EPAB: Early Pattern Aware Bayesian Model for Social Content Popularity Prediction

Qitian Wu[1], Chaoqi Yang[1], Xiaofeng Gao[1*], Peng He[2], Guihai Chen[1]
[1]Shanghai Jiao Tong University, Shanghai, China
[2]Tencent, Shenzhen, China
echo740@sjtu.edu.cn, ycqsjtu@gmail.com, gao-xf@cs.sjtu.edu.cn, paulhe@tencent.com, gchen@cs.sjtu.edu.cn

***Abstract*—The boom of information technology enables social platforms (like Twitter) to disseminate social content (like news) in an unprecedented rate, which makes early-stage prediction for social content popularity of great practical significance. However, most existing studies assume a long-term observation before prediction and suffer from limited precision for early-stage prediction due to insufficient observation. In this paper, we take a fresh perspective, and propose a novel early pattern aware Bayesian model. The early pattern representation, which stands for early time series normalized on future popularity, can address what we call early-stage indistinctiveness challenge. Then we use an expressive evolving function to fit the time series and estimate three interpretable coefficients characterizing temporal effect of observed series on future evolution. Furthermore, Bayesian network is leveraged to model the probabilistic relations among features, early indicators and early patterns. Experiments on three real-world social platforms (Twitter, Weibo and WeChat) show that under different evaluation metrics, our model outperforms other methods in early-stage prediction and possesses low sensitivity to observation time.**

I. Introduction

Nowadays, social content (like hashtags or news on Twitter) may trigger thousands even millions of posts or reposts, and further cause a huge social impact. Such phenomenon urges researchers to use early observed information to predict how many posts will arrive in the end, i.e., popularity of social content [1], [2]. If the prediction can be made at very early stage, it could bring great prophetic benefits in various domains, such as rumor monitoring [3], personalized recommendation [4] and targeted advertisement [5], etc.

There are extensive studies on content popularity prediction. In terms of methodology, they can be generally divided into two categories. The first category is feature driven method, which first extracts a set of observable features from early information (including observed time series, contextual information, user profiles or social network information) and then adopts machine learning algorithms to optimize a mapping function from features to popularity. Another category is point process method, which treats time interval between every post arrival as a random variable and utilizes stochastic process to model temporal sequence of post arrival. The sequence is used to estimate one intensity function reflecting some dynamic patterns hidden in temporal sequence. Based on the intensity function, one can further derive theoretical expectation of popularity or simulate future post arrival to conduct prediction. A slew of prior works based on these two methods have achieved decent accuracy when applied to a variety of real-world popularity prediction tasks for single posts [6], social topics [7], memes [8], hashtags [9], [10] and videos [11].

However, these studies are all based on a narrow perspective of long-term observation. For instance, [7] utilizes features extracted from dozens of hours observation to predict popularity of social topics; and even worse, [10] predicts popularity of Twitter hashtags when 80% related posts are exposed. Such 'late-arriving' prediction lacks timeliness and embodies little practical significance. For example, in rumor monitoring domain, monitors need to detect a rumor on social platform before it causes great influence, so the accurate prediction should be done soon after its emergence. Moreover, this long-term observation provides sufficient information to help decision making. Based on that, feature driven method can easily extract some effective observable features that strongly correlate with popularity, and point process method can recognize some typical patterns hidden in the observed temporal sequence and further confidently deduce the future evolution. Unfortunately, early-stage prediction only allows short-time observation (e.g., 2 hours), providing insufficient information with random noise. Under such circumstance, previous models would get stuck in a poor performance. Hence, building an accurate and interpretable model to predict content popularity at early stage can bring great practical benefits and is waiting to be solved.

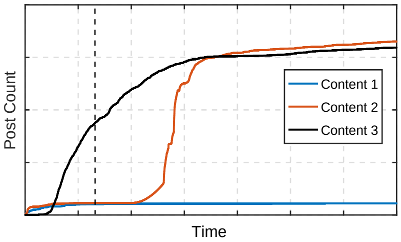

Fig. 1: Post count time series for three social contents. The dash line is observation time and its left-side time interval is early-stage observing interval.

Early-stage popularity prediction is a non-trivial task with

*Xiaofeng Gao is the corresponding author. This work is supported by the China 973 project (2014CB340303), the National Natural Science Foundation of China (61872238, 6147-2252, 61672353), the Shanghai Science and Technology Fund (1751-0740200), the CCF-Tencent Open Research Fund (RAGR20170114), and Huawei Innovation Research Program (HO2018085286).

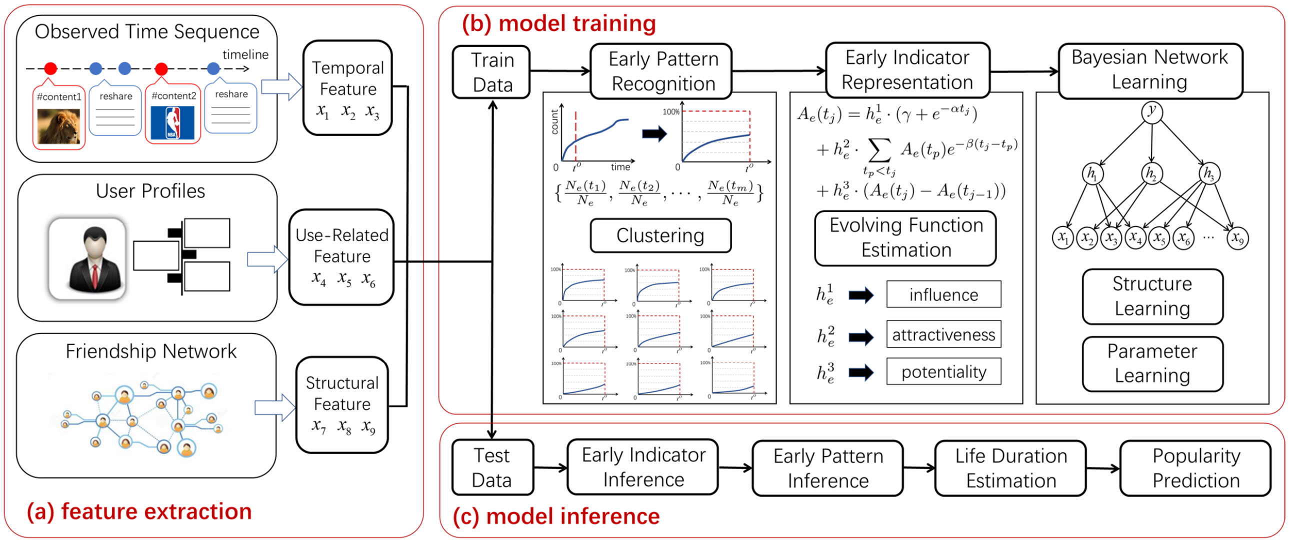
Observed Time Sequence
timeline
#content1
reshare
#content2
reshare
Temporal Feature x_1 x_2 x_3
User Profiles
Use-Related Feature x_4 x_5 x_6
Friendship Network
Structural Feature x_7 x_8 x_9
(a) feature extraction
(b) model training
Train Data
Early Pattern Recognition
Early Indicator Representation
Bayesian Network Learning
count
time
100%
$\{\frac{N_e(t_1)}{N_e}, \frac{N_e(t_2)}{N_e}, \cdots, \frac{N_e(t_m)}{N_e}\}$
Clustering
$A_e(t_j) = h_e^1 \cdot (\gamma + e^{-\alpha t_j})$
$+ h_e^2 \cdot \sum_{t_p<t_j} A_e(t_p) e^{-\beta(t_j - t_p)}$
$+ h_e^3 \cdot (A_e(t_j) - A_e(t_{j-1}))$
Evolving Function Estimation
h_e^1 influence
h_e^2 attractiveness
h_e^3 potentiality
y
h_1 h_2 h_3
x_1 x_2 x_3 x_4 x_5 x_6 ... x_9
Structure Learning
Parameter Learning
Test Data
Early Indicator Inference
Early Pattern Inference
Life Duration Estimation
Popularity Prediction
(c) model inference

Fig. 2: EPAB framework. (a) In feature extraction, we extract temporal, user-related and structural features from data sets. (b) In model training, the training data are used to conduct model learning. (c) In model inference, we utilize observed information in the test data to predict prediction based on the trained model.

two major challenges. First, some contents with similar early-stage evolution could generate quite different popularity (e.g., content 1 and content 2 in Fig. 1 behave similarly at first but have different future evolution). Such early-stage indistinctiveness phenomenon makes it hard to leverage observed time series to accurately deduce future evolution. Second, conversely, some contents with totally different evolution trends may reach a similar popularity (like content 2 and content 3 in Fig. 1). This indicates that observable features extracted from early time series could possess weak correlation with popularity, and some extra information from user profiles or social networks (spatial information of friendship networks) should be taken into account.

In this paper, we propose an early pattern aware Bayesian model (EPAB) to handle the early-stage popularity prediction task from a new perspective. The framework of our model is shown in Fig. 2. At the outset, we define a new concept, *early pattern*, to represent early-stage (observed) time series normalized on popularity. The normalization could overcome the challenge of early-stage indistinctiveness. Then, we adopt an expressive evolving function to fit the time series of contents and estimate three interpretable coefficients, called *influence*, *attractiveness* and *potentiality*, which characterize how early-stage series affect its future evolution. These coefficients named as *early indicators* bridge the gap between early patterns and observable features. Then we adopt Bayesian network to model the probabilistic relations among observable features, early indicators and early patterns. The Bayesian network aims at interpreting the observable information and latent relationship as much as possible, and select a set of effective features for early-stage prediction. Based on the trained Bayesian network, one can use the observable features to deduce the early indicators and early patterns, and further predict future popularity.

To verify our model, we conduct extensive experiments on three large data sets originated from different social platforms (Twitter, Weibo, and WeChat) and compete EPAB with six powerful baselines. Our prediction is made based on observation of 1 hour for Twitter ,WeChat and 2 hours for Weibo. Experiment results manifest that under different evaluation metrics, our model achieves significant performance improvement in early-stage popularity prediction and low sensitivity to observation time.

II. Early Pattern Aware Bayesian Model

In this section, we introduce our model EPAB (Early Pattern Aware Bayesian Model) in detail. In Section V.A$\sim$V.D, we present model training procedure including early pattern recognition, early indicator representation, early pattern and indicator modeling (using Bayesian network), and parameter learning. We will go into our motivations, specific implementations as well as model interpretation.

A. Early Patterns and Features

Firstly, we find that some contents with similar observed count series $\mathcal{S}_e^O$ but possess quite different popularity N_e. To address this challenge, we adopt an early pattern to characterize early-stage time series evolution from a global view. The early pattern for content e can be defined as $\{\frac{N_e(t_1)}{N_e}, \frac{N_e(t_2)}{N_e}, \cdots, \frac{N_e(t_m)}{N_e}\}$. The normalized term N_e in the denominator is designed to incorporate the global (or future) information. Hence, the early pattern reflects the early-stage evolution of count series normalized on the popularity.

We further adopt K-means algorithm to cluster contents, and divide content set E into K groups, i.e., $E = \cup_{k=1}^{K} E_k$. For each group E_k, every content $e \in E_k$ shares similar early pattern. Concretely, the L_2 distance is adopted to measure the similarity.

B. Early Indicator Representation

For contents in one early pattern group, we use a parametric *evolving function* to fit count series $\mathcal{S}_e$. Consider content e, use $A_e(t_j)$ to denote post count increments in time unit $[t_{j-1}, t_j]$, i.e., $A_e(t_j) = N_e(t_j) - N_e(t_{j-1})$. Then the evolving function can be defined as

$$A_e(t_j) = h_e^1 \cdot (\gamma + e^{-\alpha t_j}) + h_e^2 \cdot \sum_{t_p < t_j} A_e(t_p) e^{-\beta(t_j - t_p)} \\ + h_e^3 \cdot (A_e(t_j) - A_e(t_{j-1})) \tag{1}$$

Justification of the Model:

- The first term with coefficient h_e^1 captures current *influence* triggered by the content. Parameter γ denotes base influence, while $e^{-\alpha t_j}$ captures decaying influence as time goes by.
- The second term with coefficient h_e^2 characterizes effect of previous behaviors, and exponential factor $e^{-\beta(t_j - t_p)}$ captures the aging effect. The summation of all previous effects can reflect the capability of one content to trigger new post increment, which we name as *attractiveness*.
- The third term with coefficient h_e^3 models effect of second-order increment of post count, and we call it as *potentiality*.
- Parameters α, β, γ are common scaling factors for one early pattern group and reflect shape of count series.

The evolving function (1) can be estimated by minimizing the square loss.

$$\min \sum_{e \in E_k} \sum_{j \le n_e} \big(\frac{A_e(t_j) - A_e^*(t_j)}{A_e^*(t_j)}\big)^2, \tag{2}$$

where $A_e^*(t_j)$ denotes ground-truth increment of post count in time unit $[t_{j-1}, t_j]$.

By solving (2), we obtain three coefficients h_e^1, h_e^2, h_e^3 for each content e, which capture the early *influence*, *attractiveness* and *potentiality* respectively. These coefficients can be a representation for effect of observed time series on future series evolution, so we call them as *early indicators*. In the following, we will leverage early indicators to bridge the gap between observable features and early patterns.

C. Early Pattern and Indicator Modeling

We proceed to model the relationship among features, early indicators and early patterns. Bayesian network is a probabilistic directed acyclic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). In the DAG, nodes represent random variables, which may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies. If there is an edge from node A to node B, we say node A is a parent of node B and variable B conditionally depends on variable A. Each node is associated with a probability function that takes (as input) a particular set of values for the node's parent variables, and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node. In machine learning domain, Bayesian network can model features and labels as random variables, and study their probabilistic relations. The model learning is based on a generative perspective which aims to maximize a likelihood function. As a generative model, Bayesian network is suitable for situations with insufficient and under-sampled data, and has extensive applications in biomonitoring [12], information retrieval [13] and geographical analysis [14].

Here we utilize Bayesian network to model the relationship among features x, early indicators h and early pattern category y (here, y is a integer variable ranged from 1 to K, and represents which early pattern group the content belongs to). The early pattern category y is output layer of the network, and the features x_i, $i = 1, \cdots, 9$ form observable layer. The early indicators h_j, $j = 1, 2, 3$ compose latent layer between the early pattern and features. We define two composite nodes, $H = \{h_1, h_2, h_3\}$ and $X = \{x_1, x_2, \cdots, x_9\}$. The graphical representation of y, H and X can be denoted by Fig. 3.

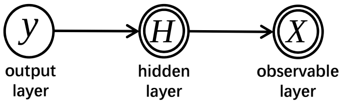

Fig. 3: Graphical representation of early pattern category y, early indicators H and features X, where H and X are two composite nodes.

For test data, we only have early information, i.e., the observable features X. We need to use features to infer the early indicators H as well as the early pattern y. One method is to leverage the posterior probability to make an optimal decision. Concretely, we can use H with the highest $p(H|X)$ as predicted early indicators and y with the highest $p(y|H)$ as predicted early patterns. According to Bayes Theorem, we have the equations for the two posterior probabilities:

$$p(y|H) = \frac{p(H|y) \cdot p(y)}{p(H)} = \frac{p(H|y) \cdot p(y)}{\sum\limits_y p(H|y) \cdot p(y)}, \tag{3}$$

$$\begin{aligned} p(H|X) &= \frac{p(X|H) \cdot p(H)}{p(X)} \\ &= \frac{p(X|H) \cdot \sum\limits_y p(H|y) \cdot p(y)}{\sum\limits_H p(X|H) \cdot \sum\limits_y p(H|y) \cdot p(y)}, \end{aligned} \tag{4}$$

To compute $p(y|H)$ and $p(H|X)$, we need to estimate $p(H|y)$, $p(X|H)$ and $p(y)$ with training data. In the following, we discuss how to estimate them in detail.

Firstly, we consider conditional probability $p(H|y)$. We observe that h_1, h_2 and h_3 are weakly correlated with each other, so we assume they are conditionally independent for low computational complexity. Thus, we have

$$p(H|y) = p(h_1|y) \cdot p(h_2|y) \cdot p(h_3|y). \tag{5}$$

By studying the characteristics of data frequency histogram, we observe that h_j tends to have a concentrated distributions for different early pattern categories. Assume h_j is a continuous random variable. Since h_j is nonnegative, we choose a unimodal distribution, Gamma distribution, to characterize it, i.e.,

$$h_j|y = k \sim Gamma(a_k^j, b_k^j), \tag{6}$$

where a_k^j is a shape parameter, and b_k^j is a rate parameter for Gamma distribution. To keep notation clean, we let $\theta_1 = \{a_k^j, b_k^j \mid 1 \le k \le K, 1 \le j \le 3\}$.

Then we proceed to probe into $p(X|H)$. We also conduct correlation test on each feature x_i with each h_j, and find that i) for one early indicator h_j, some features are strongly correlated while others are weakly correlated, and ii) for different early indicators, features with strong correlation are distinct. It is straightforward but unreasonable to equally consider all features conditionally dependent on three early indicators. Instead, we assume feature x_i is conditionally dependent on a set of early indicators, denoted by $Pa(x_i)$ (the parent nodes of x_i). $Pa(x_i)$ can possibly cover all, any two, any one, or none of three early indicators. Then we have

$$p(X|H) = \prod_{i=1}^{9} p(x_i|Pa(x_i)). \tag{7}$$

Since h_j is a continuous value, it is intractable to handle the conditional probability $p(x_i|h_j)$. Here we do a technical approximation and hash h_j into B buckets. By hashing, we convert the continuous h_j into discrete values, which we denote as h'_j for discrimination. We observe that x_i tends to have a concentrated and symmetric distribution for different h'_j, so we adopt Gaussian distribution to characterize it.

$$x_i|Pa(x_i) \sim Gaussian(\mu_b^i, \sigma_b^i), \tag{8}$$

where μ_b^i and σ_b^i are mean and standard deviation of Gaussian distribution, respectively. Here $1 \le b \le B^{|Pa(x_i)|}$. Assume $\theta_2 = \{\mu_b^i, \sigma_b^i \mid 1 \le b \le B^{|Pa(x_i)|}, 1 \le i \le 9\}$.

The prior probability of y obeys a discrete distribution, so we assume $p(y = k) = c_k$ and let $\theta_3 = \{c_k \mid 1 \le k \le K\}$.

The problem remains as two parts: i) optimize the parameters in two conditional distributions, i.e., θ_1, θ_2, and the parameters in prior probability, i.e., θ_3, and ii) optimize constitution of each $Pa(x_i)$. The first problem is to learn the parameters in Bayesian network, and the second problem is to learn the network structure.

D. Parameter and Structure Learning

Assume G is topological graph of Bayesian network. The basic objective for model training is to maximize the likelihood,

$$\begin{aligned}&\mathcal{L}(\theta_1, \theta_2, \theta_3, G)\\ =&\prod_e p(y_e) \cdot p(H_e|y_e) \cdot p(X_e|H_e)\\ =&\prod_e p(y_e) \cdot \prod_{j=1}^{3} p(h_e^j|y_e) \cdot \prod_{i=1}^{9} p(x_e^i|Pa(x_i)).\end{aligned} \tag{9}$$

The intuition of maximizing (9) is to make the trained Bayesian network interpret observable information as much as possible. Besides, we also need to take model complexity into account. Based on Minimal Description Length principle, our objective can be written as

$$\min \lambda|\theta_2| - \log \mathcal{L}(\theta_1, \theta_2, \theta_3, G), \tag{10}$$

where $|\theta_2|$ in the first term denotes number of parameters in θ_2, which is equivalent to number of edges from early indicators H to features X in Bayesian network. λ is a weight parameter, which can balance the importance between likelihood and model complexity.

Particularly, optimization for θ_1 and θ_3 in (10) is independent of that for θ_2, so we separate them apart to reduce computational cost. Firstly, we can minimize

$$\begin{aligned}&l_1(\theta_1, \theta_3)\\ =&-\log \mathcal{L}_1(\theta_1, \theta_3, G_B)\\ =&-\sum_e [\log p(y_e) + \sum_{j=1}^{3} \log p(h_e^j|y_e)]\\ =&-\sum_e \sum_{k=1}^{K} \chi_e^k [\log c_k + \sum_{j=1}^{3} \log Gamma(h_e^j|a_k^j, b_k^j)].\end{aligned} \tag{11}$$

Here χ_e^k is an eigenfunction, where $\chi_e^k = 1$ if $y_e = k$ and otherwise, $\chi_e^k = 0$. We adopt Stochastic Gradient Descendant (SGD) method to minimize (11).

Then we proceed to minimize

$$\begin{aligned}&l_2(\theta_2, G)\\ =&|\theta_2| - \log \mathcal{L}_2(\theta_2, G)\\ =&|\theta_2| - \sum_e \sum_{i=1}^{9} \log p(x_e^i|Pa(x_i))\\ =&|\theta_2| - \sum_e \sum_{i=1}^{9} \sum_{b=1}^{B^{|Pa(x_i)|}} \chi_e^b \log Gaussian(x_e^i|\mu_b^i, \sigma_b^i).\end{aligned} \tag{12}$$

If network topology G is given, then we can optimize θ_2 by

$$\frac{\partial l_2}{\partial \mu_b^i} = 0, \quad \frac{\partial l_2}{\partial \sigma_b^i} = 0. \tag{13}$$

By solving (13), we obtain

$$\hat{\mu}_b^i = \frac{\sum_e \chi_e^b x_e^i}{\sum_e \chi_e^b}, \quad \hat{\sigma}_b^i = \sqrt{\frac{\sum_e \chi_e^b (x_e^i - \hat{\mu}_b^i)^2}{\sum_e \chi_e^b}}. \tag{14}$$

We further adopt Hill-Climbing structure learning method to search optimal topology.

III. Experiment Results

In this section, we conduct experiments on three real-world datasets to verify our model.

Dataset Information. Twitter is the largest social network in the world. We use Twitter search API to collect tweets by real-time densely crawling from Aug. 13th, 2017 to Sep. 10th, 2017. Moreover, Sina Weibo is a prevalent Chinese social platform with about 0.4 billion monthly active users and also provides API for data crawling. The Weibo data set is ranged from Aug. 10th, 2017 to Dec. 22th, 2017. Our special data set is WeChat, a burgeoning social media with over 0.9 billion daily login users. WeChat Official Accounts established by individuals or companies can post articles with social information and users would share these articles in WeChat Moments. This sharing would give rise to more resharing by their fans, thus forming social content virality. We study the popularity of articles published by WeChat Official Accounts. For each data set, we filter out social contents with posts less than 50 and Table I shows the detailed information about three data sets. Each data set contains post information (like posted time and user ID), user information (like follower number) and friendship network information.

TABLE I: Basic information of data sets

Data Set	Twitter	Weibo	WeChat
Content Type	hashtag	topic	article
Post Type	tweet	microblog	share
#Contents	5,763	1,168	1 thousand
#Posts	529,059	410,733	6 million
#Follow Edges	39,614,487	5,934,504	7 million

Experiment Setup. As is depicted in Section III, observable features are extracted from observed time sequence in observing duration. Since information dissemination in Weibo appears to be slower than Twitter and WeChat, we basically consider 1 hour observation time for Twitter and WeChat and 2 hour for Weibo. Moreover, for each data set, we randomly choose 80% contents as training data and remaining contents as test data.

We adopt two classification metrics and two regression metrics to evaluate prediction performance in a multifaceted way. For classification, we consider a popularity threshold which can divide the contents into hot and non-hot contents at ratio $1:4$. Then each content is assigned with a 0-1 label indicating hot or non-hot. The two classification metrics are *F1-Score* and *Coverage*$@k$. F1-Score aggregates two-fold performance measured by recall and precision. We use F1-Score to evaluate the general classification performance of these methods in early-stage prediction. Coverage$@k$ is defined as the ratio of accurately detected top-k popular contents. For instance, if the method detects n contents that are among realistic top-k popular contents, then the Coverage$@k$ will be $\frac{n}{k}$. This metric measures ability of detecting extremely popular contents. Also, we consider two regression metrics *Mean Absolute Percent Error (MAPE)* and *Pearson's Correlation Coefficient (PCC)*.

We compare with six strong baselines proposed by recent studies: i) *Hawkes Process* [15]. ii) *SEISMIC* [6]. iii) *BEEP* [16]. iv) *ESP-TAN* [17]. v) *LARM* [11]. vi) *Support Vector Regression (SVR)*. These baselines can be divided into three categories: point process models (Hawkes, SEISMIC), generative feature driven models (PreWhether, BEEP) and discriminative feature driven models (LARM, SVR).

Results and Discussions. We organize classification and regression results in Table II. For classification task, Hawkes provides the poorest results in three data sets. Although SEISMIC gives acceptable classification on Twitter, it also performs poorly on Weibo and WeChat. These indicate that point process models are not suitable for early-stage classification, since they rely too much on sufficient observation. By contrast, LARM and SVR perform slightly better than two point process models, but the classification results are moderate compared with other Bayesian perspective methods. BEEP, and ESP-TAN provide neck-to-neck performance. In comparison, ESP-TAN gives slightly better prediction since it utilizes more features and adopts structure learning method to conduct feature selection. It is noteworthy that EPAB provides outstanding performance on three data sets. This verifies the argument that early patterns and early indicators link the early observed series to future evolution and make the popularity more predictable using observable features. This demonstrates that EPAB is competent in detecting hot content at early-stage.

As for regression, in general, the results show no significant difference from classification. Hawkes and SEISMIC both fail to fit the practical post count since they incline to provide extreme predicted value based on insufficient early information. Particularly, the MAPEs of SEISMIC for both Twitter and Weibo exceed 2, and one possible reason could be that SEISMIC makes some parametric assumptions that are custom-made for single post popularity prediction (like functional form of memory kernel [18]), which limits its generality to content popularity prediction. The regression performance of SVR model is more acceptable than classification on Weibo and WeChat, but it gives prediction with considerable deviation on Twitter. LARM performs relatively well for regression on three data sets, but its performance is still not impressive enough compared with EPAB. Remarkably, EPAB achieves significant performance improvement in both MAPE and PCC on three data sets. This verifies the argument that early patterns and early indicators link the early observed series to future evolution and make the popularity more predictable using observable features.

We probe into different observation time to study the performance variation of each method. The results is shown in Fig. 4 In the figure, we can see that with observation time increasing, both MAPE and F1-score performance improves greatly and variance of MAPE (reflected by width of the box) reduces, especially for SEISMIC and Hawkes. It indicates that point process models rely on observation time considerably. When the observation time is long enough (like), point process model could perform better than other feature driven models, which demonstrates that point process models are more suitable for long observation based prediction. By contrast, Bayesian perspective models including EPAB tend to be less sensitive to different observation time.

IV. Conclusion

This paper targets early-stage prediction for social content popularity and proposes an early pattern aware Bayesian model (EPAB). In EPAB, we use early patterns and early indicators to make future popularity predictable using observable features, based on a Bayesian perspective. Our evolving function helps to express latent relationship between early time series and

TABLE II: Early-stage experiment results for Twitter, Weibo and WeChat. The observation time for three data sets is 1h, 2h, 1h, respectively.

	Twitter				Weibo				WeChat			
	F1	C@50[1]	MAPE	PCC	F1	C@50	MAPE	PCC	F1	C@50	MAPE	PCC
SEISMIC	0.7831	0.7352	2.1723	0.2803	0.5122	0.6128	2.0278	0.1545	0.5712	0.4803	1.9870	0.1126
Hawkes	0.5143	0.4230	1.8564	0.4694	0.5076	0.4912	1.8147	0.1974	0.5543	0.6754	1.7245	0.2135
BEEP	0.8493	0.9283	-[2]	-	0.6328	0.7227	-	-	0.8017	0.9256	-	-
ESP-TAN	0.8547	0.8701	-	-	0.7340	0.8821	-	-	0.8033	0.8922	-	-
LARM	0.7571	0.8139	0.8231	0.7227	0.6107	0.8326	1.0317	0.5840	0.7461	0.9005	0.7253	0.8499
SVR	0.6557	0.6724	1.2051	0.5402	0.5349	0.6725	0.8815	0.6925	0.6211	0.8645	0.6409	0.8133
EPAB	**0.8813**	**0.9811**	**0.6343**	**0.7929**	**0.7604**	**0.9251**	**0.7704**	**0.7118**	**0.8394**	**0.9296**	**0.5517**	**0.8901**

[1] The notion C@k is short for Coverage@k.

[2] BEEP and ESP-TAN are only designed for classification.

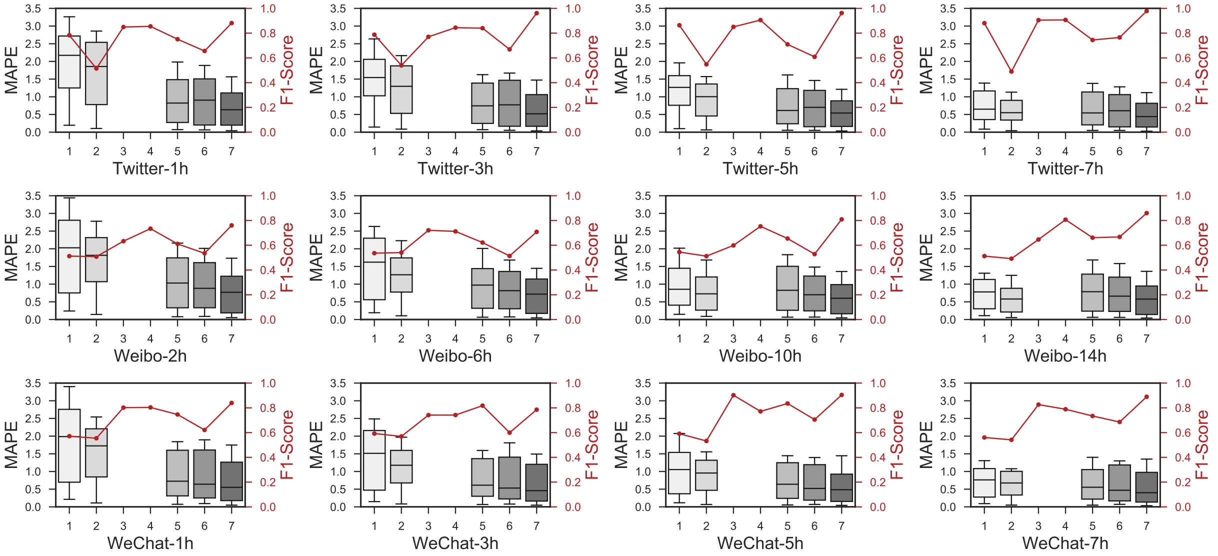

Fig. 4: MAPEs and F1-Scores for Twitter, Weibo and WeChat under different observation time. The box plot depicts five points distribution of MAPE, while the red dotted curve denotes F1-score (1: SEISMIC, 2: Hawkes, 3: BEEP, 4: ESP-TAN, 5: LARM, 6: SVR, 7: EPAB).

future evolution, and Bayesian network model sheds insights on feature selection for early-stage popularity prediction. Our experiment results show that EPAB can accurately predict content popularity at very early stage for Twitter, Weibo and WeChat data sets.

References

[1] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[2] Justin Cheng, Lada A. Adamic, P. Alex Dow, Jon M. Kleinberg, and Jure Leskovec. Can cascades be predicted? In *WWW*, 2014.

[3] Qun Wu, Tian Wang, Yiqiao Cai, Hui Tian, and Yonghong Chen. Rumor restraining based on propagation prediction with limited observations in large-scale social networks. In *ACSW*, 2017.

[4] Xiao Lin, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. Learning and transferring social and item visibilities for personalized recommendation. In *CIKM*, 2017.

[5] Hideaki Kim, Noriko Takaya, and Hiroshi Sawada. Tracking temporal dynamics of purchase decisions via hierarchical time-rescaling model. In *CIKM*, 2014.

[6] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *SIGKDD*, 2015.

[7] Weiwei Liu, Zhi-Hong Deng, Xiuwen Gong, Frank Jiang, and Ivor W. Tsang. Effectively predicting whether and when a topic will become prevalent in a social network. In *AAAI*, 2015.

[8] Lexing Xie, Apostol Natsev, John R. Kender, Matthew L. Hill, and John R. Smith. Tracking visual memes in rich-media social communities. In *ICWSM*, 2011.

[9] Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye, and Zhe Zhao. Predicting bursts and popularity of hashtags in real-time. In *SIGIR*, 2014.

[10] Bidisha Samanta, Abir De, Abhijnan Chakraborty, and Niloy Ganguly. Lmpp: A large margin point process combining reinforcement and competition for modeling hashtag popularity. In *IJCAI*, 2017.

[11] Changsha Ma, Zhisheng Yan, and Chang Wen Chen. LARM: A lifetime aware regression model for predicting youtube video popularity. In *CIKM*, 2017.

[12] Xia Jiang and Gregory F. Cooper. A bayesian spatio-temporal method for disease outbreak detection. *Journal of the American Medical Informatics Association*, 2010.

[13] Luis M. de Campos, Juan M. Fernández-Luna, and Juan F. Huete. Bayesian networks and information retrieval: an introduction to the special issue. *Information Processing Management. Elsevier*, 2004.

[14] Monidipa Das, Soumya K. Ghosh, Pramesh Gupta, V. M. Chowdary, Ravoori Nagaraja, and Vinay K. Dadhwal. FORWARD: A model for forecasting reservoir water dynamics using spatial bayesian network (spabn). *TKDE*, 2017.

[15] Peng Bao, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In *WWW*, 2015.

[16] Xiao Ma, Xiaofeng Gao, and Guihai Chen. Beep: A bayesian perspective early stage event prediction model for online social networks. In *ICDM*, 2017.

[17] Mahtab Jahanbani Fard, Ping Wang, Sanjay Chawla, and Chandan K. Reddy. A bayesian perspective on early stage event prediction in longitudina data. *TKDE*, 2016.

[18] Karthik Subbian, B. Aditya Prakash, and Lada A. Adamic. Detecting large reshare cascades in social networks. In *WWW*, 2017.