be large and have complete user profiles and relationships, that is to say, all friends of a core user set should be included in the dataset. Users should have a substantial amount of historical activities in order to infer the activity preferences of the user. The diffusion history of the microblog in question should be complete, with all timestamps and users involved. The content of the microblog should be recorded, including URLs, mentions and other special symbols. Of course, the final repost of the microblog must be included in the dataset to serve as the ground truth.

In our evaluation we have two assumptions: all users are active and the relationships do not change over the time of the diffusion. The former can be assured by ruling out users that have no recent activity and the latter can be resolved by using periodical snapshots of the network.

7.2 Data Description

Two different datasets were used for evaluation, one from Twitter[2] and the other from its Chinese equivalent Weibo[3]. 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.

7.2.1 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.

7.2.2 Weibo

The Weibo dataset is from Zhang et al. [63] and is publicly

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

available online[4]. 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 7.

Table 7: 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

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 3. 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 3 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.

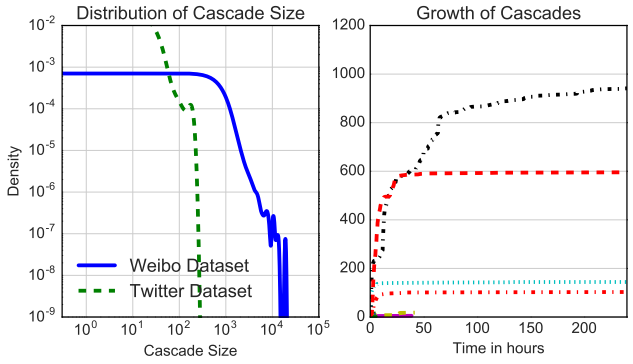

Figure 3: Distribution of cascades and their growth

7.3 Prediction Methods & Data Input

We list the prediction methods used in the experimental evaluation and compare the types of data needed for each method in Table 8.

A total of 8 feature based methods were actually implemented, 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, 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. The results of the other methods can be found in the Appendix. Four time series based model 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 user based 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.

As seen in the table, 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 number of followers of the user. This provides them with additional information about the context of the cascade. User based 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.

Table 8: Need for data of prediction methods

Method	User Activity History	User Profile	Friend Network	Item Content	Repost Time	Repost Count
Feature based	○	○	○	○	○	○
RPP	×	×	×	×	✓	✓
Weibull model	×	×	×	×	✓	✓
SEISMIC	×	✓	×	×	✓	✓
STH-Bass	×	✓	×	✓	×	✓
OCCF	✓	×	✓	✓	×	✓
OSLOR	✓	×	×	×	×	✓.

7.4 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 user based 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.

In our comparison study, we define the threshold for classification as the minimum number of reposts needed to reach the top 1% precentile. Methods that are proposed for the regression can also be applied to the classification problem given this threshold.

8. EVALUATION RESULTS

[4]arnetminer.org/Influencelocality

Table 9: Accuracy of regression on 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 10: Accuracy of regression on 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

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.

$$SE(t) = (\hat{R}_i - R_i)^2$$

$$APE(t) = \frac{|\hat{R}_i - R_i|}{R_i}$$

The median SE-time curves and median APE-time curves of different methods are shown in Figure 4 and Figure 5.

For the classification task, we rank the cascades by the number of reposts 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.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

TP, FP and FN are number of true positive instances, number of false positive instances and number of false negatives instances respectively.

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.

For the classification task, feature-based methods have a

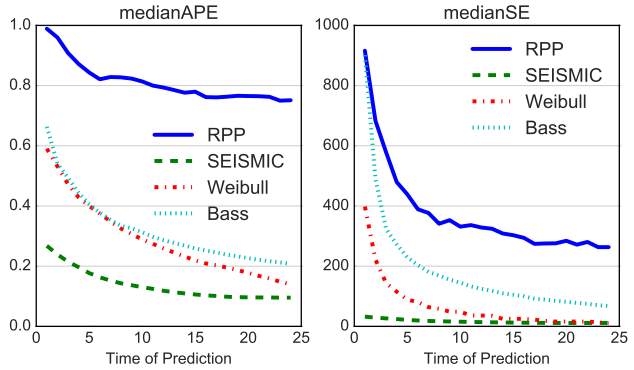

Figure 4: Regression metrics on Weibo dataset

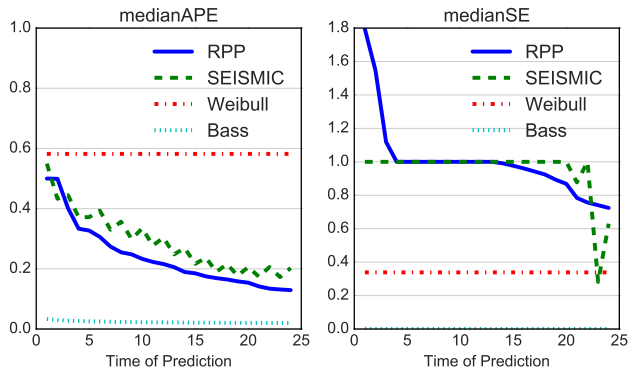

Figure 5: Regression metrics on Twitter dataset

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. 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

Table 11: Accuracy of classification on Weibo dataset

Time	6h			12h			18h			24h		
Methods	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
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

Table 12: Accuracy of classification on Twitter dataset

Time	6h			12h			18h			24h		
Methods	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
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

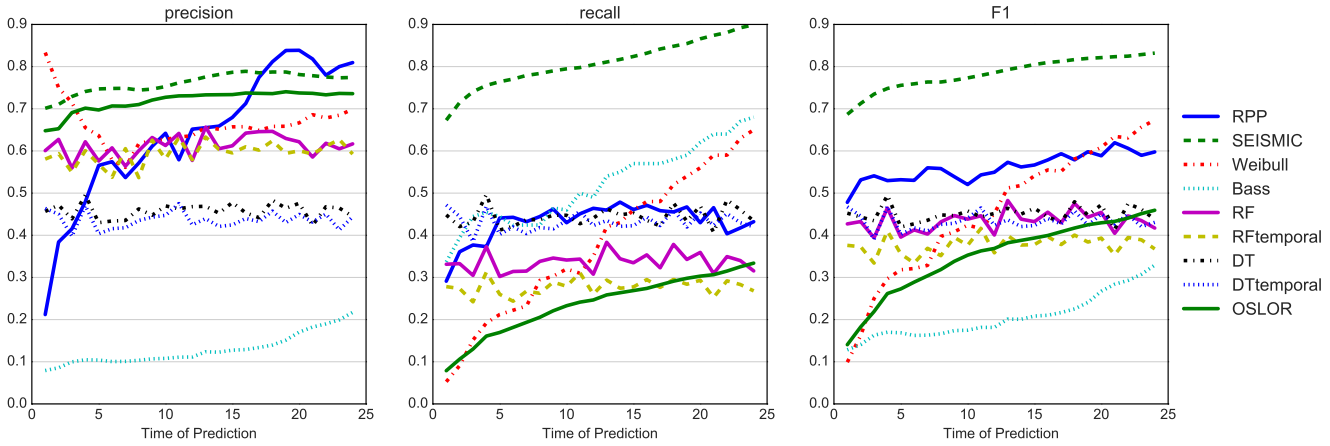

Figure 6: Classification metrics on Weibo dataset

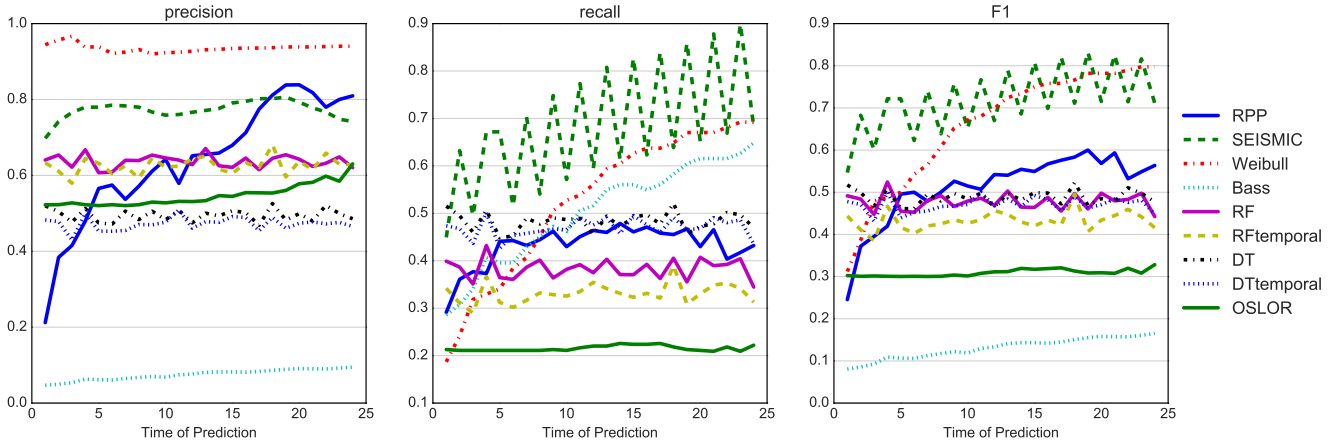

Figure 7: Classification metrics on twitter dataset

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, 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.

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 & 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 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 8, 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 9, 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 & DISCUSSION

9.1 Efficiency

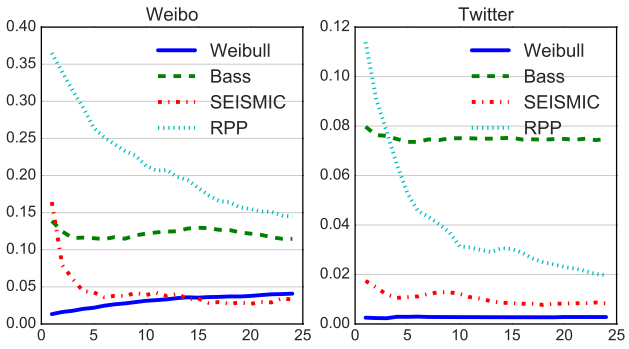

Figure 8: APE>200%

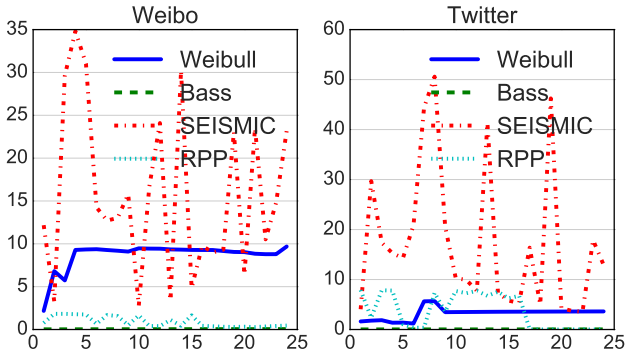

Figure 9: Kurtosis/ 10^3

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 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 13. For OCCF, all stages are performed batch-wise. It is both high in time complexity and space complexity and requires pretrained content topics as input. The feature-based method that use all features is time-consuming due to network features and content features extraction. Both feature-based methods and OSLOR method 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 Figure 10 and Figure 11, we discover that some of the prediction methods display bias. Specifically, the Weibull model produces pessimisic predic-

Table 13: 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
User based	low	batch	single

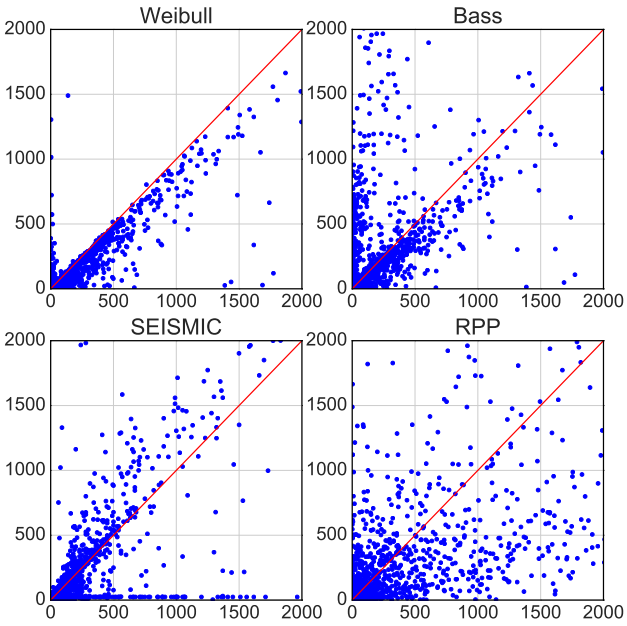

Figure 10: Comparing the predicted size to actual cascade size on Weibo dataset

tions. The Bass model, on the other hand, tends to overestimate 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 it low space efficiency, especially the popularity prediction in large community since it require 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.

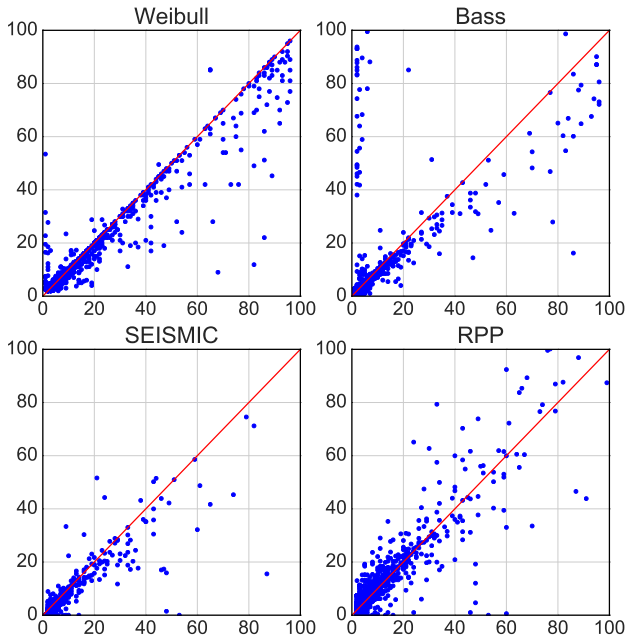

Figure 11: Comparing the predicted size to actual cascade size on Twitter dataset

10. CONCLUSION

In this paper, 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 user based—and analyze 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 a 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.

11. REFERENCES

[1] S. Alzahrani, S. Alashri, A. Koppela, H. Davulcu, and I. Toroslu. A network-based model for predicting hashtag breakouts in twitter. In *SBP*, pages 3–12, 2015.

[2] S. Asur and B. A. Huberman. Predicting the future with social media. In *Web Intelligence*, pages 492–499, 2010.

[3] E. Bakshy, J. Hofman, W. Mason, and D. Watts. Everyone's an influencer: Quantifying influence on twitter. In *ACM WSDM*, pages 65–74, 2011.

[4] P. Bao, H. Shen, J. Huang, and X. Cheng. Popularity prediction in microblogging network: a case study on sina weibo. In *ACM WWW*, pages 177–178, 2013.

[5] P. Bao, H. Shen, X. Jin, and X. Cheng. Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In *ACM WWW*, pages 9–10, 2015.

[6] F. M. Bass. A new product growth for model consumer durables. *Management science*, 15(5):215–227, 1969.

[7] J. Bian, Y. Yang, and T.-S. Chua. Predicting trending messages and diffusion participants in microblogging network. In *ACM SIGIR*, pages 537–546, 2014.

[8] G. E. Box and D. R. Cox. An analysis of transformations. *JRSS*, 26(2):211–252, 1964.

[9] W. Chanthaweethip, X. Han, N. Crespi, Y. Chen, R. Farahbakhsh, and Á. Cuevas. "current city" prediction for coarse location based applications on facebook. In *GLOBECOM*, pages 3188–3193, 2013.

[10] Y. Chen, H. Amiri, Z. Li, and T.-S. Chua. Emerging topic detection for organizations from microblogs. In *ACM SIGIR*, pages 43–52, 2013.

[11] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *ACM WWW*, pages 925–936, 2014.

[12] P. Cui, S. Jin, L. Yu, F. Wang, W. Zhu, and S. Yang. Cascading outbreak prediction in networks: a data-driven approach. In *ACM SIGKDD*, pages 901–909, 2013.

[13] S. Gao, J. Ma, and Z. Chen. Effective and effortless features for popularity prediction in microblogging network. In *ACM WWW*, pages 269–270, 2014.

[14] S. Gao, J. Ma, and Z. Chen. Modeling and predicting retweeting dynamics on microblogging platforms. In *ACM WSDM*, pages 107–116, 2015.

[15] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *ICML*, pages 666–674, 2013.

[16] J. He, W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence. Improving traffic prediction with tweet semantics. In *IJCAI*, pages 1387–1393, 2013.

[17] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *ACM WWW*, pages 57–58, 2011.

[18] Z. Hu, J. Yao, B. Cui, and E. P. Xing. Community level diffusion extraction. In *SIGMOD*, pages 1555–1569, 2015.

[19] Y. Huang, S. Zhou, K. Huang, and J. Guan. Boosting financial trend prediction with twitter mood based on selective hidden markov models. In *DASFAA*, pages 435–451, 2015.

[20] M. Jenders, G. Kasneci, and F. Naumann. Analyzing and predicting viral tweets. In *ACM WWW*, pages 657–664, 2013.

[21] B. Jiang, J. Liang, Y. Sha, R. Li, W. Liu, H. Ma, and L. Wang. Retweeting behavior prediction based on one-class collaborative filtering in social networks. In *ACM SIGIR*, pages 977–980, 2016.

[22] B. Jiang, J. Liang, Y. Sha, and L. Wang. Message clustering based matrix factorization model for retweeting behavior prediction. In *ACM CIKM*, pages 1843–1846, 2015.

[23] S. Kong, Q. Mei, L. Feng, F. Ye, and Z. Zhao. Predicting bursts and popularity of hashtags in real-time. In *ACM SIGIR*, pages 927–930, 2014.

[24] S. Kong, F. Ye, L. Feng, and Z. Zhao. Towards the prediction problems of bursting hashtags on twitter. *JASIST*, 66(12):2566–2579, 2015.

[25] A. Kupavskii, L. Ostroumova, A. Umnov, S. Usachev, P. Serdyukov, G. Gusev, and A. Kustarev. Prediction of retweet cascade size over time. In *ACM CIKM*, pages 2335–2338, 2012.

[26] R. Lemahieu, S. Van Canneyt, C. De Boom, and B. Dhoedt. Optimizing the popularity of twitter messages through user categories. In *IEEE ICDMW*, pages 1396–1401, 2015.

[27] K. Lerman and A. Galstyan. Analysis of social voting patterns on digg. In *ACM WOSN*, pages 7–12, 2008.

[28] C. Li, A. Sun, and A. Datta. Twevent: Segment-based event detection from tweets. In *ACM CIKM*, pages 155–164, 2012.

[29] W. Liu, Z.-H. Deng, X. Gong, F. Jiang, and I. W. Tsang. Effectively predicting whether and when a topic will become prevalent in a social network. In *AAAI*, pages 210–216, 2015.

[30] Z. Luo, Y. Wang, X. Wu, W. Cai, and T. Chen. On burst detection and prediction in retweeting sequence. In *PAKDD*, pages 96–107, 2015.

[31] Z. Ma, A. Sun, and G. Cong. Will this# hashtag be popular tomorrow? In *ACM SIGIR*, pages 1173–1174, 2012.

[32] Z. Ma, A. Sun, and G. Cong. On predicting the popularity of newly emerging hashtags in twitter. *JASIST*, 64(7):1399–1410, 2013.

[33] S. Maity, A. Gupta, P. Goyal, and A. Mukherjee. A stratified learning approach for predicting the popularity of twitter idioms. In *ICWSM*, pages 642–645, 2015.

[34] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: Model and implications. In *ACM SIGKDD*, pages 6–14, 2012.

[35] J. McGee, J. Caverlee, and Z. Cheng. Location prediction in social media based on tie strength. In *CIKM*, pages 459–468, 2013.

[36] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[37] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita. Self-exciting point process modeling of crime. *JASA*, 106(493):100–108, 2012.

[38] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi.

Bad news travel fast: A content-based analysis of interestingness on twitter. In *ACM WebSci*, pages 1–7, 2011.

[39] T. H. Nguyen and K. Shirai. Topic modeling based sentiment analysis on social media for stock market prediction. In *ACL*, pages 1354–1364, 2015.

[40] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

[41] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *IEEE ICDM*, pages 502–511, 2008.

[42] S. Petrovic, M. Osborne, and V. Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, pages 586–589, 2011.

[43] R. Pfitzner, A. Garas, and F. Schweitzer. Emotional divergence influences information spreading in twitter. In *ICWSM*, pages 2–5, 2012.

[44] D. M. Romero, C. Tan, and J. Ugander. On the interplay between social and topical structure. *arXiv preprint arXiv:1112.1115*, 2011.

[45] A. Sadilek, H. A. Kautz, and V. Silenzio. Predicting disease transmission from geo-tagged micro-blog data. In *AAAI*, pages 136–142, 2012.

[46] A. Sharma and D. Cosley. Distinguishing between personal preferences and social influence in online activity feeds. In *ACM CSCW*, pages 1089–1101, 2016.

[47] H.-W. Shen, D. Wang, C. Song, and A.-L. Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. In *AAAI*, pages 291–297, 2014.

[48] B. Shulman, A. Sharma, and D. Cosley. Predictability of popularity: Gaps between prediction and understanding. In *ICWSM*, pages 348–357, 2016.

[49] X. Song, L. Nie, L. Zhang, M. Akbari, and T.-S. Chua. Multiple social network learning and its application in volunteerism tendency prediction. In *SIGIR*, pages 213–222, 2015.

[50] D. H. Stern, R. Herbrich, and T. Graepel. Matchbox: Large scale online bayesian recommendations. In *ACM WWW*, pages 111–120, 2009.

[51] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *IEEE SOCIALCOM*, pages 177–184, 2010.

[52] G. Szabó and B. A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53(8):80–88, 2010.

[53] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment in twitter events. *JASIST*, 62(2):406–418, 2011.

[54] O. Tsur and A. Rappoport. What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities. In *ACM WSDM*, pages 643–652, 2012.

[55] D. Q. Vu, A. U. Asuncion, D. R. Hunter, and P. Smyth. Dynamic egocentric models for citation networks. In *ICML*, pages 857–864, 2011.

[56] S. Wang, Z. Yan, X. Hu, P. S. Yu, and Z. Li. Burst time prediction in cascades. In *AAAI*, pages 325–331, 2015.

[57] Y. Yan, Z. Tan, X. Gao, S. Tang, and G. Chen. Sth-bass: A spatial-temporal heterogeneous bass model to predict single-tweet popularity. In *DASFAA*, pages 18–32, 2016.

[58] J. Yang and S. Counts. Predicting the speed, scale, and range of information diffusion in twitter. In *ICWSM*, pages 355–358, 2010.

[59] H. Yin, B. Cui, H. Lu, Y. Huang, and J. Yao. A unified model for stable and temporal topic detection from social media data. In *IEEE ICDE*, pages 661–672, 2013.

[60] L. Yu, P. Cui, F. Wang, C. Song, and S. Yang. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In *IEEE ICDM*, pages 559–568, 2015.

[61] T. Zaman, E. Fox, E. Bradlow, et al. A bayesian approach for predicting the popularity of tweets. *AOAS*, 8(3):1583–1611, 2014.

[62] T. Zaman, R. Herbrich, J. Gael, and D. Stern. Predicting information spreading in twitter. In *NIPS*, pages 599–601, 2010.

[63] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, pages 2761–2767, 2013.

[64] J. Zhang, J. Tang, J. Li, Y. Liu, and C. Xing. Who influenced you? predicting retweet via social influence locality. *TKDD*, 9(3):25:1–25:26, 2015.

[65] X. Zhang, Z. Li, W. Chao, and J. Xia. Popularity prediction of burst event in microblogging. In *WAIM*, pages 484–487, 2014.

[66] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *ACM SIGKDD*, pages 1513–1522, 2015.

APPENDIX

A. PREDICTION IN SOCIAL NETWORKS

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. [29], [59] and [23] were originally designed to track the popularity of topics. Users can also be aggregated into communities. [18] 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 et al. used the rate of Twitter mentions to predict the box office revenue of movies [2]. 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. [19] 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. [39] 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

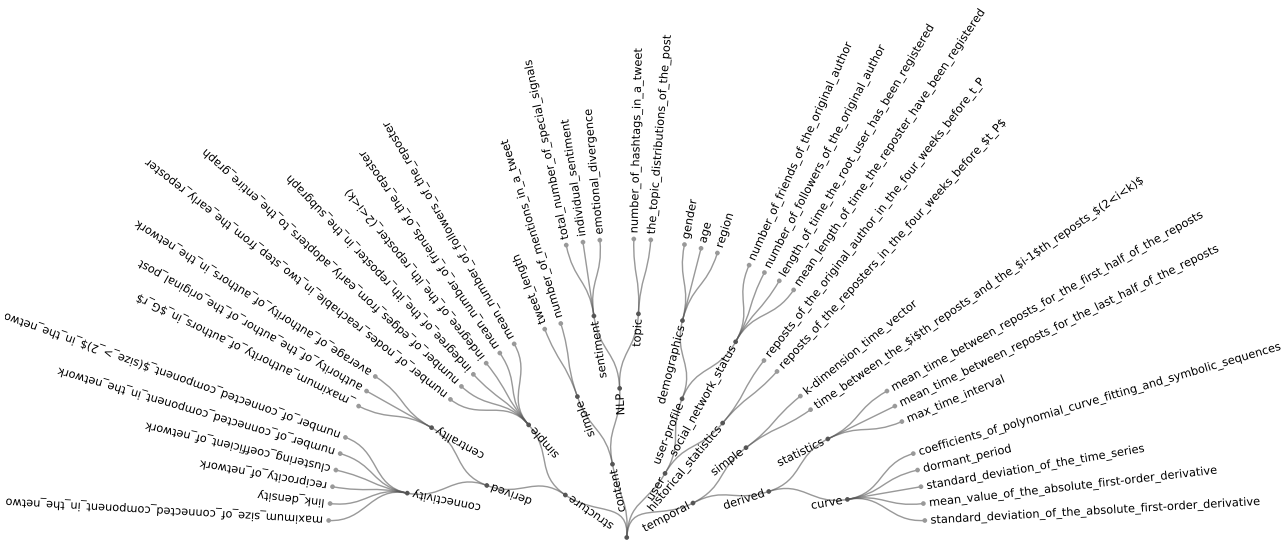

Figure 12: Hierarchy of features used for prediction

trend. Social network data has been applied to the prediction of user activities outside the network, such as volunteer tendency in [49].

Under the setting of location-based social networks (LBSN), user's locations can be predicted given the locations of his/her friends [35] or the user's profile[9]. 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 [16]. 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, [45] designed a conditional random field model to predict whether and when an 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.

B. HIERARCHY OF FEATURES

We further categorize the features used in feature-based methods. The hierarchy of the feature is shown in Figure 12.

For structural features, we divide them into derived features and simple features. Derived features are either measures of connectivity or centrality. The link density and the number of connected components are all metrics for connectivity. Intuitively, the more densely connected the network is, the highly probability messages are seen by other users. The authority scores obtained by PageRank is a measure of the centrality of the user or the influence of the user. The higher influence the user exerts, the more possible his/her post gets reposted by followers. Simple features are those that do not need extra processing and are attributes of the network, such as the in-degree of the user.

For content features, they can also be divided into derived features and simple features. Since the content is in essence text, derived features are obtained using natural language processing techniques. Two main types of derived features are sentiment and topic. The topic can either be naively defined as the hashtags attached to the microblog or extracted using topic models such as LDA. Sentiment features can be an individual sentiment or the divergence between positive and negative sentiments. As for simple features, they are often the count of total words or special characters.

For user features, we can generally categorize the features as historical statistics or the user profile. 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. Demographics reveal truths about the person behind the user id and captures homophily that exists in the real world. The network status on the other hand, may reflect the influence of the user.

For temporal features, we can either derive features by applying simple statistics or fitting curves to the time series. All temporal features try to capture the pattern in the growth of the cascade.

C. ACCURACY OF FEATURE BASED METHODS

We list the accuracy of the feature based methods in Table 14-17. From these tables we can see that in most cases, the F1 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 F1 score. As Decision Trees and Random Forest outperform the rest in terms of F1 score, we choose these two methods as representatives when comparing with other types of prediction methods.

Table 14: Accuracy of classification feature based on Weibo data (all features)

time	6h			12h			18h			24h		
method	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
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 15: Accuracy of classification feature based on Weibo data (temporal features)

time	6h			12h			18h			24h		
method	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
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

Table 16: Accuracy of classification feature based on Twitter data (all features)

time	6h			12h			18h			24h		
method	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
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 17: Accuracy of classification feature based on Twitter data (temporal features)

time	6			12			18			24		
method	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
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