Discerning Influence Patterns with Beta-Poisson Factorization in Microblogging Environments

Wei Zhao, Ziyu Guan* , Yuhui Huang, Tingting Xi, Huan Sun, Zhiheng Wang and Xiaofei He, *Fellow, IAPR*

Abstract—Social influence analysis in microblogging services has attracted much attention in recent years. However, most previous studies were focused on measuring users' (topical) influence. Little effort has been made to discern and quantify how a user is influenced. Specifically, the fact that user i retweets a tweet from author j could be either because i is influenced by j (i.e., j is a topical authority), or simply because he is "influenced" by the content (interested in the content). To mine such influence patterns, we propose a novel Bayesian factorization model, dubbed Influence Beta-Poisson Factorization (IBPF). IBPF jointly factorizes the retweet data and tweet content to quantify latent topical factors of user preference, author influence and content influence. It generates every retweet record according to the sum of two causing terms: one representing author influence, and the other one derived from content influence. To control the impact of the two terms, for each user IBPF generates a probability for each latent topic by Beta distribution, indicating how strongly the user cares about the topical authority of the author. We develop an efficient variational inference algorithm for IBPF. We demonstrate the efficacy of IBPF on two public microblogging datasets.

Index Terms—Social influence, Poisson factorization, Microblogging.

1 Introduction

MICROBLOGGING services such as Twitter have become popular social media where users can freely follow other users to receive various messages (tweets) from them. One key feature is, they provide an information diffusion mechanism (retweeting) which resembles word-of-mouth in reality. This has triggered a lot of studies that analyze users' behavior in information diffusion [28], [40] and model the underlying dynamics [8], [32], [43], [56], [60].

Among the studies revolving around microblogging data, social influence analysis has become a hot research topic in recent years [2], [5], [15], [21], [30], [31], [32], [33], [35], [42], [51]. Understanding influence patterns among users can benefit many applications, such as tweet recommendation [36], retweet prediction [56] and viral marketing [38]. Although it is difficult to define influence exactly, it can be reflected from social signals of users, e.g. followship [51], retweets [33], [35], mentions [35] and favors [15]. Among the different signals, retweets are regarded as salient evidences of influence and widely used by influence measures [39]. A retweet is intrinsically a ternary record $< i, j, m >$, providing an evidence that user i has retweeted tweet m from author j. Note that j may not be the original author who wrote m. We use "author" to represent the notion that a user gets retweeted with respect to the tweet. It has been shown that in microblogging environments direct influence dominates [32], [56].

However, most existing influence mining techniques for microblogging services are focused on measuring to what degree (and on what topics) a user influences other users. Little effort has been made to discern and quantify how a user is influenced. Specifically, the fact that user i retweets a tweet from author j could be either because i is influenced by j, or simply because he is "influenced" by the tweet content. For example, for topics such as "President Trump" a user may only retweet from the political commentators he follows, while for funny and joke tweets he would retweet from anyone, as long as the content interests him. The underlying motive could be discerned as follows: (1) if the involved author is influential on the topics of the tweet, then it is very likely that the user is influenced by the author; (2) if we often observe that the user retweets tweets with similar topics from non-influential authors, then content influence is more likely to explain the retweets. Most traditional author influence measures treat every retweet as an evidence of author influence and use them to assess author influence. However, in this work we argue that we should discern different influence patterns in retweets to better measure influence in microblogging environments. Mining such influence patterns can help us better understand the information diffusion process and benefit related applications. A similar idea of distinguishing viral users and viral topics is described in [18]. Nevertheless, the proposed model uses all latent factors to jointly explain each retweet record, which hinders discerning the two causing factors.

The retweet data can be represented as a sparse tensor encoding users' implicit feedbacks [22] (i.e. there is no explicit negative signals). Moreover, authors are usually influential on only a few topics [18]; the tweets are very

* *Corresponding author*

- *W. Zhao, Z. Guan and T. Xi are with the State Key Laboratory of Integrated Services Networks, School of Computer Science and Technology, Xidian University, Xian, China 710071. E-mail: ywzhao@mail.xidian.edu.cn; zyguan@xidian.edu.cn; ixtgtg@gmail.com*
- *Y. Huang is with FABU Technology, Hangzhou, China 310030. E-mail: solacehuang@gmail.com*
- *H. Sun is with the Department of Computer Science and Engineering, Ohio State University, OH USA 43210. E-mail: sun.397@osu.edu*
- *Z. Wang is with the Department of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China 454003. E-mail: wzhenry@eyou.com*
- *X. He is with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, CN 310027. E-mail: xiaofeihe@cad.zju.edu.cn*

Manuscript received April 19, 2005; revised August 26, 2015.

short, containing only a few topics (typically 1) [52]. These characteristics of retweet data call for a method which can efficiently learn sparse latent structures from implicit feedbacks. Poisson factorization is a technique that meets all the above requirements. Recently, Poisson factorization has been applied on different user implicit feedback datasets, generating superior performance in recommendation [12], [13], [59].

In this paper, we propose a novel Bayesian factorization model, dubbed Influence Beta-Poisson Factorization (IBPF), to mine topical influence patterns from retweet data. IBPF jointly factorizes the retweet tensor and tweet content to quantify latent topical factors of user preference, author influence and content influence simultaneously. In order to learn sparse latent factors, we impose global sparse Gamma priors on the latent factors except those for user preference. The latent preference factors of each user have a separate Gamma prior in which the rate parameter is in turn generated by a global Gamma prior. The customized rate parameters account for the variance in user activity, i.e. some tending to retweet in more topics than others. IBPF generates every retweet record according to the sum of two causing terms: one representing author influence, and the other one representing content influence. To control the impact of the two terms, for each user IBPF generates a probability for each latent topic by Beta distribution, indicating how strongly the user cares about the topical authority of the author on that topic. If the author is influential on the involved topics, the former term would dominate, while the latter term would be in charge if we observe that many similar tweets are retweeted from non-influential authors. We develop an efficient variational inference algorithm for IBPF. The optimization is efficient in that only nonzero elements of the retweet tensor and the content matrix need to be processed. To better learn topics from short tweets, we initialize the model with topics learned from the Biterm Topic Model (BTM) [52] which is exclusively designed for short texts.

The contributions of this paper are summarized as follows. (1) We study how to discern author influence and content influence for microblogging environments. The results can help to better understand information diffusion and benefit applications such as recommendation. (2) We develop a novel Bayesian factorization model IBPF for this task. IBPF is efficient and explicitly differentiates the two influence sources. Although we develop IBPF on retweet data, it can also be applied on other ternary relational data for influence analysis, e.g. favor data where $< i, j, m >$ means user i favors tweet m from author j. (3) We demonstrate the efficacy of IBPF on two microblogging datasets collected from Twitter and Sina Weibo respectively. We also explore integrating the learning results with state-of-art features for retweet prediction/tweet recommendation. The results show the performance can be boosted.

2 Related Work

In this section, we review three fields that are most related to our work: social influence analysis, retweet prediction/tweet recommendation, and Poisson factorization.

2.1 Social Influence Analysis

With the rapid growth of social Websites such as Facebook, Flickr and Twitter, much work has been done for analyzing and quantifying social influence. The pioneering work of Kempe *et al.* proposed two simple influence models and tried to locate the key influencers by influence maximization under these models [25]. Early studies on social influence were mainly focused on general influence [1], [9], [14]. Although general influence is useful for explaining global behaviors, it cannot well handle fine-grained local behaviors [34]. Tang *et al.* were among the first to quantitatively measure topic-level influence and proposed a factor graph model for the task [46].

In Microblogging environments, a lot of methods for measuring user (topical) influence have been proposed. Here we only provide a brief overview of related work. Readers can refer to [39] for a complete survey. Among those methods, some were built on the relatively static follow relationships [2], [51], while recently users' daily behaviors were found to be more effective for influence evaluation, including retweets [33], [35], mentions [35], replies [35] and favors [15]. Among different behaviors, retweet is deemed as a strong indication of influence [39]. In terms of methodology for computing influence scores, most methods fall into three categories: (1) Feature characterization [10], [35]. This kind of methods introduced various features to describe each user, trying to capture influence from different aspects, and then aggregated these features properly to generate the final influence scores. (2) Link analysis [21], [42], [51]. Methods of this style constructed graphs for users (and tweets) using various relational information (e.g. follow, retweet) and performed link analysis on the graphs to assess user influence. (3) Probabilistic generative models [2], [32]. In these methods, influence was modeled as latent variables which were learned by inference.

However, previous works were mostly focused on the "author" side. They did not investigate in detail how a "user" was influenced. For some users, the influence may only come from the topic itself. In this paper, we develop a novel probabilistic factorization model to discern how a user is influenced, which is the key difference compared to previous work. In [18], Hoang and Lim studied a similar problem of learning viral authors and viral topics from retweets. They proposed a tensor factorization model call V2S to address the problem. Our method is different from theirs in that: (1) V2S explains each retweet by both viral user and viral topic factors (thus cannot clearly separate them), while IBPF explicitly separates the two explanatory factors and uses the sum of them to generate each retweet; (2) V2S requires negative feedbacks (view but not retweet) which are hard to obtain, while IBPF can naturally handle implicit feedback data; (3) As aforementioned, with proper gamma priors IBPF could well capture the sparse latent structures in microblogging data, while V2S does not enjoy this property.

2.2 Retweet Prediction/Tweet Recommendation

Retweet prediction aims to predict whether a user will retweet the tweets from his friends [53]. A related problem is predicting the spread or popularity of a tweet [53],

[60]. In this paper we are focused on the former problem. Boyd *et al.* explored the reasons for retweeting [4]. Suh *et al.* investigated the correlation of several user statistical features with retweeting [43]. Yang *et al.* proposed a semi-supervised model with 22 features for retweet prediction, but the features were not revealed and the model only worked well on spread prediction. A social influence locality feature was proposed in [55], [56] and shown to be effective for retweet prediction. Recently, researchers also tried to use different models for retweet prediction, e.g. nonparametric generative models [57] and deep neural networks [58].

Tweet recommendation aims to address the information overload problem and tries to recommend tweets that the target user likes. It is closely related to retweet prediction since studies on this problem treat retweets as preference signals and perform evaluation accordingly. The difference is that retweet prediction outputs binary prediction results, while tweet recommendation shows users a ranked list of tweets. Like retweet prediction, most works on tweet recommendation also exploited different features to learn user preference on tweets [7], [11], [20], [29], [36], [48].

Nevertheless, none of previous works tried to discern author influence and content influence for retweet prediction or tweet recommendation. In [26], a probabilistic generative model was proposed for recommendation in Twitter which generated each word of a user from either personal interests or followees interests. However, it did not learn from explicit influence evidences, e.g. retweets. A tweet favored by a user's followees would be ranked high in recommendation, but the user may never retweet such tweets from the followees. In [23], Jiang *et al.* tried to model individual preference and interpersonal influence separately for tweet recommendation. However, the user-user influence learned was not topical and they did not exploit the ternary retweet data either. In the more general social recommendation literature, Wang *et al.* [50] tried to infer social influence on users' exposure to items and recommended items considering both user preference and the estimated exposure. In tweet recommendation, user exposure is (almost) explicit via the "following" mechanism. Our influence analysis is focused on the preference part and is orthogonal to their work. Our aim is to learn author influence and content influence patterns from ternary retweet data and investigate their helpfulness in retweet prediction/tweet recommendation.

2.3 Poisson Factorization

Poisson factorization is closely related to nonnegative matrix factorization [6]. Recently, different Poisson factorization models have been developed for applications such as recommendation [12], [13], [59], dynamic community detection [41] and anomaly detection [47]. The key difference between IBPF and existing Poisson factorization models is that we explain the data generation by combining two causing terms and control the impact of the two terms by Beta random variables. Since Beta distribution is not the conjugate prior for Poisson distribution, we develop an approximated variational inference algorithm for IBPF.

3 THE IBPF MODEL

This section details the design and specification of the IBPF model. Firstly, we formally describe the problem and define notations used by IBPF. Then model design and specification follow.

3.1 Problem Formulation and Notations

We are given a set of users and a set of tweets that are written or retweeted by the users. Let $m \in \{1, \ldots, M\}$ denote the index of the M tweets. The tweets are written with a vocabulary of N words and $n \in \{1, \ldots, N\}$ denotes the word index. Let $i \in \{1, \ldots, I\}$ be the index of I *users* who have at least retweeted one tweet, and $j \in \{1, \ldots, J\}$ be the index of J *authors* who have been retweeted by at least one user. With a slight abuse of notation, we also use the index variable to refer to the corresponding entity, e.g. i can also represent the corresponding user. The retweet collection $\mathcal{R} = \{< i, j, m >\}$ contains retweet records and can be represented by a sparse tensor $\mathbf{R} \in \{0, 1\}^{I \times J \times M}$ where

$$R_{ijm} = \begin{cases} 1, & \text{if } < i, j, m > \in \mathcal{R} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The tweet text contents can be represented as a sparse matrix $\mathbf{W} \in \mathbb{I}^{M \times N}$:

$$W_{mn} = \begin{cases} occr\_num, & \text{if } n \text{ occurs in } m \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Then the problem is, given $\mathbf{R}$ and $\mathbf{W}$, to infer the latent topical factors for author influence, user preference and content influence for users.

The intuition is that each retweet record $< i, j, m >$ is an evidence of influence, meaning that user i is influenced by either author j or the content of m only. Let T be the number of topics and t be the index. We define $\mathbf{u}_i$, $\mathbf{v}_j$, $\boldsymbol{\theta}_m$ and $\boldsymbol{\beta}_n$ to be length-T vectors containing user i's topical preference intensities, author j's topical influence intensities, tweet m's topic intensities and word n's topic intensities respectively. We further define a length-T vector of probabilities, $\boldsymbol{\alpha}_i$, where α_{it} measures the likelihood that user i follows authority authors on topic t. Since they are probabilities, we have $\alpha_{it} \in [0, 1], \forall i, t$. The lower the probability, the more likely the user follows her own interests. Hence, the vector $\boldsymbol{\alpha}_i$ can be regarded as encoding user i's influence pattern. The problem then becomes inferring $\mathbf{u}_i$, $\mathbf{v}_j$, $\boldsymbol{\theta}_m$, $\boldsymbol{\beta}_n$ and $\boldsymbol{\alpha}_i$ for all i, j, m, n, based on $\mathbf{R}$ and $\mathbf{W}$. A summarization of notations is provided in Table 1.

3.2 Model Design

The basic idea of Poisson factorization is that each observed variable is generated by a Poisson distribution where the parameter is decided by a dot product of the involved latent variables. In our case, a straightforward idea is to generate the retweet tensor by means of CANDECOMP/PARAFAC (CP) decomposition [27]: $R_{ijm} \sim \mathrm{Poisson}(\sum_t u_{it} v_{jt} \theta_{mt})$. Intuitively, this accumulates the author influence (v_{jt}) and user preference (u_{it}) on topics of the tweet (θ_{mt}), representing the case that i is influenced by j. However, authors are usually influential on only a few topics [18]. They cannot be influential on every topic. This could be learned from the

TABLE 1
A summarization of notations.

Symbols	Descriptions
$I/J/M/N/T$	No. of users/authors/tweets/words/topics
$i/j/m/n/t$	Idx of users/authors/tweets/words/topics
$\mathbf{u}_i$	user i's latent topical preference
ξ_i	user i's customized prior rate parameter
$\boldsymbol{\alpha}_i$	influence pattern of user i
$\mathbf{v}_j$	author j's latent topical influence
f_j	author j's scale factor
$\boldsymbol{\theta}_m$	tweet m's latent topics
$\boldsymbol{\beta}_n$	word n's topic associations
$\mathbf{R}$, $\mathbf{W}$	observed retweets and tweet contents
a^u, a^ξ, a^v, a^θ, a^β, a^α, b^α, a^f	shape hyperparameters of Gamma or Beta prior distributions for the corresponding variables
b^ξ, b^v, b^θ, b^β, b^f	rate hyperparameters of Gamma prior distributions for corresponding variables

retweet data: authors get retweeted frequently on certain topics but receive few retweets on (or never concern about) other topics. If j's influence factors does not match the topics of m, $\sum_t u_{it}v_{jt}\theta_{mt}$ will be very small regardless of $\mathbf{u}_i$, rendering it not a good explanation for R_{ijm}. For example, a tweet about an emergency could attract many retweets, but the author may rarely post or retweet about the corresponding topics.

The above example means that the user is simply interested in the tweet itself. To model such content influence cases, we construct two causing terms for data generation: $\sum_t \alpha_{it}u_{it}v_{jt}\theta_{mt}$ and $\sum_t(1-\alpha_{it})u_{it}\theta_{mt}$. The first term is activated when user i tends to only follow topical authorities on the topics of tweet m, i.e. the probability α_{it} is large; the second term handles the case where user i is interested in the content (α_{it} is small) regardless of the author. Note that the second term does not involve $\mathbf{v}_j$ since the user only cares about the content in that case. We model each retweet evidence by the combination of the two causing terms: $R_{ijm} \sim \mathrm{Poisson}(\sum_t \alpha_{it}u_{it}v_{jt}\theta_{mt} + \sum_t(1-\alpha_{it})u_{it}\theta_{mt})$. Intuitively, $\boldsymbol{\alpha}_i$ encodes the influence pattern of user i. α_{it} should be large if we observe that user i always retweet from topical authorities on topic t; it should be small if user i retweet a lot from non-influential authors on topic t.

A straightforward idea for discerning authority influence and content influence is to define a binary "switch" random variable b_{ijm} for each retweet evidence. If $b_{ijm} = 1$, $R_{ijm} \sim \mathrm{Poisson}(\sum_t u_{it}v_{jt}\theta_{mt})$; if $b_{ijm} = 0$, we generate R_{ijm} by $\mathrm{Poisson}(\sum_t u_{it}\theta_{mt})$. However, this idea has three drawbacks: (1) it requires defining a binary variable for every element in $\mathbf{R}$. The space and time costs ($O(IJM)$) could quickly become prohibitive as the counts of users, authors and tweets increase. (2) the large number of parameters it introduces would make the model more vulnerable to overfitting. (3) b_{ijm} is not explicitly correlated to the topical factors. Hence, it is difficult to capture influence patterns at the topic level. Our design of the $\boldsymbol{\alpha}$ variables can (1) enjoy the sparsity property of $\mathbf{R}$ to allow efficient model learning (will be discussed at the end of this section); (2) generate a moderate number of parameters and (3) naturally capture topical influence patterns.

To ensure the latent factors capture topics in text, we factorize the content matrix $\mathbf{W}$ as $W_{mn} \sim \mathrm{Poisson}(\sum_t \theta_{mt}\beta_{nt})$ [13].

Next, we discuss the design of priors on the latent variables. Firstly, recall α_{it} represents the probability that user i follows influential authors on topic t. The Beta distribution is a natural choice for priors of probabilities (e.g. as a prior for the probability parameter of Bernoulli distributions). In this work, we impose a balanced unimodal Beta prior on $\boldsymbol{\alpha}_i$'s, which means no bias is imposed on users' influence patterns. Secondly, we place Gamma priors on the latent variables for users, authors, tweets and words, since the Gamma distribution is conjugate with the Poisson distribution and can naturally govern nonnegative variables [6], [12], [13]. Moreover, in the microblogging environment, (1) an author is usually known to be influential on a few topics; (2) tweets are limited in length and only convey a few topics; (3) a specific word does not appear in many topics (we remove stop words and general words). The Gamma distribution is desirable here for encouraging sparse representations of $\mathbf{v}_j$'s, $\boldsymbol{\theta}_m$'s and $\boldsymbol{\beta}_n$'s. This can be achieved by placing a Gamma prior on these variables with shape parameter less than 1 [12]. Finally, we impose a hierarchical Gamma prior on $\mathbf{u}_i$'s to account for different activity degrees of users [12]. Details of IBPF are presented in the next subsection.

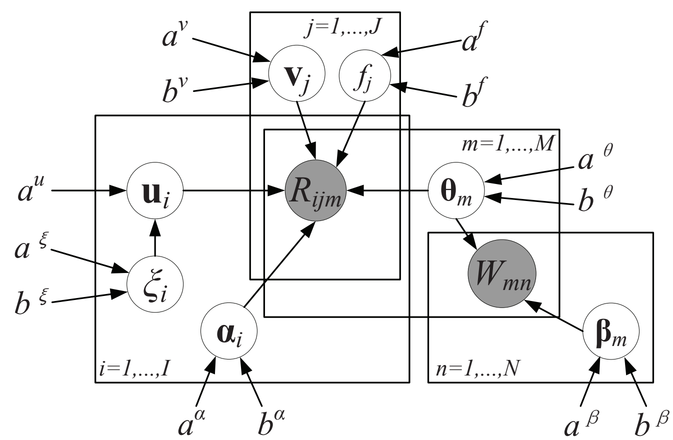


Fig. 1. The graphical model of IBPF.

3.3 Model Specification

The graphical model of IBPF is shown in Figure 1. Here we detail the model by the order of data generation and then discuss how IBPF can efficiently handle sparse implicit retweet data.

The latent factors of each $\mathbf{u}_i$ are generated according to the following hierarchical process [12]:

$$\xi_i \sim \mathrm{Gamma}(a^\xi, b^\xi)$$
$$u_{it} \sim \mathrm{Gamma}(a^u, \xi_i), \quad \forall t \in \{1, \ldots, T\}$$

where ξ_i is the rate parameter of the Gamma prior for $\mathbf{u}_i$, which is in turn generated by another Gamma prior. In this way, each user has a customized Gamma prior due to ξ_i. As a result of properties of the Gamma rate parameter, users with smaller ξ tend to have a latent preference vector with larger size. In other words, smaller ξ corresponds to those users who tend to retweet more actively and on more topics than other users. The influence pattern varibles of user i is generated by a Beta prior:

$$\alpha_{it} \sim \mathrm{Beta}(a^\alpha, b^\alpha), \quad \forall t \in \{1, \dots, T\}$$

where we set $a^\alpha = b^\alpha > 1$ to achieve a balanced unimodal distribution. The t-th latent factors of $\mathbf{v}_j$, $\boldsymbol{\theta}_m$ and $\boldsymbol{\beta}_n$ are generated as

$$\begin{aligned} v_{jt} &\sim \mathrm{Gamma}(a^v, b^v), \\ \theta_{mt} &\sim \mathrm{Gamma}(a^\theta, b^\theta), \\ \beta_{nt} &\sim \mathrm{Gamma}(a^\beta, b^\beta) \end{aligned}$$

With all the latent vectors in hand, we then can generate $\mathbf{R}$ and $\mathbf{W}$ as discussed in Section 3.2.

The whole generative process of IBPF is summarized as follows:

1) For each user i:
   a) Draw activity $\xi_i \sim \mathrm{Gamma}(a^\xi, b^\xi)$;
   b) For each topic t, draw topic preference $u_{it} \sim \mathrm{Gamma}(a^u, \xi_i)$ and author influence probability $\alpha_{it} \sim \mathrm{Beta}(a^\alpha, b^\alpha)$.
2) For each author j:
   a) Draw j's scale factor $f_j \sim \mathrm{Gamma}(a^f, b^f)$
   b) For each topic t, draw j's influence on t: $v_{jt} \sim \mathrm{Gamma}(a^v, b^v)$.
3) For each tweet m and each topic t, draw m's intensity on t: $\theta_{mt} \sim \mathrm{Gamma}(a^\theta, b^\theta)$.
4) For each word n and each topic t, draw n's intensity on t: $\beta_{nt} \sim \mathrm{Gamma}(a^\beta, b^\beta)$.
5) For each combination of m and n, draw word count $\mathbf{W}$ as $W_{mn} \sim \mathrm{Poisson}(\sum_t \theta_{mt}\beta_{nt})$.
6) For each combination of i, j and m, draw the binary retweet behavior $R_{ijm} \sim \mathrm{Poisson}(f_j[\sum_t \alpha_{it} u_{it} v_{jt} \theta_{mt} + \sum_t (1-\alpha_{it}) u_{it} \theta_{mt}])$.

In the generation of variables for authors, we also draw a scale factor f_j for each author j. When generating R_{ijm}, f_j provides a normalization effect. The reason for introducing normalization terms is that the number of retweets received by an author usually follows a power law distribution. Without normalization, the most popular authors could dominate the learning process, squeezing the latent topical space into a few dimensions. To explain in detail, a popular author j could easily obtain extremely high intensity values on some topical dimensions in $\mathbf{v}_j$ due to large numbers of retweets from j on those topics. Therefore, the model would tend to use those topical factors to explain every retweets from j, dragging tweets (and also related users) on minority topics to those dominating topics (recall that we impose a sparse Gamma prior on tweet latent factors). Our preliminary experiments also confirmed this phenomenon. The scale factor f_j could help alleviate this domination issue in $\mathbf{v}_j$ by jointly explaining the retweets from j. As shown by Step 6 of the generation process, f_j is engaged in explaining all the retweets from j. Therefore, more popular authors would have higher f, which could help keep the values of all the dimensions in $\mathbf{v}$ at a moderate level. On the other hand, f_j does not affect the topical assignment of each R_{ijm}, since every latent dimension t is multiplied by the same author scale factor f_j.

Note that in the last two steps we iterate through all the elements in $\mathbf{R}$ and $\mathbf{W}$. Nevertheless, the inference of IBPF is efficient since we only need to process nonzero elements. This can be reflected from the data likelihood under IBPF which is an important part in posterior inference. Take the retweet tensor $\mathbf{R}$ as an example. The likelihood of R_{ijm} is

$$\begin{aligned} & p(R_{ijm} | \mathbf{u}_i, \boldsymbol{\alpha}_i, \mathbf{v}_j, \boldsymbol{\theta}_m) \\ &= \left(\sum_t f_j u_{it} \theta_{mt} (\alpha_{it} v_{jt} + 1 - \alpha_{it}) \right)^{R_{ijm}} \\ &\quad \times \exp\left[- \sum_t f_j u_{it} \theta_{mt} (\alpha_{it} v_{jt} + 1 - \alpha_{it}) \right] \end{aligned}$$

Here we omit the factorial normalizer since R_{ijm} can only take 0 or 1. The log likelihood of the whole tensor is

$$\begin{aligned} & \log p(\mathbf{R} | \mathbf{u}, \mathbf{y}, \mathbf{v}, \boldsymbol{\theta}) \\ &= \sum_{R_{ijm}=1} R_{ijm} \log \left(\sum_t f_j u_{it} \theta_{mt} (\alpha_{it} v_{jt} + 1 - \alpha_{it}) \right) \\ &\quad - \sum_t \Bigg((\sum_i u_{it} \alpha_{it}) (\sum_j f_j v_{jt}) (\sum_m \theta_{mt}) \\ &\qquad + (\sum_i u_{it} (1 - \alpha_{it})) (\sum_j f_j) (\sum_m \theta_{mt}) \Bigg) \end{aligned}$$

Hence, only the nonzero part of $\mathbf{R}$ affects data likelihood, i.e. the first line on the right hand side of $=$. Since the latent factors are nonnegative, the second term can be calculated efficiently, and indicates that the zero part of $\mathbf{R}$ only contributes to making latent factors sparse.

4 Inference

Let $\Omega = \{\xi_i, \mathbf{u}_i, \boldsymbol{\alpha}_i, \mathbf{v}_j, \boldsymbol{\theta}_m, \boldsymbol{\beta}_n, f_j | \forall i, j, m, n\}$ be the set of all latent variables and $\Delta = \{a^\xi, b^\xi, a^u, a^\alpha, b^\alpha, a^v, b^v, a^\theta, b^\theta, a^\beta, b^\beta, a^f, b^f\}$ be the set of all hyperparameters. The objective of Bayesian inference is to learn the posterior distribution $p(\Omega | \mathbf{R}, \mathbf{W}, \Delta)$. However, the posterior is very complex and computationally intractable. We resort to mean-field variational inference [24] to learn the model.

4.1 Variational Inference

The basic idea of variational inference is to approximate the posterior distribution by a variational distribution $q(\Omega)$ where the latent variables are governed by free parameters. The marginal log likelihood of the observed data can then be rewritten as

$$\begin{aligned} \log p(\mathbf{R}, \mathbf{W} | \Delta) &= \int q(\Omega) \log p(\mathbf{R}, \mathbf{W} | \Delta) d\Omega \\ &= \int q(\Omega) \log \frac{p(\mathbf{R}, \mathbf{W}, \Omega | \Delta) q(\Omega)}{p(\Omega | \mathbf{R}, \mathbf{W}, \Delta) q(\Omega)} d\Omega \\ &= \int q(\Omega) \log \frac{p(\mathbf{R}, \mathbf{W}, \Omega | \Delta)}{q(\Omega)} d\Omega \\ &\quad + \int q(\Omega) \log \frac{q(\Omega)}{p(\Omega | \mathbf{R}, \mathbf{W}, \Delta)} d\Omega \end{aligned} \tag{3}$$

The second term of (3) is the Kullback-Leibler (KL) divergence between $q(\Omega)$ and the posterior which is our objective to minimize. Since the marginal data likelihood is a constant, minimizing the KL-divergence is equivalent to maximizing the first term of (3), a lower bound for the marginal data log likelihood:

$$\mathcal{L}(q) \triangleq \mathbb{E}_q[\log p(\mathbf{R}, \mathbf{W}, \Omega|\Delta)] - \mathbb{E}_q[\log q(\Omega)] \tag{4}$$

where $\mathbb{E}_q$ denotes taking expectation under $q(\Omega)$.

To specify $q(\Omega)$, we first note that the Poisson generative distributions of W_{mn} and R_{ijm} involve a summation in their rate parameters. This makes the inference difficult. Following [6], [12], [13], we exploit the superposition property of Poisson random variables to add auxiliary latent variables for facilitating the inference. Specifically, for each R_{ijm} we define two length-T vectors of latent variables $\mathbf{S}^{\alpha}_{ijm}$ and $\mathbf{S}^{\bar{\alpha}}_{ijm}$, where $S^{\alpha}_{ijmt} \sim \text{Poisson}(\alpha_{it}u_{it}f_j v_{jt}\theta_{mt})$, $S^{\bar{\alpha}}_{ijmt} \sim \text{Poisson}((1-\alpha_{it})u_{it}f_j\theta_{mt})$, and $R_{ijm} = \sum_t (S^{\alpha}_{ijmt} + S^{\bar{\alpha}}_{ijmt})$; for each W_{mn}, we add one length-T vector of latent variables $\mathbf{Z}_{mn}$ and let $Z_{mnt} \sim \text{Poisson}(\theta_{mt}\beta_{nt})$ and $W_{mn} = \sum_t Z_{mnt}$. Note these auxiliary variables are random only for nonzero R_{ijm} and W_{mn}. Hence, we do not need to concern about auxiliary variables for zeros in $\mathbf{R}$ and $\mathbf{W}$. The computational cost brought by the auxiliary variables is still linear in the number of positive training evidences (i.e., nonzero elements in $\mathbf{R}$ and $\mathbf{W}$).

After adding these auxiliary variables, the IBPF model becomes conditionally conjugate, except for the influence pattern variables $\boldsymbol{\alpha}_i$'s. The Poisson likelihood based on $\boldsymbol{\alpha}$ is not conjugate with the corresponding prior which is a Beta distribution. Hence, we adopt the Laplace variational technique proposed in [49] to optimize $\boldsymbol{\alpha}$. The key idea is using Laplace approximation to approximate the general optimal form of $\log q(\boldsymbol{\omega}_k)$, the logarithm of the variational distribution for the k-th latent variable. The optimal form is [3]

$$\log q(\boldsymbol{\omega}_k) \propto \mathbb{E}_{q(\neg k)}[\log p(X, \Omega)] \triangleq g(\boldsymbol{\omega}_k) \tag{5}$$

where X denotes the observed data and $\mathbb{E}_{q(\neg k)}[\cdot]$ means taking expectation with respect to all the q distributions except $q(\boldsymbol{\omega}_k)$. $g(\boldsymbol{\omega}_k)$ is approximated by a second-order Taylor expansion around its maximum:

$$g(\boldsymbol{\omega}_k) \approx g(\boldsymbol{\omega}^*_k) + \frac{1}{2}(\boldsymbol{\omega}_k - \boldsymbol{\omega}^*_k)^T \nabla^2 g(\boldsymbol{\omega}^*_k)(\boldsymbol{\omega}_k - \boldsymbol{\omega}^*_k) \tag{6}$$

where $\boldsymbol{\omega}^*_k$ denotes the value that maximizes $g(\boldsymbol{\omega}_k)$. Since the approximation is around $\boldsymbol{\omega}^*_k$, the first-order term vanishes as $\nabla g(\boldsymbol{\omega}^*_k) = 0$. According to Eqs. (5) and (6), $q(\boldsymbol{\omega}_k)$ then becomes

$$\begin{aligned} q(\boldsymbol{\omega}_k) &\propto \exp(g(\boldsymbol{\omega}_k)) \\ &\approx \exp\left[g(\boldsymbol{\omega}^*_k) + \frac{1}{2}(\boldsymbol{\omega}_k - \boldsymbol{\omega}^*_k)^T \nabla^2 g(\boldsymbol{\omega}^*_k)(\boldsymbol{\omega}_k - \boldsymbol{\omega}^*_k)\right] \end{aligned} \tag{7}$$

Eq. (7) suggests that we can approximate $q(\boldsymbol{\omega}_k)$ by a normal distribution:

$$q(\boldsymbol{\omega}_k) \approx \mathcal{N}(\boldsymbol{\omega}^*_k, -[\nabla^2 g(\boldsymbol{\omega}^*_k)]^{-1}) \tag{8}$$

It also shows how we update the mean and variance variables by $\boldsymbol{\omega}^*_k$ and $g(\cdot)$. The normal form of $q(\boldsymbol{\omega}_k)$ arises naturally in the derivation.

The other latent variables are all conjugate, so their factorized mean-field variational distributions take the same form as their *complete conditionals*, the conditional distributions given all the other variables [3]. It can be easily verified that the complete conditionals for $\text{concat}(\mathbf{S}^{\alpha}_{ijm}, \mathbf{S}^{\bar{\alpha}}_{ijm})$ and $\mathbf{Z}_{mn}$ are multinomial distributions and those for the other conjugate variables are all Gamma distributions. In summary, we define $q(\Omega)$ as

$$\begin{aligned} q(\Omega) = &\prod_i \text{Gamma}(\xi_i|\tilde{a}^{\xi}_i, \tilde{b}^{\xi}_i) \prod_{i,t} \text{Gamma}(u_{it}|\tilde{a}^{u}_{it}, \tilde{b}^{u}_{it}) \\ &\prod_{i,t} \mathcal{N}(\alpha_{it}|\mu_{it}, \sigma^2_{it}) \prod_{j,t} \text{Gamma}(v_{jt}|\tilde{a}^{v}_{jt}, \tilde{b}^{v}_{jt}) \\ &\prod_{m,t} \text{Gamma}(\theta_{mt}|\tilde{a}^{\theta}_{mt}, \tilde{b}^{\theta}_{mt}) \prod_{n,t} \text{Gamma}(\beta_{nt}|\tilde{a}^{\beta}_{nt}, \tilde{b}^{\beta}_{nt}) \\ &\prod_{i,j,m} \text{Mult}(\text{concat}(\mathbf{S}^{\alpha}_{ijm}, \mathbf{S}^{\bar{\alpha}}_{ijm})|R_{ijm}, \boldsymbol{\phi}^S_{ijm}) \\ &\prod_{m,n} \text{Mult}(\mathbf{Z}_{mn}|W_{mn}, \boldsymbol{\phi}^Z_{mn}) \prod_j \text{Gamma}(f_j|\tilde{a}^f_j, \tilde{b}^f_j) \end{aligned}$$

where $\tilde{a}$'s, $\tilde{b}$'s are variational shape and rate parameters for the corresponding variables, μ_{it}, σ^2_{it} are mean and variance of the normal variational distribution of α_{it}, and $\boldsymbol{\phi}^S_{ijm}/\boldsymbol{\phi}^Z_{mn}$ are variational multinomial parameters residing on 2T/T simplex.

Replacing $q(\Omega)$ in Eq. (4) with the above definition, we get a computable $\mathcal{L}(q)$. Next, we show how to optimize $\mathcal{L}(q)$ with respect to all the variational parameters. The influence pattern variable α_{it} can be updated according to Eq. (8). However, in our case we need to specify $g(\cdot)$ and how to maximize it. In particular, $g(\alpha_{it})$ is derived as follows

$$\begin{aligned} g(\alpha_{it}) =&\mathbb{E}_{q(\neg\alpha_{it})}[\log p(\mathbf{R}, \mathbf{W}, \Omega|\Delta)] \\ =&\log p(\alpha_{it}|a^{\alpha}, b^{\alpha}) + \sum_{j,m} \mathbb{E}_{q(\neg\alpha_{it})}[\log p(S^{\alpha}_{ijmt}, S^{\bar{\alpha}}_{ijmt} \\ &|u_{it}, \alpha_{it}, v_{jt}, \theta_{mt}, f_j)] + const \\ =&(a^{\alpha} - 1 + \sum_{j,m} R_{ijm}\phi^S_{ijmt}) \log \alpha_{it} + (b^{\alpha} - 1 \\ &+ \sum_{j,m} R_{ijm}\phi^S_{ijm(t+T)}) \log(1 - \alpha_{it}) + \alpha_{it} \frac{\tilde{a}^u_{it}}{\tilde{b}^u_{it}} \\ &\times (\sum_m \frac{\tilde{a}^{\theta}_{mt}}{\tilde{b}^{\theta}_{mt}})(\sum_j \frac{\tilde{a}^f_j}{\tilde{b}^f_j}(1 - \frac{\tilde{a}^v_{jt}}{\tilde{b}^v_{jt}})) + const \\ \triangleq& c_1 \log \alpha_{it} + c_2 \log(1 - \alpha_{it}) + c_3 \alpha_{it} + const \end{aligned} \tag{9}$$

Since $p(\mathbf{R}, \mathbf{W}, \Omega|\Delta)$ can be factorized according to the dependency graph in Figure 1, $\mathbb{E}_{q(\neg\alpha_{it})}[\log p(\mathbf{R}, \mathbf{W}, \Omega|\Delta)]$ is actually a summation of the expectations of the logarithms of those factorized probability terms. The first two terms after the second equal sign in the above derivation are those in $\mathbb{E}_{q(\neg\alpha_{it})}[\log p(\mathbf{R}, \mathbf{W}, \Omega|\Delta)]$ related to α_{it}, and *const* represents terms that do not involve α_{it}. For clarity, we use c_1, c_2 and c_3 to denote the coefficients of $\log \alpha_{it}$, $\log(1-\alpha_{it})$ and α_{it}, respectively. To maximize $g(\alpha_{it})$, we simply differentiate it with respect to α_{it} and set the derivative to 0:

$$c_3\alpha^2_{it} + (c_1 + c_2 - c_3)\alpha_{it} - c_1 = 0 \tag{10}$$

If $c_3 = 0$, the solution is simply $\alpha^*_{it} = c_1/(c_1 + c_2)$. Recall that we place a balanced unimodal Beta prior on α_{it}, which means $a^{\alpha} > 1$ and $b^{\alpha} > 1$. Hence, it is guaranteed that $c_1 > 0, c_2 > 0$, and consequently we obtain a valid α^*_{it} in

$[0,1]$. When $c_3 \neq 0$, Eq. (10) is a typical quadratic equation which has two solutions. In this case we need to prove that a valid solution exists. The following proposition shows that Eq. (10) must have a solution in $(0,1)$ and provides a guide for selecting the valid solution.

Proposition 1. *If we set $a^\alpha > 1$ and $b^\alpha > 1$, the following solution of (10) always lies within $(0,1)$:*

$$s \triangleq \frac{\sqrt{(c_1+c_2-c_3)^2+4c_1c_3}-(c_1+c_2-c_3)}{2c_3} \tag{11}$$

Proof. Since $a^\alpha > 1, b^\alpha > 1$ and $\sum_{j,m} R_{ijm}\phi^S_{ijmt}$ and $\sum_{j,m} R_{ijm}\phi^S_{ijm(t+T)}$ are always nonnegative, we have $c_1 > 0, c_2 > 0$. c_3 can be either positive or negative (the case of $c_3 = 0$ is already discussed). We analyze the two cases separately.

When $c_3 > 0$, we know $c_1c_3 > 0$ and $c_2c_3 > 0$. we can derive the lower bound of s:

$$s > \frac{|c_1+c_2-c_3|-(c_1+c_2-c_3)}{2c_3}$$

If $c_1+c_2-c_3 \geq 0$, this fraction is equal to 0; Otherwise, $c_1+c_2-c_3 < 0$, which leads to $(c_1+c_2)/c_3 < 1$. The fraction becomes $1-(c_1+c_2)/c_3 > 0$. Therefore, we can safely conclude $s > 0$. The upper bound of s can be obtained as follows

$$\begin{aligned} s =& \frac{\sqrt{(c_1+c_2+c_3)^2-4c_2c_3}-(c_1+c_2-c_3)}{2c_3} \\ <& \frac{(c_1+c_2+c_3)-(c_1+c_2-c_3)}{2c_3} = 1 \end{aligned}$$

When $c_3 < 0$, we have $c_1c_3 < 0$ and $c_2c_3 < 0$. Note that the denominator of s is negative in this case. The lower bound derivation becomes

$$s > \frac{(c_1+c_2-c_3)-(c_1+c_2-c_3)}{2c_3} = 0$$

Its upper bound is

$$\begin{aligned} s =& \frac{\sqrt{(c_1+c_2+c_3)^2-4c_2c_3}-(c_1+c_2-c_3)}{2c_3} \\ <& \frac{|c_1+c_2+c_3|-(c_1+c_2-c_3)}{2c_3} \end{aligned}$$

If $c_1+c_2+c_3 < 0$, we have $[-(c_1+c_2+c_3)-(c_1+c_2-c_3)]/2c_3 = (c_1+c_2)/(-c_3) < 1$. When $c_1+c_2+c_3 \geq 0$, the last fraction is equal to 1. Hence, s is upper bounded by 1. This completes the proof. □

By Proposition 1, we update $\mu_{it} = s$. σ^2_{it} is updated by $-(g''(\alpha^*_{it}))^{-1}$ as follows

$$\sigma^2_{it} = -(g''(\alpha^*_{it}))^{-1} = \frac{1}{\frac{c_1}{s^2}+\frac{c_2}{(1-s)^2}} \tag{12}$$

For the other variational parameters, we take the gradient of $\mathcal{L}(q)$ with respect to them and set to zero to obtain the coordinate ascent update rules. The update rules are listed as follows

$$\tilde{a}^\xi_i = a^\xi + Ta^u, \qquad \tilde{b}^\xi_i = b^\xi + \sum_t \frac{\tilde{a}^u_{it}}{\tilde{b}^u_{it}} \tag{13}$$

$$\begin{aligned} \tilde{a}^u_{it} &= a^u + \sum_{j,m} R_{ijm}(\phi^S_{ijmt} + \phi^S_{ijm(t+T)}) \\ \tilde{b}^u_{it} &= \frac{\tilde{a}^\xi_i}{\tilde{b}^\xi_i} + \sum_{j,m} \frac{\tilde{a}^f_j}{\tilde{b}^f_j}\frac{\tilde{a}^\theta_{mt}}{\tilde{b}^\theta_{mt}}(1+\mu_{it}\frac{\tilde{a}^v_{jt}}{\tilde{b}^v_{jt}} - \mu_{it}) \end{aligned} \tag{14}$$

$$\tilde{a}^v_{jt} = a^v + \sum_{i,m} R_{ijm}\phi^S_{ijmt}, \quad \tilde{b}^v_{jt} = b^v + \sum_{i,m} \mu_{it}\frac{\tilde{a}^u_{it}}{\tilde{b}^u_{it}}\frac{\tilde{a}^f_j}{\tilde{b}^f_j}\frac{\tilde{a}^\theta_{mt}}{\tilde{b}^\theta_{mt}} \tag{15}$$

$$\begin{aligned} \tilde{a}^f_j &= a^f + \sum_{i,m} R_{ijm} \\ \tilde{b}^f_j &= b^f + \sum_{i,m,t} \frac{\tilde{a}^u_{it}}{\tilde{b}^u_{it}}\frac{\tilde{a}^\theta_{mt}}{\tilde{b}^\theta_{mt}}(1+\mu_{it}\frac{\tilde{a}^v_{jt}}{\tilde{b}^v_{jt}} - \mu_{it}) \end{aligned} \tag{16}$$

$$\begin{aligned} \tilde{a}^\theta_{mt} &= a^\theta + \sum_{i,j} R_{ijm}(\phi^S_{ijmt} + \phi^S_{ijm(t+T)}) + \sum_n W_{mn}\phi^Z_{mnt} \\ \tilde{b}^\theta_{mt} &= b^\theta + \sum_{i,j} \frac{\tilde{a}^f_j}{\tilde{b}^f_j}\frac{\tilde{a}^u_{it}}{\tilde{b}^u_{it}}(1+\mu_{it}\frac{\tilde{a}^v_{jt}}{\tilde{b}^v_{jt}} - \mu_{it}) + \sum_n \frac{\tilde{a}^\beta_{nt}}{\tilde{b}^\beta_{nt}} \end{aligned} \tag{17}$$

$$\tilde{a}^\beta_{nt} = a^\beta + \sum_m W_{mn}\phi^Z_{mnt}, \quad \tilde{b}^\beta_{nt} = b^\beta + \sum_m \frac{\tilde{a}^\theta_{mt}}{\tilde{b}^\theta_{mt}} \tag{18}$$

$$\phi^S_{ijmt} \propto \begin{cases} \exp\{\mathbb{E}_q[\log \alpha_{it}] + \Psi(\tilde{a}^u_{it}) \\ -\log \tilde{b}^u_{it} + \Psi(\tilde{a}^v_{jt}) - \log \tilde{b}^v_{jt} & t \leq T \\ +\Psi(\tilde{a}^\theta_{mt}) - \log \tilde{b}^\theta_{mt}\} \\ \exp\{\mathbb{E}_q[\log(1-\alpha_{it})] + \Psi(\tilde{a}^u_{it}) \\ -\log \tilde{b}^u_{it} + \Psi(\tilde{a}^y_{it}) - \log \tilde{b}^y_{it} & T < t \leq 2T \\ +\Psi(\tilde{a}^\theta_{mt}) - \log \tilde{b}^\theta_{mt}\} \end{cases} \tag{19}$$

$$\phi^Z_{mnt} \propto \exp\{\Psi(\tilde{a}^\theta_{mt}) - \log \tilde{b}^\theta_{mt} + \Psi(\tilde{a}^\beta_{nt}) - \log \tilde{b}^\beta_{nt}\} \tag{20}$$

where $\Psi(\cdot)$ denotes the Digamma function. These update equations can also be derived by taking $\mathbb{E}_q[\cdot]$ of the conditional parameters of the corresponding latent variables' complete conditionals [13], since the complete conditionals are in exponential family and we let $q(\Omega)$ take the same forms. Note in Eq. (19) we need to compute $\mathbb{E}_q[\log \alpha_{it}]$ and $\mathbb{E}_q[\log(1-\alpha_{it})]$. These expectations are ill-defined since logarithm is undefined for negative values. However, we can still approximate them by Taylor expansion:

$$\begin{aligned} \mathbb{E}_q[\log \alpha_{it}] &\approx \mathbb{E}_q[\log \mu_{it} + \frac{1}{\mu_{it}}(\alpha_{it}-\mu_{it}) - \frac{1}{2\mu^2_{it}}(\alpha_{it}-\mu_{it})^2] \\ &= \log \mu_{it} - \frac{1}{2\mu^2_{it}}\sigma^2_{it} \end{aligned}$$

$$\mathbb{E}_q[\log(1-\alpha_{it})] \approx \log(1-\mu_{it}) - \frac{1}{2(1-\mu_{it})^2}\sigma^2_{it}$$

These approximations are reasonable as long as the corresponding normal variational distribution is concentrated near the mean, especially when the mean is near 0 or 1. We find in experiments that for ambiguous cases (i.e., μ_{it} near 0.5), σ^2_{it} is typically near 0.1; when μ_{it} is near 0 or 1, σ^2_{it} is on the order of 10^{-6}, meaning that our confidence is high due to the observed retweets. Hence, the approximations are acceptable in practice. The optimization algorithm simply updates each variational parameter in turn until convergence.

4.2 Computational Complexity

The space cost of IBPF is mainly due to $\mathbf{R}$, $\mathbf{W}$, and the variational parameters of $\mathbf{u}$, $\boldsymbol{\alpha}$, $\mathbf{v}$, $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. $\mathbf{R}$ and $\mathbf{W}$ can be efficiently stored in a sparse representation. The space cost of the variational parameters is $O(2T(2I+J)+T(M+N+2))$. As can be seen from the update equations, the other parameters only rely on the accumulation of $\boldsymbol{\phi}^S$ and $\boldsymbol{\phi}^Z$. Hence, we do not need to store them. They can be computed on the fly. There are some intermediate variables to store, e.g. c_1, c_2 and c_3 in Eq. (10). However, they have similar space complexity as the main parameters. Since T is usually not large, we can see the space cost of IBPF is efficient.

Regarding time complexity, the update costs of $\boldsymbol{\phi}^S$, $\boldsymbol{\phi}^Z$ and all the variational shape parameters depend linearly on the number of nonzero elements in $\mathbf{R}$ and/or $\mathbf{W}$. The calculation for variational rate parameters is also efficient: 1) different tweets/words share the same T rate parameters; (2) the nested summations in the update equations can be computed efficiently, e.g. $\sum_{i,m}(\tilde{a}^u_{it}/\tilde{b}^u_{it})(\tilde{a}^\theta_{mt}/\tilde{b}^\theta_{mt}) = (\sum_i \tilde{a}^u_{it}/\tilde{b}^u_{it})(\sum_m \tilde{a}^\theta_{mt}/\tilde{b}^\theta_{mt})$. The cost for rate parameters is $O(x^I IT + x^J JT + x^M MT + x^N NT + x^T T)$ with proper constant weights x^I, x^J, x^M, x^N and x^T. For $\boldsymbol{\alpha}$, we need to obtain c_1, c_2 and c_3 in Eq. (10) and then calculate μ_{it} and σ^2_{it} according to Eq. (11) and Eq. (12). It is easy to see that the time cost of calculating c_1 and c_2 is similar to that for Gamma shape parameters, i.e. linear in the number of nonzero elements in $\mathbf{R}$. The time cost for c_3 is similar to that for Gamma rate parameters. Hence, the time cost of updating $\boldsymbol{\alpha}$ only contributes to the constant weights of the above complexity results. In experiment, we also implement a GPU version of IBPF and report the running time test results.

4.3 Inference and Model Usage

Once the model is learned, $\mathbf{v}$, $\mathbf{u}$, $\boldsymbol{\alpha}$ characterize author influence, user preference and user influence pattern, respectively. We can use the obtained $q(\Omega)$ to approximate the posterior and to infer the latent variables by taking expectations under $q(\Omega)$. For example, we can get $\mathbb{E}_q[v_{jt}] = \tilde{a}^v_{jt}/\tilde{b}^v_{jt}$, $\mathbb{E}_q[u_{it}] = \tilde{a}^u_{it}/\tilde{b}^u_{it}$, $\mathbb{E}_q[\theta_{mt}] = \tilde{a}^\theta_{mt}/\tilde{b}^\theta_{mt}$ and $\mathbb{E}_q[\alpha_{it}] = \mu_{it}$. The inferred results can be used for further analysis or predictive tasks. In particular, we could assess the retweet tendency of any combination (i, j, m) as

$$\tilde{R}_{ijm} = \mathbb{E}_q[\sum_t \alpha_{it} u_{it} v_{jt} \theta_{mt} + \sum_t (1-\alpha_{it}) u_{it} \theta_{mt}] \quad (21)$$

$\tilde{R}_{ijm}$ can be either used directly or incorporated into existing methods as a feature for retweet prediction or tweet recommendation.

5 EXPERIMENTS

In this section, we empirically evaluate IBPF on two microblogging datasets collected from Sina Weibo[1] and Twitter[2], respectively. In the first part, we show and analyze the learning results of IBPF. Then we investigate applying the learning results on retweet prediction/tweet recommendation.

TABLE 2
Statistics of the datasets.

Dataset	# users	# authors	# tweets	# retweets
Weibo	1,164,556	446,014	215,037	16,566,756
Twitter	41,164	132,167	428,861	504,620

5.1 Datasets & Experimental Setup

Two publicly available microblogging datasets are used in the exepiments.

Weibo[3] The Weibo dataset is extracted from the dataset used in [55]. Since [55] only concerned who retweeted a tweet but not whom it was retweeted from, there is no explicit author information in the dataset. However, the author information could be inferred from the retweet list field and social relationships. Hence, we process the dataset to get 16,566,756 retweets with inferred author information. Then we extract related users, authors and tweets to form our Weibo dataset.

Twitter[4] The Twitter dataset [54] contains 11,408,918 retweets, but the tweets are written in various languages and there is no detailed information about authors (only names). We filter the tweets to only keep those that are written in English. The filtered dataset contains 504,620 retweets. We summarize the statistics of the two datasets in Table 2.

Experimental setup We run IBPF on a server with i7-4930K CPU, 64GB memory and GeForce GTX TITAN GPU. Following [12], [13], we set all the Gamma prior hyperparameters at 0.3. We set $a^\alpha = b^\alpha = 2$ for $\boldsymbol{\alpha}$ to construct a balanced unimodal Beta prior as aforementioned. The variational parameters of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ are initialized by the results of BTM [52] and kept fixed in the early stage of the learning process. We also initialize the variational parameters of $\mathbf{u}$ and $\mathbf{v}$ by aggregating the tweet BTM topic vectors of the user's/author's involved retweets. The number of topics T can be determined by nonparametric topic models [19], which is out of scope of this work. For the sake of simplicity, we set $T = 50$.

5.2 Analysis of Learning Results

We apply IBPF on the whole datasets to mine specific influence patterns. To provide a general view of the influence patterns, we summarize the learned $\boldsymbol{\alpha}$ for the users. Specifically, for each topic t we count the numbers of users with $\alpha_{it} > 0.6$ and $\alpha_{it} < 0.4$ respectively, and calculate their proportion. The topics are sorted in descending order of this proportion. Intuitively, proportion values significantly higher than 1 means users tend to follow topical authorities (high author influence), while proportion values much lower than 1 indicates that users are more likely to follow their own interests (high content influence). Note we do not

1. http://weibo.com
2. http://twitter.com
3. https://cn.aminer.org/#Weibo-Net-Tweet
4. https://cn.aminer.org/#Twitter-Dynamic-Net

TABLE 3
Typical topics exhibiting author influence/content influence in Weibo (translated).

Topics for author influence			Topics for content influence		
Topic	Top words	Top influential authors	Topic	Top words	Top influenced users
Politics	China, we, people, society, democracy, nation, cultural revolution, corruption	1813080181, 1182389073, 1189591617	Healthy bedtime	health, weak up, sleep, body, midnight, self, night, everyday, women	1462948420, 1903335275, 1840682035
Sale campaigns	friend, fan, chance, draw, campaign, follow, lottery, free	2179589753, 2607667467, 1670645393	Self-cultivation	generous, contempt, life, suffer losses, voice, calm, status, distress	1890679643, 1890617574, 2664508573
Music	video, music, concert, high definition, song, premiere, album, handclap	1642591402, 1830442653, 1920061532	Fun	video, wonderful, funny, share, titter, cute, like, awesome	1795128591, 1850098094, 1678626237

consider $0.4 \leq \alpha_{it} \leq 0.6$ since that case often means lack of evidence due to data sparsity.

Typical topics with high/low proportion values are shown in Tables 3 and 4, for Weibo and Twitter, respectively. We also show top authors (users) names for each topic with high values of $\mathbb{E}_q[v_{jt}]$ (low values of $\mathbb{E}_q[\alpha_{it}]$)[5]. The results are intuitive. High author influence is often connected with debatable topics such as *Politics* and professional topics such as *Sale campaigns* and *Music*, while high content influence is often conveyed by topics such as *Fun*, *Philosophy* and *Charity*. Users are more easily influenced by the content in these topics. We find for daily life topics, content influence also tends to dominate. This is because the retweet becomes a social means in this context between normal users who are friends. Normal users are hard to obtain high topical influence. The *Healthy bedtime* topic is also intuitive since it would not require much professional prestige. Regarding topics with high author influence, the learned top influencers are indeed influential accounts on the topics[6]. We also investigate the behaviors of the top active users in influential topics and find they retweet from different non-influential authors on the corresponding topics.

5.3 Retweet Prediction/Tweet Recommendation

As aforementioned in Section 2.2, retweet prediction and tweet recommendation are closely related. The general goal is to predict user preference on tweets by retweeting behaviors. Unlike research for general recommendation which is focused on algorithms [16], [17], [44], [45], state-of-the-art methods in this sub-field [11], [20], [36], [56] have proposed various features for the two tasks. Here we treat the score computed in Eq. (21) as a new feature and investigate whether it can further improve the performance of the two tasks.

Experimental settings & Baselines Since there is no author information in Twitter dataset, we only use Weibo dataset in this experiment. We adopt an evaluation methodology similar to that of [20]. Firstly, we filter users with retweets less than 15. This results in 108,707 target users. For each target user, we sort his incoming tweets from followees in ascending order of time and for each positive instance sample 4 negative instances that are nearest to it by time. The intuition is that non-retweets surrounding retweets are more likely to be real negative instances (i.e. viewed but not retweeted). Unlike [56], we do not construct a balanced dataset since that would make the recommendation task easier. The filtered incoming tweet list is split into 5 segments with equal size. We train the model on one segment and test it on the next. The averaged performance is reported. In order to fairly compare the effectiveness of different features, we employ factorization machines (FM) [37] as the common model and feed different sets of features into it. We employ features from stat-of-the-art works as baselines: (1) **F-CoFM** [20]: this is the set of features used by the CoFM method for tweet recommendation, including features from content, meta data, relationships etc.; (2) **F-FFM** [11]: features used by the feature-aware factorization model (FFM) including node and edge features for each <user, author, tweet>; (3) **F-Diff**: diffusion based features proposed in [36]; (4) **InfLoc**: the influence locality feature proposed in [55], [56]. For retweet prediction, we use precision, recall, F1 and accuracy as the evaluation metrics [55]. Normalized Discount Cumulative Gain (NDCG) and Mean Average Precision (MAP) are used to measure recommendation performance. We define Precision as the number of correctly recommended tweets (i.e. those which are retweeted by the user) divided by the number of all recommended tweets. Average Precision (AP) is the average of precision scores after each correctly recommended tweet:

$$\mathrm{AP} = \frac{\sum_i \mathrm{Precision@}i \times \mathrm{corr}_i}{\text{No. of correctly recommended tweets}} \tag{22}$$

where $\mathrm{Precision@}i$ is the precision at ranking position i and $\mathrm{corr}_i = 1$ if the tweet at position i is correctly recommended, otherwise $\mathrm{corr}_i = 0$. MAP is the mean of average precision scores over all target users. NDCG at position k is defined as

$$\mathrm{NDCG@k} = Z_k \sum_{i=1}^{k} (2^{r_i} - 1)/\log_2(i+1) \tag{23}$$

where r_i is the relevance rating of tweet at rank i. In our case, r_i is 1 if the corresponding tweet is retweeted by the user and 0 otherwise. Z_n is chosen so that the perfect ranking has a NDCG value of 1.

The results are shown in Tables 5 and 6, for prediction and recommendation respectively. Here "All" means we combine all baseline features and IBPF and "All baselines" is a combination of all the 4 baselines. We can see although IBPF alone cannot beat baselines with a lot of features, when it is integrated with features from all the baselines,

5. Since user names are in Chinese, for Weibo we show user IDs and one can find them by typing "weibo.com/ID" in browers.

6. The last two accounts for the topic Politics in Weibo were removed. They are Zhiqiang Ren and Chengpeng Li who can be found on Wikipedia

TABLE 4
Typical topics exhibiting author influence/content influence in Twitter.

Topics for author influence			Topics for content influence		
Topic	Top words	Top influential authors	Topic	Top words	Top influenced users
Political news	pass, state, nation, bill, group, rule, public, major	CBSRadioNews, cnnbrk, BreakingNews	Life philosophy	heart, open, eye, close, side, speak, touch, soul	citizenparker, cashbandicoot_, PutrisdEffendi
TV shows	show, live, watch, video, perform, season, catch, air	GMA, TheEllenShow, todayshow	Daily life	omg, sleep, bed, kick, ha, hahaha, jump, ring	anai699, CuzImAlwaysON, billuko
Music	play, song, listen, hear, sound, sing, amaz, danc	chrisbrown, ladygaga, gagadaily	Charity	money, pay, save, problem, million, spend, extra, bank	shesFEARLESS13, Cathyliscious, kellywilliams4

TABLE 5
Performance comparison for retweet prediction.

Features	Precision	Recall	F1	Accuracy
F-CoFM	0.790	0.607	0.685	0.888
F-FFM	0.724	0.587	0.648	0.825
F-Diff	0.542	0.422	0.475	0.800
InfLoc	0.604	0.558	0.580	0.749
IBPF	0.872	0.252	0.391	0.851
All baselines	0.804	0.676	0.734	0.901
All	**0.888**	**0.708**	**0.788**	**0.923**

TABLE 6
Performance comparison for tweet recommendation.

Features	NDCG@1	NDCG@3	NDCG@5	MAP
F-CoFM	0.904	0.844	0.724	0.586
F-FFM	0.641	0.531	0.459	0.409
F-Diff	0.746	0.643	0.591	0.517
InfLoc	0.472	0.441	0.405	0.374
IBPF	0.786	0.708	0.695	0.482
All baselines	0.911	0.862	0.742	0.596
All	**0.975**	**0.933**	**0.927**	**0.631**

the performance can be boosted significantly. This indicates that IBPF can provide additional new knowledge for the two tasks based on existing knowledge in the state-of-art features. The reason could be that no baseline feature exploits the ternary relations of users, authors and tweets (only pairwise relations are considered). Note the InfLoc feature does not achieve good performance since we removed retweets for which we cannot infer their authors (Section 5.1). A more complete structural diffusion view is required by InfLoc.

5.4 Time Complexity

Here we investigate the time costs of the algorithms in terms of one iteration of parameter updating. As mentioned in Section 4.2, we implement both CPU version and GPU version of IBPF where the GPU version uses GPU to concurrently compute ϕ^S, ϕ^Z and update variational parameters. We vary the number of nonzero elements in $\mathbf{R}$ and $\mathbf{W}$ and fix all the other quantities in a simulation dataset based on the Weibo dataset. The results are shown in Figure 2. The axes are in log-scale for clarity. We can see that the time cost of the CPU version grows linearly with the number of sparse relations, which conforms with the analysis in Section 4.2, while the GPU version is very efficient, e.g. 19.8s for processing 50M retweets plus 50M tweet-word relations.

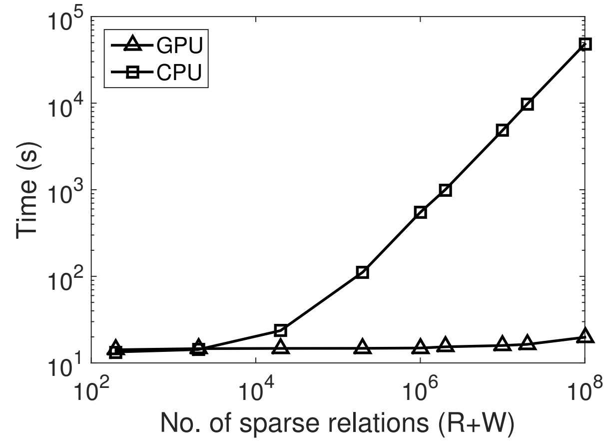


Fig. 2. Time costs as the number of sparse relations in $\mathbf{R}$ and $\mathbf{W}$ vary.

6 Conclusions & Future Work

We propose a new influence mining method for microblogging data, named Influence Beta-Poisson Factorization (IBPF). IBPF can automatically discern author influence and content influence in an optimal sense from retweet evidences. We develop efficient variational optimization algorithms for IBPF. The effectiveness and efficiency of IBPF are tested on two large-scale public microblogging datasets.

There are several possible extensions that can be made to IBPF. First, we will investigate how to adapt IBPF to the online setting, to cope with the evolution nature of microblogging data. This could be addressed by developing stochastic variational inference [19] algorithms for IBPF. We will also try to integrate nonparametric topic modeling into IBPF seamlessly to capture the number of topical factors automatically.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61672409, 61522206, 61373118, 61876144), the Major Basic Research Project of Shaanxi Province (Grant No. 2017ZDJC-31), Shaanxi Province Science Fund for Distinguished Young Scholars (Grant No. 2018JC-016) and the Science and Technology Plan Program in Shaanxi Province of China (Grant No. 2017KJXX-80). The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

[1] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *SIGKDD*, pages 7–15. ACM, 2008.

[2] B. Bi, Y. Tian, Y. Sismanis, A. Balmin, and J. Cho. Scalable topic-specific influence analysis on microblogs. In *WSDM*, pages 513–522. ACM, 2014.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, Secaucus, NJ, USA, 2006.

[4] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *2010 43rd Hawaii International Conference on System Sciences (HICSS)*, pages 1–10. IEEE, 2010.

[5] A. E. Cano, S. Mazumdar, and F. Ciravegna. Social influence analysis in microblogging platforms–a topic-sensitive based approach. *Semantic Web*, 5(5):357–372, 2014.

[6] A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009, 2009.

[7] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *SIGIR*, pages 661–670. ACM, 2012.

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

[9] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *SIGKDD*, pages 160–168. ACM, 2008.

[10] V. R. Embar, I. Bhattacharya, V. Pandit, and R. Vaculin. Online topic-based social influence analysis for the wimbledon championships. In *SIGKDD*, pages 1759–1768. ACM, 2015.

[11] W. Feng and J. Wang. Retweet or not?: personalized tweet re-ranking. In *WSDM*, pages 577–586. ACM, 2013.

[12] P. Gopalan, J. M. Hofman, and D. M. Blei. Scalable recommendation with poisson factorization. *arXiv preprint arXiv:1311.1704*, 2013.

[13] P. K. Gopalan, L. Charlin, and D. Blei. Content-based recommendations with poisson factorization. In *NIPS*, pages 3176–3184, 2014.

[14] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, pages 241–250. ACM, 2010.

[15] B. Hajian and T. White. Modelling influence in a social network: Metrics and evaluation. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 497–500. IEEE, 2011.

[16] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua. Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912*, 2018.

[17] X. He, J. Tang, X. Du, R. Hong, T. Ren, and T.-S. Chua. Fast matrix factorization with non-uniform weights on missing data. *arXiv preprint arXiv:1811.04411*, 2018.

[18] T.-A. Hoang and E.-P. Lim. Retweeting: An act of viral users, susceptible users, or viral topics? In *SDM*, pages 569–577. SIAM, 2013.

[19] M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[20] L. Hong, A. S. Doumith, and B. D. Davison. Co-factorization machines: modeling user interests and predicting individual decisions in twitter. In *WSDM*, pages 557–566. ACM, 2013.

[21] J. Hu, Y. Fang, and A. Godavarthy. Topical authority propagation on microblogs. In *CIKM*, pages 1901–1904. ACM, 2013.

[22] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. Ieee, 2008.

[23] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang. Social contextual recommendation. In *CIKM*, pages 45–54. ACM, 2012.

[24] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[25] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *SIGKDD*, pages 137–146. ACM, 2003.

[26] Y. Kim and K. Shim. Twitobi: A recommendation system for twitter using probabilistic modeling. In *ICDM*, pages 340–349. IEEE, 2011.

[27] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[28] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, pages 591–600. ACM, 2010.

[29] D. Li, Z. Luo, Y. Ding, J. Tang, G. Guo-Zheng Sun, X. Dai, J. Du, J. Zhang, and S. Kong. User-level microblogging recommendation incorporating social influence. *Journal of the Association for Information Science and Technology*, 68(3):553–568, 2017.

[30] J. Li, C. Liu, J. X. Yu, Y. Chen, T. Sellis, and J. S. Culpepper. Personalized influential topic search via social network summarization. *IEEE TKDE*, 28(7):1820–1834, 2016.

[31] J. Li, X. Wang, K. Deng, X. Yang, T. Sellis, and J. X. Yu. Most influential community search over large social networks. In *ICDE*, pages 871–882. IEEE, 2017.

[32] L. Liu, J. Tang, J. Han, and S. Yang. Learning influence from heterogeneous social networks. *Data Mining and Knowledge Discovery*, 25(3):511–544, 2012.

[33] X. Liu, H. Shen, F. Ma, and W. Liang. Topical influential user analysis with relationship strength estimation in twitter. In *2014 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1012–1019. IEEE, 2014.

[34] S. A. Macskassy and M. Michelson. Why do people retweet? anti-homophily wins the day! In *ICWSM*, pages 209–216, 2011.

[35] A. Pal and S. Counts. Identifying topical authorities in microblogs. In *WSDM*, pages 45–54. ACM, 2011.

[36] Y. Pan, F. Cong, K. Chen, and Y. Yu. Diffusion-aware personalized social update recommendation. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 69–76. ACM, 2013.

[37] S. Rendle. Factorization machines with libfm. *ACM TIST*, 3(3):57, 2012.

[38] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *SIGKDD*, pages 61–70. ACM, 2002.

[39] F. Riquelme and P. González-Cantergiani. Measuring user influence on twitter: A survey. *Information Processing & Management*, 52(5):949–975, 2016.

[40] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, pages 695–704. ACM, 2011.

[41] A. Schein, J. Paisley, D. M. Blei, and H. Wallach. Inferring polyadic events with poisson tensor factorization. In *Proceedings of the NIPS 2014 Workshop on Networks: From Graphs to Rich Data*, 2014.

[42] A. Silva, S. Guimarães, W. Meira Jr, and M. Zaki. Profilerank: finding relevant content and influential users based on information diffusion. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, page 2. ACM, 2013.

[43] 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 *2010 IEEE second international conference on Social computing (SocialCom)*, pages 177–184. IEEE, 2010.

[44] J. Tang, X. He, X. Du, F. Yuan, Q. Tian, and T.-S. Chua. Adversarial training towards robust multimedia recommender system. *arXiv preprint arXiv:1809.07062*, 2018.

[45] J. Tang, X. Shu, G.-J. Qi, Z. Li, M. Wang, S. Yan, and R. Jain. Tri-clustered tensor completion for social-aware image tag refinement. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1662–1674, 2017.

[46] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *SIGKDD*, pages 807–816. ACM, 2009.

[47] M. Turcotte, J. Moore, N. Heard, and A. McPhall. Poisson factorization for peer-based anomaly detection. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 208–210. IEEE, 2016.

[48] I. Uysal and W. B. Croft. User oriented tweet ranking: a filtering approach to microblogs. In *CIKM*, pages 2261–2264. ACM, 2011.

[49] C. Wang and D. M. Blei. Variational inference in nonconjugate models. *Journal of Machine Learning Research*, 14(Apr):1005–1031, 2013.

[50] M. Wang, X. Zheng, Y. Yang, and K. Zhang. Collaborative filtering with social exposure: A modular approach to social recommendation. In *AAAI*, 2018.

[51] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM*, pages 261–270. ACM, 2010.

[52] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *WWW*, pages 1445–1456. ACM, 2013.

[53] Z. Yang, J. Guo, K. Cai, J. Tang, J. Li, L. Zhang, and Z. Su. Understanding retweeting behaviors in social networks. In *CIKM*, pages 1633–1636. ACM, 2010.
[54] J. Zhang, Z. Fang, W. Chen, and J. Tang. Diffusion of "following" links in microblogging networks. *IEEE TKDE*, 27(8):2093–2106, 2015.
[55] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, pages 2761–2767, 2013.
[56] J. Zhang, J. Tang, J. Li, Y. Liu, and C. Xing. Who influenced you? predicting retweet via social influence locality. *ACM TKDD*, 9(3):25, 2015.
[57] Q. Zhang, Y. Gong, Y. Guo, and X. Huang. Retweet behavior prediction using hierarchical dirichlet process. In *AAAI*, pages 403–409, 2015.
[58] Q. Zhang, Y. Gong, J. Wu, H. Huang, and X. Huang. Retweet prediction with attention-based deep neural network. In *CIKM*, pages 75–84. ACM, 2016.
[59] W. Zhang and J. Wang. A collective bayesian poisson factorization model for cold-start local event recommendation. In *SIGKDD*, pages 1455–1464. ACM, 2015.
[60] 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 *SIGKDD*, pages 1513–1522. ACM, 2015.

Wei Zhao received the B.S., M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2002, 2005 and 2015, respectively. He is currently an associated professor in the School of Computer Science and Technology at Xidian University. His research direction is pattern recognition and intelligent systems, with specific interests in attributed graph mining and search, machine learning, signal processing and precision guiding technology.

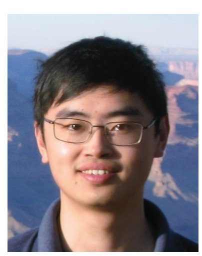
Ziyu Guan received the B.S. and Ph.D. degrees in Computer Science from Zhejiang University, Hangzhou China, in 2004 and 2010, respectively. He had worked as a research scientist in the University of California at Santa Barbara from 2010 to 2012, and as a professor in the School of Information and Technology of Northwest University, China from 2012 to 2018. He is currently a professor with the School of Computer Science and Technology, Xidian University. His research interests include attributed graph mining and search, machine learning, expertise modeling and retrieval, and recommender systems.

Yuhui Huang received the B.S. degree from Xidian University, Xi'an China, in 2015 and received the M.S. degree from Zhejiang University, Hangzhou China, in 2015. He is currently an algorithm engineer in FABU Technology. His research interests include machine learning and data mining.

Tingting Xi received her B.S. degree from Nankai University, Tianjin China, in 2017. She is currently a graduate student in Xidian University, Xi'an China. Her research interests include social network mining and recommender systems.

Huan Sun is an assistant professor in the Department of Computer Science and Engineering at the Ohio State University (OSU). Prior to that, She spent half a year as a visiting scientist at the University of Washington, and received the Ph.D. degree in Computer Science from University of California, Santa Barbara (2015) and the B.S. degree in EEIS from the University of Science and Technology of China (2010). Her current research interests lie in data mining, and machine learning with emphasis on text mining and understanding, network analysis, and human behavior understanding.

Zhiheng Wang received the B.S. degree in mechatronic engineering from the Beijing Institute of Technology, Beijing China, in 2004, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing China, in 2009. He is currently a Professor with the Department of Computer Science and Technology, Henan Polytechnic University, Jiaozuo China. His research interests include computer vision, pattern recognition, and image processing.



Xiaofei He received the BS degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision.