Information Systems 42 (2014) 59–77

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation

Younghoon Kim, Kyuseok Shim *

Department of Electrical and Computer Engineering, Seoul National University, Kwanak P.O. Box 34, Seoul 151-600, Republic of Korea

ARTICLE INFO

Article history:
Received 7 November 2013
Received in revised form
24 November 2013
Accepted 26 November 2013
Recommended by: D. Shasha
Available online 4 December 2013

Keywords:
Twitter
Recommendation system
Probabilistic model
Collaborative filtering
LDA
Matrix factorization

ABSTRACT

Twitter provides search services to help people find users to follow by recommending popular users or the friends of their friends. However, these services neither offer the most relevant users to follow nor provide a way to find the most interesting tweet messages for each user. Recently, collaborative filtering techniques for recommendations based on friend relationships in social networks have been widely investigated. However, since such techniques do not work well when friend relationships are not sufficient, we need to take advantage of as much other information as possible to improve the performance of recommendations.

In this paper, we propose *TWILITE*, a recommendation system for Twitter using probabilistic modeling based on latent Dirichlet allocation which recommends top-K users to follow and top-K tweets to read for a user. Our model can capture the realistic process of posting tweet messages by generalizing an LDA model as well as the process of connecting to friends by utilizing matrix factorization. We next develop an inference algorithm based on the variational EM algorithm for learning model parameters. Based on the estimated model parameters, we also present effective personalized recommendation algorithms to find the users to follow as well as the interesting tweet messages to read. The performance study with real-life data sets confirms the effectiveness of the proposed model and the accuracy of our personalized recommendations.

1. Introduction

Twitter has emerged recently as a new medium in spotlight for communication. Twitter offers a unique mechanism of information diffusion by allowing each user to receive all messages (called tweets) from those whom he follows. We refer to those who follow a user as *followers* and refer to those whom a user follows as *followees*.

Generally, users would like to follow other users who post interesting messages. However, assisting users to find new people to follow is not a simple task. Twitter itself provides a service to help people find the users to follow by recommending popular users or the friends of their friends. However, these services do not offer the most relevant users to follow for a user. Furthermore, recommending the most interesting tweet messages for a user will be very useful, but Twitter does not provide this functionality.

There has been much work done on developing new approaches for recommendation systems over the last decade. The interest in the area still remains high because personalized recommendations have many practical applications (e.g., shopping cart recommendations of Amazon, MovieLens and AdaptiveInfo.com). Moreover, with the recent fast growth of Social Network Services (SNSs) such as Twitter and Facebook, the problem of recommending other users or other users' posted messages to a user with common interests has received a lot of attention.

* Corresponding author. Tel.: +82 2 880 7269.
E-mail address: shim@kdd.snu.ac.kr (K. Shim).

0306-4379/$ - see front matter
http://dx.doi.org/10.1016/j.is.2013.11.003

Many recommendation systems such as [7,20,24] take the approach of collaborative filtering which recommends friends or articles based on the pattern of the other users who have connected to a similar set of friends in the past. However, if a user is new to the system and thus has little friends, we need to take advantage of not only the friend relationships but also the content of their messages for a better recommendation.

Due to the distinctive feature of Twitter which allows users to post their Twitter messages and to follow others, we have more opportunities for better recommendations than the other traditional applications. For example, even for the users with few Twitter messages, we can provide better recommendations using the Twitter messages of their followees. To take advantage of such a distinctive feature of Twitter, we previously proposed *TWITOBI*, a recommendation system for Twitter using probabilistic modeling for collaborative filtering in [22].

The probabilistic model used in *TWITOBI* was a generalization of the probabilistic latent semantic indexing (PLSI) in [16]. The model assumes that topics are selected by following not only the user's preference to the topics but also the preference of the user's followees to the topics. An Expectation–Maximization (EM) algorithm was next developed to learn the parameters of our model by maximizing the log-likelihood of expectation. The performance study with Twitter data showed the effectiveness and scalability of our algorithms for Twitter. However, our previous PLSI model does not capture the generative process of establishing friend relationships in Twitter. Furthermore, it is known in [6,29] that PLSI suffers from the overfitting problem compared to the latent Dirichlet allocation [5].

In this paper, we propose a new probabilistic model which can capture the generative process of posting tweet messages as well as establishing friend relationships. It is reported that users in social network have a strong tendency to follow other users who have many common friends [2]. Since the matrix factorization is one of the most successful models for describing the connections between users by assuming that a user has a high chance to be a friend with the ones who have many common friends [27,40,42], we exploit the friend relationships between users using matrix factorization to enhance the accuracy of recommendations for Twitter. In our model, we seamlessly combine the probabilistic model based on LDA to model the process of posting tweet messages and the collaborative filtering based on matrix factorization to capture the process of establishing friend relationships. We next develop an inference algorithm that utilizes the variational EM algorithm and the matrix factorization.

To learn model parameters for LDA models, Gibbs sampling [13] or variational EM algorithms has been widely used. Since it is simpler to draw an inference algorithm using Gibbs sampling rather than a variational EM algorithm, many probabilistic models based on LDA utilize Gibbs sampling. However, Gibbs sampling methods usually require many iterations to find a solution resulting in too slow convergence. Furthermore, since Gibbs sampling does not guarantee the convergence of log-likelihood, it is hard to determine when to stop the iterations [19]. On the other hand, deriving a variational EM algorithm is sometimes difficult depending on models but it converges faster and guarantees the convergence. In this paper, we thus develop a variational EM algorithm for our LDA model.

The contributions of this paper are as follows:

- We propose a novel probabilistic generative model which is suitable for representing the activities in Twitter. Our model represents a realistic process of posting tweet messages by exploiting the LDA model as well as the process of connecting to friends by utilizing matrix factorization.
- We develop an inference algorithm which utilizes a variational EM algorithm with matrix factorization to learn the posterior distributions and model parameters of our LDA model.
- We also provide the ranking algorithms for recommending top-K followees or top-K tweets to a user using the estimated posterior distributions and model parameters.
- By performance study with Twitter data, we show the effectiveness of our model as well as the accuracy of the top-K followee and tweet recommendation algorithms for Twitter.

The rest of this paper is organized as follows. After discussing related work in Section 2, we provide our problem formulations as well as the preliminary works in Section 3. We next propose a probabilistic generative model in Section 4 and develop an inference algorithm using a variational EM algorithm with matrix factorization for our model in Section 5. Section 6 presents how to utilize our model parameters for recommendations. Finally, the performance study is given in Section 7 and we summarize the paper in Section 8.

2. Related work

We first discuss the model-based collaborative filtering algorithms [15,17,28,31,32,36,41] and next describe the recent works on recommendations for social networks [27,40,42].

Model-based algorithms build their models to describe the behaviors of users using training data and utilize the trained models to predict the users' preference on the items unseen in the training data. Examples of model-based approaches include the probabilistic models [5,22], random walk models [11,41], latent factor models [18,30] and combined ones [36] of probabilistic and latent factor models.

A naive recommendation method which can be used for Twitter is to recommend users or contents based on similarity. In [9,14], TF-IDF weighting is used for recommending users to follow and tweet messages to read based on the cosine similarity between their friends and tweet messages. Even if those algorithms consider the tweet messages as well as followers and followees, a simple use of TF-IDF weighting makes the algorithms to suffer the quality of recommendations.

More complex probabilistic models were later proposed in various recommendation applications. In [31], the Markov process was used to model the purchasing process of market basket data. In [17,23,28,32], recommendations utilizing the PLSI in [16] were also investigated. However, these algorithms build models based on the relationships between users and items, and thus the followee and follower relationships between users in Twitter cannot be simply modeled by these modeling techniques. In [8,26], LDA in [5] is generalized to model both latent topics and hidden communities of users. However, these techniques do not consider the relationships between users and thus, we cannot use those techniques directly for Twitter data.

To utilize the relationships between users in social networks, recommendation algorithms based on random walk models were proposed in [11,41]. In [41], after building a graph of items to represent the similarities between items, they compute personalized PageRanks [15] to recommend the top-*K* ranked items. In [11], the personalized PageRank was generalized for bipartite graphs. However, these algorithms do not use the contents such as tweet messages. Furthermore, since these algorithms do not consider the preference weights to friends in the random walk, in the social networks such as Twitter where users usually follow many other users, the performance of recommendations can be largely degraded [22].

Recently, many latent factor models utilizing matrix factorization methods have been proposed [27,40,42] for recommendation systems in social networks. All of such methods estimate the latent factor vectors of users where each factor measures the extent to which the user is connected with the other users according to its corresponding factor. Since latent factor models can take advantage of not only the explicit relationships between users but also the implicit relationships between two users without any connection, they are effective for recommending users in social networks where people form different clusters according to their preferences. However, since they use friend relationships only without the contents generated by users, the performance suffers when a user does not have enough number of friends yet.

In [36], a system called CTR was proposed to recommend scientific papers to each user in CiteULike [10]. CiteULike is a specialized search engine to retrieve the technical papers and it allows people to share their favorite papers by voting thumbs up for the papers. To make recommendations by utilizing both user's votes and the content of papers, they combine matrix factorization [30] and LDA model [5] together, and showed that CTR outperforms the recommendations based on either matrix factorization or LDA model only [36]. Since the model utilizes the voting rates on the contents produced by users while we use the followee relationships in Twitter instead, it is hard to use the model in [36] for recommendations in Twitter.

In [22], we proposed a recommendation algorithm based on a probabilistic model, called TWITOBI, which generalizes PLSI in [16] to consider the tweet messages as well as the friend relationships. The model assumes that topics are selected by following not only the user's preference to the topics but also the preference of user's followees to the topics. However, the PLSI model does not include the generative process of establishing friend relationships in Twitter. Furthermore, it is known in [6,29] that PLSI suffers from the overfitting problem compared to LDA [5]. In this paper, we thus extend the model presented in [22] such that the new model utilizes LDA to model the process of posting messages and uses a matrix factorization technique to capture the process of establishing the friend relationships.

User	Followees	Tweets
a_1	a_2, a_3, a_4	Harry, Potter, Movie, Wallstreet, EmWatson
a_2	a_1, a_4, b_2, b_3	Harry, Potter, EmWatson, Economic
a_3	a_1, b_1	Harry, EmWatson, Movie
a_4	a_1, a_2	Movie, Harry
b_1	b_2, b_3, b_4, a_3	Harry, Financial, Crisis, IMF, Economic
b_2	b_1, b_3, b_4	EmWatson, Financial, Crisis, IMF, Economic
b_3	b_1, b_2, b_4	Wallstreet, Crisis, Economic
b_4	a_1, b_1, b_2	Harry, Potter, IMF, Financial
b_5	a_1, b_3	Wallstreet, IMF, Economic, Crisis

Fig. 1. An example of users and their tweets.

3. Preliminaries

In this section, we first provide the definitions to be used and formulate the problems of recommending followees and tweets for Twitter users. We next briefly introduce latent Dirichlet allocation [5] as well as matrix factorization [18]. Finally, we present TWITOBI, which is our probabilistic model proposed previously for Twitter in [22].

3.1. Problem formulation

Assume that we are given a tweet collection *D* and a set of users *U* with the relationships of followees and followers. We formulate the following two recommendation problems:

- *Top-K followee recommendation*: For a user *u*, choose top-*K* users who are not *u*'s followees and whom *u* would like to follow the most.
- *Top-K tweet recommendation*: For a user *u*, find top-*K* tweets not written by *u* in *D* which the user *u* would like the most.

We next present a running example that will be used to illustrate our proposed probabilistic model and the detailed steps of our inference algorithm in the following section.

Example 3.1. Consider the nine users with their followees and tweets shown in Fig. 1. The tweets written by every user are presented after eliminating stop-words. In our tweet message data, there exist two topics 'Financial crisis' and 'Harry Potter'.

Top-1 followee recommendation: Suppose that we recommend a new followee for a_3 whose followees are $\{a_1, b_1\}$. If we choose followee candidates by considering the 'friends of friends', the candidates are $\{a_2, a_4, b_2, b_3, b_4\}$ in which a_2

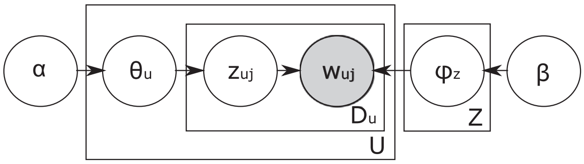

Fig. 2. The graphical representation of LDA.

and a_4 are the followees of a_1, and the rest of users are the followees of b_1. Note that we have two groups of $G_1=\{a_1, a_2, a_3, a_4\}$ and $G_2=\{b_1, b_2, b_3, b_4, b_5\}$ where those in each group have more connections to the users in the same group. With two friends a_1 and b_1 of a_3 from different groups, it is not clear to recommend either the friends of a_1 or the friends of b_1 because a_3 has the same number of users from both groups as his followees. However, considering that the major topic of the tweet by a_3 is 'Harry Potter', it is probably better to recommend a_2 or a_4 who is also interested in 'Harry Potter'.

Let us next consider the problem of recommending a new followee for b_4. If we choose the candidates by utilizing 'friends of friends' again, the candidates become $\{a_2, a_3, a_4, b_3\}$. Note that the tweet of b_4 is on both topics evenly. With our previous probabilistic model of *TWITOBI*, it happens to give a larger weight to the followee a_1 than b_1 and b_2 because a_1 is the only followee who is interested in 'Harry Potter'. Thus, in this case, *TWITOBI* would recommend one of the followees of a_1 to b_4. However, since he has much more connections to the users who are interested in 'Financial crisis', recommending b_3 to b_4 is probably better. To overcome such a drawback of *TWITOBI*, we propose another probabilistic model which utilizes the connections between users for recommendations.

Top-1 tweet recommendation: Suppose that we want to recommend a tweet to b_5 whose followees are a_1 and b_3 in different groups. If we recommend one of the tweets by his followees, it is hard to select the tweet message to recommend based on the friend relationships only. However, if we examine his tweet that is mainly about 'Financial crisis', we can find that it is better to recommend the tweet posted by b_3. On the other hand, consider the problem of recommending a tweet to b_4 whose tweet includes the two topics evenly. Since b_4 has more friends in the group of G_2, it seems to be better to recommend a tweet message written by b_1 or b_2. However, according to the same reason when we recommend the top-1 followee to b_4 in this example, *TWITOBI* recommends the tweet of a_1 to b_4. Our proposed probabilistic model to be presented later will recommend the tweet of b_3 who has more connections with b_4 instead. □

3.2. Latent Dirichlet allocation

LDA [5] is a generative probabilistic model that utilizes a set of *latent topics*, each of which represents the distribution over a fixed number of vocabularies to describe the generative process of documents. Let U and D_u denote the set of users and the bag of words generated by a user $u \in U$ respectively. Let V be the set of distinct words appearing in a bag of words D_u at least once for a user $u \in U$. We use Z to denote the set of latent topics where the number of topics is given as a parameter. In the generative process of LDA, each user u has his own preference over the topics represented by a probabilistic distribution $\vec{\theta_u}$, which is a multinomial distribution over Z, and each topic z also has a multinomial distribution over V, denoted by $\vec{\phi_z}$.

Fig. 2 shows the graphical representation of the LDA model which shows the following generative process. In the graphical representation, circles are used to denote random variables or model parameters in probabilistic models. Shaded circles are the variables or parameters that are observed in the data and white circles are the unobserved ones which will be estimated by inference algorithms. Arrows represent dependencies between variables. The generative process of LDA is presented below:

1. For each topic $z \in Z$,
   - Draw a multinomial distribution $\phi_z \sim Dir(\vec{\beta})$.
2. For each user $u \in U$,
   - Draw a multinomial distribution $\theta_u \sim Dir(\vec{\alpha})$.
   - For each word $w \in D_u$,
     (a) Draw a topic $z \sim Multinomial(\vec{\theta_u})$.
     (b) Draw a word $w \sim Multinomial(\vec{\phi_z})$.

The LDA model assumes that the multinomial distributions $\vec{\theta_u}$ and $\vec{\phi_z}$ are drawn from conjugate prior distributions, called Dirichlet distribution, whose parameters are given as $\vec{\alpha}$ and $\vec{\beta}$ respectively. Each word w in D_u is assumed to be selected by first drawing a topic z with following the topic preference distribution $\vec{\theta_u}$ and next choosing a word w from the corresponding distribution $\vec{\phi_z}$ of the selected topic z. According to the LDA model, the probability that a word w is generated by a user u is estimated as follows:

$$\int Dir(\theta_u;\alpha)\left(\sum_{z=1}^{|Z|}\theta_{uz}\phi_{zw}\right)d\theta_u.$$

3.3. Matrix factorization

Many popular latent factor models are based on matrix factorization [25]. In their basic form, matrix factorization computes the latent factor vectors of users and items (i.e., the object rated by the user), which are inferred from the item rating patterns of users. These methods have become popular in recent years due to good scalability with predictive accuracy. In the case of modeling the friend relationships in Twitter, users and items correspond to followers and followees respectively.

Matrix factorization models generate latent factor vectors in dimensionality d to represent the patterns to follow and to be followed. Then, each followee relationship between two users is defined by the inner product of their factor vectors. That is, each user u has a followee factor vector $\vec{x_u} \in \mathbb{R}^d$ and a follower factor vector $\vec{y_u} \in \mathbb{R}^d$. For a user u, each element of $\vec{x_u}$ represents the tendency to follow other users with respect to its corresponding factor. Similarly, each element of $\vec{y_u}$ measures the popularity of a user u to be followed by other users according to its corresponding factor. Thus, the dot product, $\overrightarrow{x_u}^T \cdot \overrightarrow{y_v}$, represents the extent to which a user u follows another user v in Twitter. This approximates u's interest on the user v, which is denoted by $\hat{r}_{uv}$, as follows:

$$\hat{r}_{uv} = \overrightarrow{x_u}^T \cdot \overrightarrow{y_v}$$

3.4. *TWITOBI: a generalized PLSI model for Twitter*

We proposed personalized recommendation algorithms for Twitter previously in [22], called *TWITOBI*, which utilizes a probabilistic model of the behavior of writing tweet messages by generalizing the probabilistic latent semantic indexing (PLSI) in [16]. The model assumes that the topics are not only selected by a user but also chosen under the influence of the followees of the user. Let U and D_u denote the set of users and the bag of words generated by a user u respectively. Let Z be the set of latent topics. F_u is a set of u's followees in U. Then, its generative model for tweet messages is as follows:

- For each user $u \in U$,
  - For each word $w \in D_u$,
    1. Decide whether to choose the topic of his own interest with the probability α or one of their followees' interests with the probability $(1-\alpha)$ where α is a given constant in the range of $[0, 1]$.
    2. If the user u decided to choose a topic with u's own interest,
       (a) Choose a topic $z \sim Multinomial(p(z|u))$.
    3. If the user u decided to choose a topic based on the interest of one of u's followees,
       (a) Select a friend $f \sim Multinomial(p(f|u))$.
       (b) Select a topic $z \sim Multinomial(p(z|f))$.
    4. Choose a word $w \sim Multinomial(p(w|z))$.

The model *TWITOBI* estimates the probability that a user u generates a word w in his tweet messages as the following equation:

$$\sum_{z=1}^{|Z|} \left[\alpha p(z|u) + (1-\alpha) \sum_{v \in F_u} p(z|v) p(w|z) \right]$$

4. TWILITE: our new probabilistic model for Twitter

In this section, we first describe the generative process of our model, called *TWILITE*, a model for *Twi*tter using probai*li*stic modeling based on la*te*nt Dirichlet allocation. Then, we present an inference algorithm to estimate the topic preference distributions of users to generate tweet messages as well as the latent factor vectors of users to establish friend relations.

4.1. *Our generative model*

Let U be the set of $|U|$ users which contains user IDs from 1 to $|U|$ (i.e., $U = \{1, \ldots, |U|\}$) and let D denote the collection of tweet messages represented by a set

a

Notation	Description
U	the set of user IDs in Twitter
V	the set of word IDs in which each word appears at least once in D
D_u	the bag-of-words which are the concatenation of tweet messages posted by u
D	the set of D_us for all $u \in U$
w_{uj}	the jth word in D_u such that $w_{uj} \in V$
$n(u,t)$	the number of occurrence of the tth word of V in D_u
F_u	the set of followees of u
F	the set of F_us for all $u \in U$
f_{uj}	the jth followee in F_u such that $f_{uj} \in U$

b

Notation	Description
$\vec{\theta_u}$	a multinomial distribution that represents the preference of u for the topics in Z
Θ	the set of $\vec{\theta_u}$s for all $u \in U$
$\vec{\varphi_k}$	a multinomial distribution that represents the relevance of words in V for the kth topic
Φ	the set of $\vec{\varphi_k}$s for all $k \in Z$
$\vec{\pi_u}$	a multinomial distribution that represents the preference of u for the followees in F_u
Π	and the set of $\vec{\pi_u}$s for all $u \in U$
s_{uj}	the random variable for denoting u's followee who influences u to select w_{uj}
z_{uj}	the random variable for representing a topic chosen to select w_{uj}
$\vec{x_u}$	the d-dimensional latent factor vectors of the patterns that u *follows* the other users
$\vec{y_u}$	the d-dimensional latent factor vectors of the patterns that u *is followed* by the other users

Fig. 3. Notations: (a) observed data and (b) random variables and model parameters.

$\{D_1, \ldots, D_{|U|}\}$ where $D_u \in D$ represents a virtual document generated by concatenating every tweet message posted by $u \in U$. We also let $V = \{1, \ldots, |V|\}$ be a set of word IDs in which each word appears at least once in D. The virtual document D_u can be represented by a sequence $\langle w_{u1}, w_{u2}, \ldots, w_{u|D_u|} \rangle$ of D_u where w_{uj} represents the jth word generated by the user $u \in U$ and should be one of the word IDs in the vocabulary V (i.e., $w_{uj} \in V$). Let $F = \{F_1, \ldots, F_{|U|}\}$ be the friend relationships of the users in U where $F_u \in F$ is the set of followees of $u \in U$. We represent F_u by $\{f_{u1}, f_{u2}, \ldots, f_{u|F_u|}\}$ where f_{uj} denotes the jth followee of u and is represented by one of user IDs in U (i.e., $f_{uj} \in U$). Since we consider the followees of a user u as those who can influence the user u when posting tweet messages and u may also be affected by himself, we include the user u in F_u as well. The notations to be used throughout the paper are summarized with brief descriptions in Fig. 3.

Random variables for the generative process of tweet messages: In our generative model, we assume that each word in a tweet message is chosen for one of latent topics as the model of LDA [5] does. Let Z denote the set of latent topics which are indexed with the integers from 1 to $|Z|$. However, we assume that the topics are not only selected by a user but also chosen under the influence of the friends of the user. Let $\mathcal{F}_u$ represent the set of u's friends by whom u is influenced when u selects a topic. Correspondingly, we introduce the following posterior probability distributions, which participate in our model as random variables:

- $\overrightarrow{\theta_u}$ is a multinomial probability distribution over Z and θ_{uk} denotes the probability with which the kth topic is selected by u. We denote Θ as the set of $\overrightarrow{\theta_u}$s for all $u \in U$. We assume that $\overrightarrow{\theta_u}$ follows a Dirichlet distribution with a given parameter $\overrightarrow{\alpha}$ (i.e., $Dir(\overrightarrow{\alpha})$).
- $\overrightarrow{\varphi_k}$ represents a multinomial distribution over the words in V and φ_{kt} is the probability with which the tth word in V is chosen for the kth topic. We use Φ to denote the set of $\overrightarrow{\varphi_k}$s for all $k \in Z$ and $\overrightarrow{\varphi_k}$ is assumed to follow a prior $Dir(\overrightarrow{\beta})$ where $\overrightarrow{\beta}$ is a given parameter.
- $\overrightarrow{\pi_u}$ denotes a multinomial distribution over the friends in $\mathcal{F}_u$ and $\pi_{u\ell}$ represents the probability that a user $u \in U$ selects the ℓth friend in $\mathcal{F}_u$ for selecting a topic. We refer to the set of $\overrightarrow{\pi_u}$s for every $u \in U$ as Π. We assume that $\overrightarrow{\pi_u}$ follows $Dir(\overrightarrow{\gamma_u})$ where the parameter $\overrightarrow{\gamma_u}$ is given for each user u.

We assume that the friends $\mathcal{F}_u$ for every user $u \in U$ is given as a prior to our generative model. However, since we cannot actually acquire such friend information, we will use the observed set of followees F_u as $\mathcal{F}_u$ in our inference algorithm. Furthermore, our model includes the following discrete random variables:

- s_{uj} is a random variable that represents one of u's followees who influence u to select the jth word in D_u. We use $\overrightarrow{s_u}$ and $\mathbb{S}$ to denote the sequence of s_{uj} for $j = 1, \ldots, |D_u|$ and the set of $\overrightarrow{s_u}$s for every $u \in U$ respectively.
- z_{uj} denotes a random variable that is the topic chosen by u for selecting the jth word in D_u. We use $\overrightarrow{z_u}$ and $\mathbb{Z}$ to represent the sequence of z_{uj} for $j = 1, \ldots, |D_u|$ and the set of $\overrightarrow{z_u}$s for all $u \in U$ respectively.

Model parameters for the generative process of friend relationships: The connections between users in Twitter are influenced by not only the preference to friends when they post tweet messages but also the patterns of users when they establish friend relationships with other users [2]. That is, the more they have common friends, the stronger tendency to follow other users the users in social networks have. Thus, exploiting both the topic preference in tweet messages and the patterns of friend relationships may enhance the accuracy of recommendations in Twitter.

For recommendations in social networks, collaborative filtering using matrix factorization has been widely studied [27,40,42] since collaborative filtering is suitable for utilizing not only explicit feedbacks such as friend relationships but also implicit feedbacks such as no-friend relationships which represents the case that a user does not follow another user for a pair of two users [18]. Thus, we utilize collaborative filtering by matrix factorization in our model and introduce the model parameters below:

- $\overrightarrow{x_u}$ is a d-dimensional latent factor vector in $\mathbb{R}^d$ representing the pattern that the user u *follows* other users. We use $\mathbb{X}$ to denote the set of $\overrightarrow{x_u}$s for every $u \in U$.
- $\overrightarrow{y_u}$ is a d-dimensional vector in $\mathbb{R}^d$ of the pattern that u *is followed* by other users. We use $\mathbb{Y}$ to denote the set of $\overrightarrow{y_u}$s for every $u \in U$.

In our model, we combine probabilistic topic modeling and collaborative filtering with matrix factorization by utilizing the following probability distribution which is referred to the probability that a user u selects v to follow among the users in U:

$$p(f_{uv} | \overrightarrow{\pi_u}, \overrightarrow{x_u}, \overrightarrow{y_v}) = \begin{cases} \frac{\pi_{uv}}{\zeta_u} e^{-(1 - \overrightarrow{x_u}^T \cdot \overrightarrow{y_v})^2} & \text{if } v \in F_u \\ \frac{\lambda}{\zeta_u} e^{-(0 - \overrightarrow{x_u}^T \cdot \overrightarrow{y_v})^2} & \text{if } v \notin F_u \end{cases} \tag{1}$$

where λ is a constant that is usually set to a small value close to 0 and ζ_u is a normalization function shown below:

$$\zeta_u = \sum_{v \in F_u} \pi_{uv} e^{-(1 - x_u^T \cdot y_v)^2} + \sum_{v \in U - F_u} \lambda e^{-(0 - x_u^T \cdot y_v)^2}$$

The probability function in Eq. (1) becomes higher for a user $v \in F_u$ as $\overrightarrow{x_u}^T \cdot \overrightarrow{y_v}$ is close to 1. In contrast, for a user $v \notin F_u$, the probability function becomes higher as $\overrightarrow{x_u}^T \cdot \overrightarrow{y_v}$ approaches to 0. Note that the probability function is also in proportion to π_{uv}. The reason is that a user v, who is preferred by the user u for selecting a topic of word when u writes tweet messages, has more chance to be followed by u.

In Fig. 3, we summarize the random variables and model parameters used in our model. We next present our probabilistic generative model of writing tweet messages and establishing friend relationships.

Generative process of TWILITE: The graphical representation of the generative process of *TWILITE* is presented in Fig. 4. According to the generative model, a user would write his tweets by repeatedly performing the following

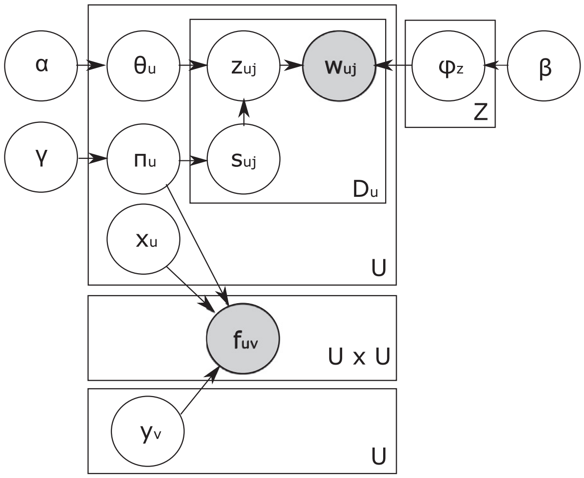

Fig. 4. A graphical model of *TWILITE*.

steps stochastically while sampling words and decide whether to follow the other users or not according to the probability in Eq. (1):

1. For each topic $z \in Z$,
   - Draw a multinomial distribution $\phi_z \sim Dir(\beta)$.
2. For each user $u \in U$,
   - Draw a multinomial distribution $\pi_u \sim Dir(\gamma_u)$.
   - Draw a multinomial distribution $\theta_u \sim Dir(\alpha)$.
   - For each word $w \in D_u$,
     (a) Draw a friend $i \sim Multinomial(\pi_u)$ among F_u.
     (b) If i is 0, draw a topic $z \sim Multinomial(\theta_u)$. Otherwise, draw a topic $z \sim Multinomial(\theta_{f_{ui}})$.
     (c) Draw a word $w \sim Multinomial(\phi_z)$.
3. For each user $u \in U$,
   - Select $|F_u|$ number of followees according to Eq. (1).

4.2. Likelihood of Twitter data

We now formulate the likelihood function for the observed data according to the proposed generative model. Since not only the observed data D and F are generated independently but also D_u and F_u of each user u are produced independently, the likelihood $\mathbb{L}$ of generating the observed data according to our generative model becomes $\mathbb{L} = p(D,F) = \prod_{u=1}^{|U|} p(D_u) \prod_{u=1}^{|U|} p(F_u)$. Furthermore, since each word in D_u and each followee relationship in F_u are generated independently, we obtain

$$\mathbb{L} = \left[\prod_{u=1}^{|U|} p(\Theta, \Phi, \Pi) \prod_{j=1}^{|D_u|} p(w_{uj}|\Theta, \Phi, \Pi) \right] \left[\prod_{u=1}^{|U|} \prod_{v=1}^{|U|} p(f_{uv}) \right]$$

Given three posterior distributions, $\overrightarrow{\theta_{s_{uj}}}$, $\overrightarrow{\varphi_{z_{uj}}}$ and $\overrightarrow{\pi_u}$, the joint distribution of a word w_{uj} in D_u, a topic z_{uj} for selecting the word and a user s_{uj} for selecting the topic is given by

$$\begin{aligned} p(w_{uj}, z_{uj}, s_{uj} | \overrightarrow{\theta_{s_{uj}}}, \overrightarrow{\varphi_{z_{uj}}}, \overrightarrow{\pi_u}) &= p(s_{uj}|\overrightarrow{\pi_u}) p(z_{uj}|s_{uj}, \overrightarrow{\theta_{s_{uj}}}) p(w_{uj}|z_{uj}, \overrightarrow{\varphi_{z_{uj}}}) \\ &= \pi_{u,s_{uj}} \theta_{s_{uj},z_{uj}} \varphi_{z_{uj},w_{uj}} \end{aligned} \quad (2)$$

By summing over z_{uj} and s_{uj}, we obtain the marginal distribution of w_{uj} below:

$$p(w_{uj}|\overrightarrow{\theta_{s_{uj}}}, \overrightarrow{\varphi_{z_{uj}}}, \overrightarrow{\pi_u}) = \sum_{s_{uj}=1}^{|F_u|} \sum_{z_{uj}=1}^{|Z|} \pi_{u,s_{uj}} \theta_{s_{uj},z_{uj}} \varphi_{z_{uj},w_{uj}} \quad (3)$$

To compute $p(f_{uv})$, we simply utilize the probability $p(f_{uv}|\overrightarrow{\pi_u}, \overrightarrow{x_u}, \overrightarrow{y_v})$ presented in Eq. (1). Then, by integrating over Θ, Φ and Π, we obtain the likelihood $\mathbb{L}$ as follows:

$$\begin{aligned} \mathbb{L} = \int_\Theta \int_\Phi \int_\Pi \Bigg[& \prod_{u=1}^{|U|} Dir(\overrightarrow{\theta_u}; \alpha) \prod_{z=1}^{|Z|} Dir(\overrightarrow{\varphi_z}; \beta) \prod_{u=1}^{|U|} Dir(\overrightarrow{\pi_u}; \gamma_u) \\ & \cdot \prod_{u=1}^{|U|} \prod_{j=1}^{|D_u|} \sum_{s_{uj}=1}^{|F_u|} \sum_{z_{uj}=1}^{|Z|} \pi_{u,s_{uj}} \theta_{s_{uj},z_{uj}} \varphi_{z_{uj},w_{uj}} \\ & \cdot \prod_{u=1}^{|U|} \prod_{v=1}^{|U|} p(f_{uv}|\overrightarrow{\pi_u}, \overrightarrow{x_u}, \overrightarrow{y_v}) \Bigg] d\Theta\, d\Phi\, d\Pi \end{aligned} \quad (4)$$

5. The variational EM algorithm

In our inference algorithm, given a collection of observations, we would like to compute the posterior distribution of the latent variables such as θ_u, π_u and φ_z when the likelihood function is maximized. However, calculating such distributions based on the likelihood function obtained in Eq. (4) is intractable similar to the likelihood functions of other probabilistic models such as [1,5] that utilize Dirichlet priors. In such cases, a variational EM algorithm [21] is one of the most popular methods for computing posterior distributions since it converges faster than the other inference algorithms such as Gibbs sampling and guarantees the convergence [19].

To apply a variational EM algorithm, we first derive the lower bound of log-likelihood in Eq. (4) by utilizing Jensen's inequality [39] as follows:

$$\begin{aligned} \log \mathbb{L} &= \log \int_{\Theta,\Phi,\Pi} \sum_{\mathbb{S},\mathbb{Z}} p(D, F, \mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)\, d\Theta\, d\Phi\, d\Pi \\ &= \log \int_{\Theta,\Phi,\Pi} \sum_{\mathbb{S},\mathbb{Z}} q(\mathbb{S}\mathbb{Z}, \Theta, \Phi, \Pi) \frac{p(D, F, \mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)}{q(\mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)}\, d\Theta\, d\Phi\, d\Pi \\ &\geq \int_{\Theta,\Phi,\Pi} \sum_{\mathbb{S},\mathbb{Z}} q(\mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi) \log \frac{p(D, F, \mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)}{q(\mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)}\, d\Theta\, d\Phi\, d\Pi \\ &= \mathbb{F}(q_{(\mathbb{S},\mathbb{Z},\Theta,\Phi,\Pi)}, \mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi) \end{aligned} \quad (5)$$

In the above, we let $\mathbb{F}(q_{(\mathbb{S},\mathbb{Z},\Theta,\Phi,\Pi)}, \mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)$ be a lower bound on $\log \mathbb{L}$. Note that $\mathbb{F}$ is a function of the free distribution $q(\mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)$ and the random variables $\{\mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi\}$. We perform the mean field approximation [5] to approximate the free distribution by factorizing $q(\mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi)$. In the case of our *TWILITE* model, a tractable mean field approximation can be obtained by

$$\begin{aligned} q(\mathbb{S}, \mathbb{Z}, \Theta, \Phi, \Pi) &= q(\mathbb{S}) q(\mathbb{Z}) q(\Theta) q(\Phi) q(\Pi) \\ &= \prod_{u=1}^{|U|} \prod_{j=1}^{|D_u|} \delta_{uj}(s_{uj}) \phi_{uj}(w_{uj}) \prod_{u=1}^{|U|} Dir(\theta_u; a_u) \\ & \prod_{z=1}^{|Z|} Dir(\varphi_z; b_z) \prod_{u=1}^{|U|} Dir(\pi_u; c_u) \end{aligned} \quad (6)$$

where δ_{uj} and ϕ_{uj} are the approximate multinomial distributions for the random variables s_{uj} and z_{uj} respectively. Furthermore, a_u, b_k and c_u are the approximate Dirichlet priors for $\overrightarrow{\pi_u}$, $\overrightarrow{\phi_k}$ and $\overrightarrow{\theta_u}$ respectively. By introducing these approximate model parameters, the lower bound $\mathbb{F}$ can be factorized into a tractable form. In order not to break the flow of reading, we provide the definition of the lower bound $\mathbb{F}$ with respect to the approximate parameters later in Appendix A.

We next maximize $\mathbb{F}$ with respect to the approximate parameters a, b, c, ϕ, δ and ρ. Let Σ denote the set of every model parameter (i.e., $\Sigma = \{\alpha, \beta, \gamma, \mathbb{X}, \mathbb{Y}\}$). To find a local optimum of the lower bound, we iterate between the steps of fitting the variational distribution q to approximate the posterior in E-step and maximizing the corresponding lower bound for the likelihood with respect to the parameters in M-step

$$\text{E-Step}: \; q^{(t+1)} \leftarrow \arg\max_q \mathbb{F}(q, \Sigma^{(t)}), \tag{7}$$

$$\text{M-Step}: \; \Sigma^{(t+1)} \leftarrow \arg\max_\Sigma \mathbb{F}(q^{(t+1)}, \Sigma). \tag{8}$$

Variational E-step: Tightening the lower bound with respect to approximate parameters is equivalent to minimizing the KL-divergence between the free distribution q and the posterior distributions Θ, Φ and Π [5,37]. Using the convexity of KL-divergence, we perform an iterative fixed-point method for the minimization. By computing the derivatives of the KL-divergence for each approximate parameter of ϕ, δ and ρ, we obtain the following equations for updates in each iteration:

$$\phi_{uj}(z) \propto \exp\left[\sum_{v=1}^{|U|}\sum_{\ell=1}^{|F_u|} I(f_{u\ell}=v)\delta_{uj}(\ell)(\Psi(a_{vz}) - \Psi(a_{v\bullet})) + (\Psi(b_{zw_{uj}}) - \Psi(b_{z\bullet}))\right] \tag{9}$$

$$\delta_{uj}(\ell) \propto \exp\left[\sum_{z=1}^{|Z|} \phi_{uj}(z)(\Psi(a_{vz}) - \Psi(a_{v\bullet})) + (\Psi(c_{u\ell}) - \Psi(c_{u\bullet}))\right] \tag{10}$$

$$\rho_{uv}(f) \propto \begin{cases} \exp\left[(\Psi(c_{uv}) - \Psi(c_{u\bullet})) + (1 - \overrightarrow{x_u}^T \cdot \overrightarrow{y_u})^2\right] & \text{if } f=1 \\ \exp\left[\log\lambda + (-\overrightarrow{x_u}^T \cdot \overrightarrow{y_u})^2\right] & \text{if } f=0 \end{cases} \tag{11}$$

where $a_{u\bullet}$, $b_{z\bullet}$ and $c_{u\bullet}$ denote $\sum_{z=1}^{|Z|} a_{uz}$, $\sum_{t=1}^{|V|} b_{zt}$ and $\sum_{j=1}^{|F_u|} c_{uj}$ respectively, and $\Psi(x)$ represents the first derivative of the log $\Gamma(x)$ function.

For the remaining approximate parameters a, b and c, we can also obtain the equations for updating the parameters as follows:

$$a_{uz} = \alpha_z + \sum_{i=1}^{|F_u|}\sum_{j=1}^{|D_u|} I(f_{ui}=u)\phi_{uj}(z)\delta_{uj}(i) + \sum_{v \in R_u}\sum_{i=1}^{|F_v|}\sum_{j=1}^{|D_u|} I(f_{vi}=u)\phi_{vj}(z)\delta_{vj}(i) \tag{12}$$

$$b_{zt} = \beta_t + \sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|} I(w_{uj}=t)\phi_{uj}(z) \tag{13}$$

$$c_{uv} = \gamma_{uv} + \sum_{j=1}^{|D_u|} \delta_{uj}(v) \tag{14}$$

where R_u denotes the set of user IDs who follow u and I is an indicator function, which returns 1 if a given condition is true and 0 otherwise. The derivations of the above equations are provided in Appendix B.

Variational M-step: In order to find the parameters $\mathbb{X}$ and $\mathbb{Y}$ which maximize $\mathbb{F}$ where $\mathbb{X}$ and $\mathbb{Y}$ denote the set of $\overrightarrow{x_u}$ and $\overrightarrow{y_u}$ for every $u \in U$ respectively, we use the iterative method for matrix factorization in [18]. If we represents $\mathbb{F}$ with the terms related to $\mathbb{X}$ and $\mathbb{Y}$ only, $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ becomes

$$\mathbb{F}(\mathbb{X}, \mathbb{Y}) = -\sum_{u=1}^{|U|}\sum_{v=1}^{|U|} \rho_{uv}(f_{uv} - \overrightarrow{x_u}^T \cdot \overrightarrow{y_v})^2 - \sum_{u=1}^{|U|}\sum_{v=1}^{|U|} \rho_{uv} \log\zeta_u \tag{15}$$

Note that computing $\mathbb{X}$ and $\mathbb{Y}$ that maximize the above $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ is exactly equivalent to the cost minimization in weighted matrix factorization addressed in [18]. We thus borrow its iterative algorithm to find $\mathbb{X}$ and $\mathbb{Y}$ in the M-step. Furthermore, since many of recent works on matrix factorization introduce the regularization term to their cost functions to avoid over-fitting, we regularize $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ too as follows:

$$\mathbb{F}(\mathbb{X}, \mathbb{Y}) = -\sum_{u=1}^{|U|}\sum_{v=1}^{|U|} \rho_{uv}(f_{uv} - \overrightarrow{x_u}^T \cdot \overrightarrow{y_v})^2 - \sum_{u=1}^{|U|}\sum_{v=1}^{|U|} \rho_{uv} \log\zeta_u + \tau\sum_{u=1}^{|U|}(\|\overrightarrow{x_u}\|^2 + \|\overrightarrow{y_u}\|^2) \tag{16}$$

where τ is a regularization constant and $\|\overrightarrow{x}\|$ is the ℓ_2-norm of a vector $\overrightarrow{x}$. Then, the minimization in M-step is accomplished by iteratively updating both $\overrightarrow{x_u}$ and $\overrightarrow{y_u}$ with

$$\overrightarrow{x_u} = (Y^T \cdot P_u \cdot Y + \omega I)^{-1} \cdot Y^T \cdot P_u \cdot \overrightarrow{f_u} \tag{17}$$

$$\overrightarrow{y_u} = (X^T \cdot Q_u \cdot X + \omega I)^{-1} \cdot X^T \cdot Q_u \cdot \overrightarrow{f_u} \tag{18}$$

where X and Y are the $|U| \times |U|$ matrices such that the uth column of each matrix is $\overrightarrow{x_u}$ and $\overrightarrow{y_u}$ respectively. Furthermore, in the above equations, P_u is a $|U| \times |U|$ diagonal matrix such that the value in the vth row and vth column of P_u is $\rho_{uv} - p(f_{uv})\sum_{z=1}^{|Z|}\rho_{uv}$ with a user v whom u follows (i.e., $v \in F_u$) where $p(f_{uv})$ is the probability function in Eq. (1). Similarly, Q_u is also a $|U| \times |U|$ diagonal matrix such that the value in the vth row and the vth column of Q_u is $\rho_{uv} - p(f_{uv})\sum_{z=1}^{|Z|}\rho_{uv}$ with a user v who follows u (i.e., $v \in R_u$). The detailed steps to derive Eqs. (17) and (18) by extending the matrix factorization algorithm of [18] are provided later in Appendix C.

We refer to the inference algorithm using the variational EM technique for our model as *TWILITE-VEM*. The algorithm iterates the E-step and the M-step repeatedly until the lower bound of log-likelihood $\mathbb{F}$ in Eq. (4) converges. For the variational E-step and M-step, we invoke *VEM-ITER*. After *VEM-ITER* finishes in each iteration step, we execute *OPT-PRIORS* to find the optimal Dirichlet priors α and β that maximize the lower bound $\mathbb{F}$. Since the terms which contain α in $\mathbb{F}$ are exactly the same with those for LDA mode in [5], we borrow Newton's method *OPT-PRIORS* in [5] to update α. Similarly, β is also updated by *OPT-PRIORS*. The pseudocode of *TWILITE-VEM* is provided in Fig. 5. Since *TWILITE-VEM* is only guaranteed to find a local maximum of the likelihood, we

Function TWILITE-VEM($U, V, T, F, Z, \alpha, \beta, \gamma$)
begin
$\forall u \in U, \forall z \in Z$ and $\forall u \in U$, initialize $\vec{a_u}$, $\vec{b_z}$ and $\vec{c_u}$ randomly;
until $\mathbb{F}$ converges **do** {
  VEM-ITER($A, B, C; X, Y$);
  OPT-PRIORS(α, β);
}
end

Fig. 5. The TWILITE-VEM algorithm.

perform multiple trials to obtain a local maximum that is close to a global optimum.

The pseudocode of *VEM-ITER*, which is invoked in *TWILITE-VEM* and executes the variational E-step and M-step, is presented in Fig. 6. In the E-step of *VEM-ITER*, we update the variational parameters $\overrightarrow{\phi_{uj}}$, $\overrightarrow{\delta_{uj}}$, $\overrightarrow{a_u}$, $\overrightarrow{b_z}$ and $\overrightarrow{c_u}$ to maximize the lower bound $\mathbb{F}$ using Eqs. (9)–(14). In a naive method to compute the equations, we can simply compute each variational parameter one by one using those equations. However, to improve the convergence rate of the variational EM algorithm by incooperating the dependencies between variational parameters, we utilize the nested variational inference algorithm by introducing another iteration in *VEM-ITER* in the same way as the inference algorithms in [1,5] work. Since Eq. (9) to update $\overrightarrow{\phi_{uj}}$ for each user u is dependent on the variational parameter $\overrightarrow{\delta_{uj}}$ only and Eq. (10) for $\overrightarrow{\delta_{uj}}$ also depends on $\overrightarrow{\phi_{uj}}$ only, for each u, we compute Eqs. (9) and (10) repeatedly until we achieve the convergence of $\mathbb{F}(\overrightarrow{\phi_{uj}}, \overrightarrow{\delta_{uj}})$ shown below:

$$\begin{aligned}\mathbb{F}(\overrightarrow{\phi_{uj}}, \overrightarrow{\delta_{uj}}) &= \sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|}\sum_{z=1}^{|Z|}\sum_{v=1}^{|U|} I(f_{u\ell}=v)\phi_{uj}(z)\delta_{uj}(\ell)(\Psi(a_{vz})-\Psi(a_{v\bullet})) \\ &+ \sum_{j=1}^{|D_u|}\sum_{t=1}^{|V|}\sum_{z=1}^{|Z|} I(w_{uj}=v_t)\phi_{uj}(z)(\Psi(b_{kt})-\Psi(b_{z\bullet})) \\ &+ \sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|} \delta_{uj}(\ell)(\Psi(c_{u\ell})-\Psi(c_{u\bullet})) \\ &- \sum_{j=1}^{|D_u|}\sum_{z=1}^{|Z|} \phi_{uj}(z)\log\phi_{uj}(z) - \sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|} \delta_{uj}(\ell)\log\delta_{uj}(\ell)\end{aligned} \tag{19}$$

where the equation is expressed in terms of $\overrightarrow{\phi_{uj}}$ and $\overrightarrow{\delta_{uj}}$ only. The lines 3–11 in the pseudocode of *VEM-ITER* represent such steps of our nested variational inference. After calculating $\overrightarrow{\phi_{uj}}$ and $\overrightarrow{\delta_{uj}}$ with every user, we next update the rest of variational parameters $\overrightarrow{a_u}$, $\overrightarrow{b_z}$ and $\overrightarrow{c_u}$ (lines 12–17). Finally, in the M-step of *VEM-ITER*, we update $\overrightarrow{x_u}$ and $\overrightarrow{y_u}$ that maximize the lower bound $\mathbb{F}$ by repeatedly calculating Eqs. (17) and (18) until $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ in Eq. (16) converges (lines 20–24).

Computing posterior probabilities from variational parameters: We finally compute the posterior probabilities Θ, Φ and Π utilizing the variational parameters calculated at the last iteration when $\mathbb{F}$ is converged. Using the variational parameters $\overrightarrow{a_u}$, $\overrightarrow{b_k}$ and $\overrightarrow{c_u}$ which are the Dirichlet priors of $\overrightarrow{\theta_u}$, $\overrightarrow{\beta_k}$ and $\overrightarrow{\pi_u}$ respectively, we can set those posterior distributions to their expectations computed as

Function VE-ITER($A, B, C; X, Y$)
begin
1. initialize $a_{uz}^{next} = \alpha_z$, $b_{zt}^{next} = \beta_t$ and $c_{u\ell}^{next} = \gamma_{u\ell}$;
2. **for** u = 1 to $|U|$ **do** {
3.   initialize $\vec{\phi_{uj}}$ and $\vec{\delta_{uj}}$ randomly;
4.   **until** $\mathbb{F}(\vec{\phi_{uj}}, \vec{\delta_{uj}})$ in Equation (19) converges **do** {
5.   **for** j = 1 to $|N_u|$ **do** {
6.   **for** z = 1 to $|Z|$ **do** update $\phi_{uj}(z)$ by Equation (9);
7.   **for** f = 1 to $|F_u|$ **do** update $\delta_{uj}(\ell)$ by Equation (10);
8.   }
9.   normalize $\phi_{uj}(z)$ to satisfy $\sum_{z=1}^{|Z|}\phi_{uj}(z) = 1$;
10.   normalize $\delta_{uj}(v)$ to satisfy $\sum_{v=1}^{|F_u|}\delta_{uj}(v) = 1$;
11.   }
12.   **for** z = 1 to $|Z|$ **do** {
13. $a_{uk}^{next} = a_{uk}^{next} + \phi_{uj}(z)\delta_{uj}(u)$;
14.   **for** v =1 to $|F_u|$ **do** $a_{f_{uv},z}^{next} = a_{f_{uv},z}^{next} + \phi_{uj}(z)\delta_{uj}(v)$;
15. $b_{zw_{uj}}^{next} = b_{zw_{uj}}^{next} + \phi_{uj}(z)$;
16.   }
17.   **for** v = 1 to $|F_u|$ **do** $c_{uv}^{next} = c_{uv}^{next} + \delta_{uj}(v)$;
18. }
19. set $a_{uz} = a_{uz}^{next}$, $b_{zt} = b_{zt}^{next}$ and $c_{u\ell} = c_{u\ell}^{next}$;
20. **for** u = 1 to $|U|$ **do** { **for** v = 1 to $|U|$ **do** update ρ_{uv} by Equation (11); }
21. **until** $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ in Equation (16) converges **do** {
22.   **for** u = 1 to $|U|$ **do** update $\vec{x_u}$ by Equation (17);
23.   **for** u = 1 to $|U|$ **do** update $\vec{y_u}$ by Equation (18);
24. }
end

Fig. 6. An iteration of TWILITE-VEM.

follows [38]:

$$\theta_{uz} = a_{uz} \Big/ \sum_{j=1}^{|Z|} a_{uj} \quad \text{for } z = 1, \ldots, |Z|.$$

$$\beta_{zt} = b_{zt} \Big/ \sum_{j=1}^{|V|} b_{zj} \quad \text{for } t = 1, \ldots, |V|.$$

$$\gamma_{uv} = c_{uv} \Big/ \sum_{j=1}^{|F_u|} c_{uj} \quad \text{for } v = 1, \ldots, |F_u|.$$

6. Utilizing our model parameters for recommendations

Once the posterior distributions in our model are estimated, recommendations can be made by utilizing the model parameters. In this section, we propose the top-K recommendation algorithms for a user to select followees and tweet messages respectively.

Recommending top-K followees: To recommend top-K followees that a user u would like to follow in Twitter utilizing the estimated model parameters, we develop two recommendation methods based on friend-of-friend and personalized PageRank.

(1) *Friend-of-friends recommendation*: A simple followee recommendation for a target user u is to select top-K users among the followees of u's followees. Using the estimated posterior distributions $\overrightarrow{\pi_u}$, we compute the score for each user v in $((\bigcup_{u' \in F_u \wedge u \neq u'} F_{u'}) - F_u)$, which is the set of followees of u's followees who do not appear in F_u, as follows:

$$FS_u(v) = (1-\omega_{FOF})\sum_{u'=1}^{|U|}\sum_{\ell=1}^{|F_u|}\sum_{\ell'=1}^{|F_{u'}|} I(f_{u\ell}=u')I(f_{u'\ell'}=v)\pi_{uu'}\cdot\pi_{u'v} + \omega_{FOF}\overrightarrow{x_u}^T\overrightarrow{y_v} \tag{20}$$

where the first term denotes the probability that u is influenced by v when u selects each word in his tweets using the distributions $\overrightarrow{\pi_u}$s, and the second term a

u	$\vec{\theta_u}$
a_1	⟨ 0.23, 0.77 ⟩
a_2	⟨ 0.28, 0.72 ⟩
a_3	⟨ 0.30, 0.70 ⟩
a_4	⟨ 0.28, 0.72 ⟩
b_1	⟨ 0.55, 0.45 ⟩
b_2	⟨ 0.55, 0.45 ⟩
b_3	⟨ 0.49, 0.51 ⟩
b_4	⟨ 0.70, 0.30 ⟩
b_5	⟨ 0.48, 0.52 ⟩

b

u	F_u	$\vec{\pi_u}$
a_1	a_2, a_3, a_4	⟨ 0.29, 0.24, 0.23, 0.24 ⟩
a_2	a_1, a_4, b_2, b_3	⟨ 0.21, 0.22, 0.21, 0.18, 0.18 ⟩
a_3	a_1, b_1	⟨ 0.34, 0.42, 0.24 ⟩
a_4	a_1, a_2	⟨ 0.32, 0.35, 0.33 ⟩
b_1	b_2, b_3, b_4, a_3	⟨ 0.23, 0.21, 0.23, 0.32 ⟩
b_2	b_1, b_3, b_4	⟨ 0.24, 0.23, 0.24, 0.28 ⟩
b_3	b_1, b_2, b_4	⟨ 0.18, 0.20, 0.20, 0.28, 0.14 ⟩
b_4	a_1, b_1, b_2	⟨ 0.24, 0.26, 0.25, 0.25 ⟩
b_5	a_1, b_3	⟨ 0.27, 0.20, 0.53 ⟩

c

u	$\vec{x_u}$	$\vec{y_u}$
a_1	⟨ 0.7136, 0.7006 ⟩	⟨ 0.7168, 0.6973 ⟩
a_2	⟨ 0.7179, 0.6962 ⟩	⟨ 0.7148, 0.6994 ⟩
a_3	⟨ 0.7169, 0.6971 ⟩	⟨ 0.7177, 0.6963 ⟩
a_4	⟨ 0.7130, 0.7011 ⟩	⟨ 0.7148, 0.6994 ⟩
b_1	⟨ 0.7232, 0.6906 ⟩	⟨ 0.7225, 0.6914 ⟩
b_2	⟨ 0.7232, 0.6907 ⟩	⟨ 0.7225, 0.6914 ⟩
b_3	⟨ 0.7228, 0.6911 ⟩	⟨ 0.7222, 0.6917 ⟩
b_4	⟨ 0.7231, 0.6907 ⟩	⟨ 0.7228, 0.6911 ⟩
b_5	⟨ 0.7221, 0.6917 ⟩	⟨ 0.7221, 0.6917 ⟩

d

	harry	crisis	economic	potter	imf	movie	wallstreet	finance	emwatson
Topic 1	0.0275	0.2250	0.2771	0.0184	0.2250	0.0157	0.0467	0.1715	0.0207
Topic 2	0.4107	0.0238	0.0316	0.2130	0.0239	0.2134	0.1937	0.0212	0.2794

Fig. 7. The estimated posterior distributions $\vec{\theta_u}$, $\vec{\pi_u}$, $\vec{\varphi_z}$, $\vec{x_u}$ and $\vec{y_u}$.

represents the score that measures the extent to which u follows v using the latent factor vectors $\vec{x_u}$ and $\vec{y_v}$ weighted by a constant ω_{FOF}. Then, we choose the top-K users whose $FScore_u(v)$s are the K largest among the users in $((\bigcup_{u' \in F_u \wedge u \neq u'} F_{u'}) - F_u)$.

(2) *Personalized PageRank recommendation*: We next utilize the random surfer model of personalized PageRank for recommendation in [41]. Let a user u is a random surfer who visits other Twitter users by following the links to followees. The random surfer always begins surfing by following one of his own followees. In the currently visiting user v, he decides to follow the link to v's followees with probability d and visits the jth user in F_v with probability π_{uj}. Sometimes, with probability $(1-d)$, the random surfer stops to follow the followees' links and starts a new surfing from one in F_u. In this case, the random surfer u chooses a starting user $v \in F_u$ with probability π_{uj} when v is the jth user in F_u. The recursive equation for the rank $PR_u(v)$ that the random surfer u visits the user v is formulated as the following:

$$PR_u(v) = (1-d) \cdot \sum_{j=1}^{|F_u|} I(v = f_{uj})\pi_{uj} + d \cdot \sum_{w \in R_v} \sum_{j=1}^{|F_u|} I(v = f_{wj})\pi_{wj} \cdot PR_u(w). \qquad (21)$$

The rank score $PR_u(v)$ can be computed with the power method (see [33]) used for computing PageRank in [4], which repeats the computation in Eq. (21) until $PR_u(v)$ of every user does not change any more. After we obtain $PR_u(v)$ for every user v, we can recommend the top-K ranked users to the user u by computing $PRS_u(v)$ defined below:

$$PRS_u(v) = (1-\omega_{PR})PR_u(v) + \omega_{PR}\vec{x_u}^T\vec{y_v} \qquad (22)$$

where the second term is a score using latent factor vectors $\vec{x_u}$ and $\vec{y_v}$ weighted by ω_{PR} similar to Eq. (20).

Recommending top-K tweets: Let t denote a tweet message which is a set of word IDs in V. Assume that t is written by a user v in U. It is important to recommend interesting tweets even if they are not written by the followees. Thus, we compute the score function $TWS_v(t)$ that measures the extent to which u would like to read the tweet message t posted by v as follows:

$$TWS_u(t) = (1-\omega_{TW}) \cdot \sum_{j=1}^{|F_u|} I(v = f_{uj})\pi_{uj} + \omega_{TW} \sum_{z=1}^{|Z|} \theta_{vz} \cdot \max_{w \in t} \phi_{zw}, \qquad (23)$$

where ω_{TW} is a constant for weighting.

The first term of $S_u(t)$ in Eq. (23) is to compute u's preference to the author v of the tweet t. The second term in Eq. (23) represents u's preference to the topics appearing in the tweet t. Since the length of each tweet t is generally very short, for a specific topic, only a small number of words describing its topics will show up in t. Thus, if we simply use $\sum_{w \in t}\phi_{kw}$ as the similarity of t to each topic k, long tweet messages tend to have high scores for every topic, since $\sum_{w \in t}\phi_{kw}$ becomes generally larger with increasing number of words in t. Thus, we use $\max_{w \in t} \phi_{kw}$ as the similarity of t to the topic k appearing in the tweet t instead. We also found this tendency by experimental study with real-life data.

Example 6.1. Consider the example of users and their tweets in Fig. 1. Since the tweets contain two topics which are 'Financial crisis' and 'Harry Potter', we set the number of topics to 2 in this example. The estimated posterior distributions $\vec{\theta_u}$ and $\vec{\pi_u}$ for the nine users are shown in Fig. 7(a) and (b) respectively. The model parameters $\vec{x_u}$ and $\vec{y_u}$ are presented in Fig. 7(c).

Top-1 followee recommendation: Suppose that we recommend a new followee for a_3 whose two followees are $\{a_1, b_1\}$. If we decide to choose candidates by considering the 'friends of friends', the candidates become $\{a_2, a_4, b_2, b_3, b_4\}$. The values of $FS_u(v)$s in Eq. (20) for $u = a_3$ with $\omega_{FOF} = 0.5$ are provided in Fig. 8(a). As we expected, our algorithm recommends a_2 and a_4 to a_3 rather than b_2, b_3 and b_4.

Let us discuss the top-1 followee recommendation for b_4. If we choose the candidates by utilizing the 'friends of friends' criteria again, the candidates become $\{a_2, a_3, a_4, b_3\}$. For each candidate user, we obtain the values of $FS_u(v)$s with $\omega_{FOF} = 0.5$ as shown in Fig. 8(b). Our algorithm selects b_3 for the top-1 followee recommendation as we expected. Note that *TWITOBI* recommends a_1 for the same example as discussed in Example 3.1. Thus, we can see that *TWILITE* is more efficient than the existing work.

a

v	a_2	a_4	b_2	b_3	b_4
$FS_u(v)$	0.5577	0.5577	0.52522	0.53227	0.52681

b

v	a_2	a_3	a_4	b_3
$FS_u(v)$	0.5322	0.5599	0.53222	0.5638

Fig. 8. The score $FS_u(v)$ for (a) $u = a_3$ and (b) $u = b_4$.

u	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4	b_5
b_4	0.5778	0.6285	0.5710	0.5710	0.6585	0.6389	0.6261	-	0.6261
b_5	0.6163	0.6226	0.6116	0.6116	0.6727	0.6392	0.6173	0.6599	-

Fig. 9. The score $TW_u(v)$ of the tweet message of each user with $u = b_4$ and $u = b_5$.

Top-1 tweet recommendation: Assume that we want to determine a tweet message to recommend to b_4 and b_5. Since their tweets are mainly about ‘Financial crisis’, we expected that they will prefer the tweets by b_1, b_2 and b_3 to those by the other users whose tweets handling the same topic to b_4 and b_5. In Fig. 9, we present the scores of $TWS_u(t)$s using $\omega_{TW} = 0.5$ for all the tweets by the nine users. For both b_4 and b_5, our algorithm recommends the tweet of b_1 who is not only interested in ‘Financial crisis’ but also connected from many users with the same interest. □

7. Experiments

We empirically evaluated the performance of our proposed algorithms. All experiments reported in this section were performed on the machines with Intel(R) Core(TM)2 Duo CPU 2.66 GHz and 2 GB of main memory running Linux operating systems. All algorithms were implemented using a Java Compiler of version 1.7. The source codes of the algorithms implemented in Java are available online.[1] For our experiments, we implemented the following inference algorithms.

- *TWILITE-VEM*: This denotes the implementation of our variational EM algorithm presented in Section 5.
- *TWITOBI-EM*: It is the implementation of the EM algorithm presented in our previous work [22].
- *LDA-VEM*: This denotes the implementation of the variational EM algorithm of LDA [5].
- *MF*: This represents the matrix factorization algorithm proposed in [18].

We also implemented the recommendation algorithms using the model parameters obtained with the above algorithms as follows:

- *TWILITE-FOF-F*: This is the implementation of our top-*K* followee recommendation algorithm based on the ‘friend of friends’ criteria presented in Section 6 which uses $\overrightarrow{\pi_u}$s produced by *TWILITE-VEM*.
- *TWILITE-PR-F*: It represents our top-*K* followee recommendation algorithm based on personalized PageRank presented in Section 6 which utilizes $\overrightarrow{\pi_u}$s as the weights to select the followees in random walking.
- *TWILITE-T*: It is the implementation of our top-*K* tweet recommendation algorithm introduced in Section 6.
- *TWITOBI-F*: This is the implementation of the top-*K* followee recommendation algorithm in [22] which uses the model parameters obtained by *TWITOBI-EM*.
- *TWITOBI-T*: It is the implementation of the top-*K* tweet recommendation algorithm in [22] that utilizes the result obtained by *TWITOBI-EM*.
- *LDA-T*: This denotes the recommendation algorithm utilizing the posterior distribution estimated by *LDA-VEM*. For each tweet message t, we recommend the top-*K* messages according to the score $\sum_{z=1}^{|Z|} \theta_{uz} \cdot \max_{w \in t} \phi_{zw}$ for a user u similar to *TWILITE-T*.
- *PR-F*: It is the followee recommendation using personalized PageRank method in [41].
- *MF-F*: It represents the followee recommendation using the latent factor vectors obtained by *MF* [18].
- *TFIDF-F*: It is the implementation of the simple followee recommendation method proposed in [14]. This algorithm uses TF-IDF weighting with the followees/followers of a user and the words in tweets written by the user as briefly described in Section 2.
- *TFIDF-T*: It is the implementation of the simple tweet recommendation algorithm presented in [9] which uses TF-IDF weighting with the words in tweets written by users as we summarized in Section 2.

Since a single trial of a variational EM algorithm generally finds a local maximum of the likelihood, variational EM algorithms typically perform multiple trials to obtain several local maxima and choose the best one as the model parameter values. In our experiments, all EM algorithms perform 10 trials to handle the problem of local maxima. We also assume that the convergence of each trial is obtained when the difference of the lower bounds $\mathbb{F}$ of previous and current iterations is less than 100.

7.1. Data sets

For experimental study, we evaluate the algorithms on real-life data sets. We downloaded 12,098,339 tweet messages of 8405 Twitter users using the Twitter API [34]. We call this data as ORG-DATA. The users in ORG-DATA have 221 followees on the average.

Test data: To generate a test data set for evaluating the top-K followee recommendation algorithms, we first selected 400 users randomly among the users each of whom has at least 50 followees in ORG-DATA. For each selected user u, we next extracted 10 followee relationships randomly among u's followees from ORG-DATA.

[1] http://kdd.snu.ac.kr/twilite/twilite-src.zip

User	Tweets
BarackObama	Presenting:41:kalebaskew National:26:kalebaskew Teacher:7:changeequation Year Award:40:thedrivein 11:45am
BarackObama	President's:26:kalebaskew remarks:26:kalebaskew death:19:kalebaskew Osama:19:SenJohnMcCain bin:19:SenJohnMcCain Laden:19:SenJohnMcCain full video:10:SenJohnMcCain
BarackObama	July:40:thedrivein we begin to bring:48:thedrivein our:6:thedrivein troops:19:SenJohnMcCain home:48:thedrivein from Afghanistan:19:SenJohnMcCain SOTU:26:kalebaskew
FP_Magazine	Libyan:19:self opposition:19:self leader:26:SenJohnMcCain coming Washington:26:politico next week:6:SenJohnMcCain
FP_Magazine	ibishblog Bahrain:19:self creating:26:politico new terrorist:19:SenJohnMcCain threat:19:SenJohnMcCain
changeequation	Did U know Less than 14 engineering:13:semlink students:7:BayerMSMS are African:13:semlink American:26:BayerMSMS Hispanic:7:BayerMSMS
changeequation	RT Samsungtweets Did U know 2009 only 34 8th graders:7:BayerMSMS were proficient:7:BayerMSMS higher:31:BayerMSMS math:7:BayerMSMS test:26:BayerMSMS
changeequation	Shortages:6:BayerMSMS force:40:BayerMSMS educators:7:BayerMSMS teach:31:semlink subjects outside:48:semlink specialty:40:BayerMSMS areas:40:BayerMSMS
RPLife	fan:4:TwilightPoison videos:4:TwilightPoison cute:4:TwilightPoison Rob:4:TwilightPoison smiling:38:self talking:4
RPLife	VIDEO:4:TwilightPoison ET Talks:4:TwilightPoison MTV:4:TwilightPoison Movie:4:TwilightPoison Awards:4:TwilightPoison Rob:4:TwilightPoison Taylor:4:TwilightPoison kiss:38:self Rob's:4:TwilightPoison F Bomb
RPLife	Rob:4:TwilightPoison talks:4:TwilightPoison BD wedding:5:self Best:4:TwilightPoison Kiss:38:self award:4:TwilightPoison career:31:self after:5:self Twilight:4:TwilightPoison

Fig. 10. An example of users and their tweets.

Then, the selected followee relationships of the 400 users were removed from ORG-DATA. We refer to the selected followee relationships as *TEST-F-DATA*. In addition, to produce a test data set for evaluating the top-K tweet recommendation algorithms, we chose another 400 users randomly from ORG-DATA. For each selected user u, we extracted 10 tweet messages randomly selected among those posted by u from ORG-DATA. Then, the selected tweets for the 400 users were eliminated from ORG-DATA. We call the selected tweet messages *TEST-T-DATA*.

Training data: After removing all followee relationships in TEST-F-DATA and all tweet messages in TEST-T-DATA from ORG-DATA, the remaining data in ORG-DATA was used to estimate model parameters for all inference algorithms. We refer to the training data set as *TRAIN-DATA*.

7.2. Inference examples with Twitter data

We first illustrate the result of probabilistic modeling by *TWILITE* using real-life data. When we execute *TWILITE-VEM* with *TRAIN-DATA*, we store the variational parameters $\overrightarrow{\varphi_{uj}}$ and $\overrightarrow{\delta_{uj}}$ which represent the probabilities that each word w_{uj} in D_u is selected by which topic and is influenced by which followees of u respectively. In other words, $\overrightarrow{\varphi_{uj}}$ and $\overrightarrow{\delta_{uj}}$ denote the probability distributions of the random variables z_{uj} and s_{uj} introduced in Section 3 respectively. In Fig. 10, we present 11 tweet messages written by 4 users with annotating each word w_{uj} with the topic ID whose probability is the largest among the elements of $\overrightarrow{\varphi_{uj}}$ and the user ID whose probability is the largest among $\overrightarrow{\delta_{uj}}$. For example, 'fan:4:TwilightPoison' in the first message by Twitter user 'RPLife' represents that the word 'fan' is selected for the 4th topic and influenced by one of his followees with user ID 'TwilightPoison'.

As we expected, these distributions seem to capture effectively the underlying topics and followees that affect each user when he selects the words in his tweet messages. For example, the topic 19 represents the topic of 'terrorism' which is used for selecting the words 'death', 'Osama bin Laden', 'troops', 'Libyan opposition' and 'threat' in the example. We can see that in the tweet messages of 'BarackObama', which is the account of US president, the words selected for the topic 19 are mainly influenced by the followee 'SenJohnMcCain' who is a Republican Senator of US and has posted many tweets about terrorism. The messages by 'FP_Magazine' who frequently posts political news are also shown to be affected by 'SenJohnMnCain' when they post news about terrorism.

The user ID 'changeequation' mainly posts messages about the issues of mathematics education. The user has 77 followees with interests on various topics but our inference algorithm founds that the words in its tweet messages are selected under the influence of 'BayerMSMS' and 'semlink' all of which mainly post messages about mathematics education. Another example is the case with the user ID 'RPLife'. He is a movie star 'Rob Pattinson' featured in the movie 'Twilight' and writes messages about the topic 4 which frequently occurs for the words 'fan',

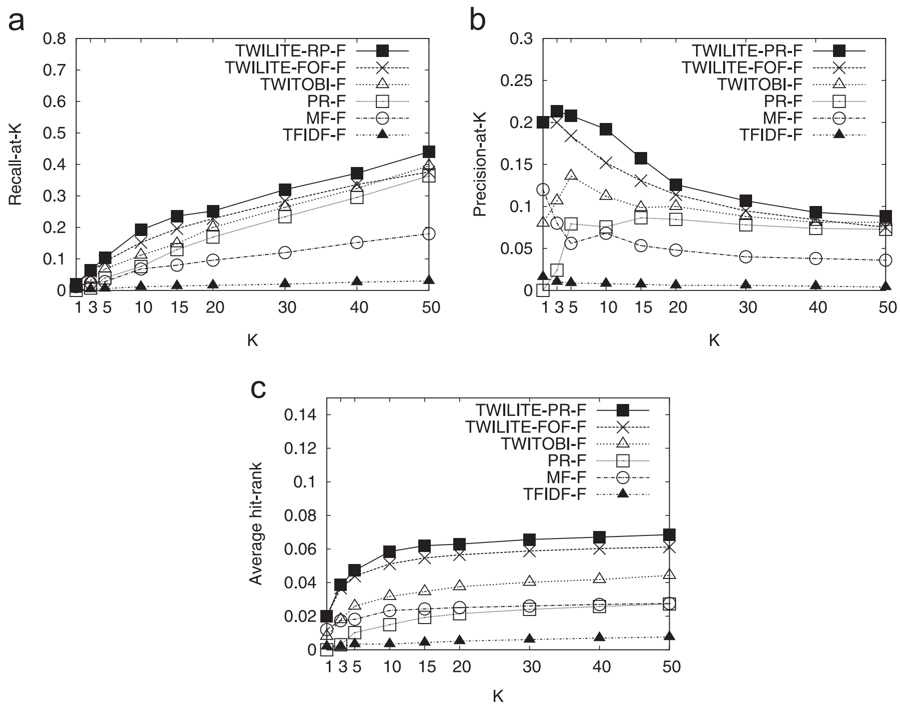

Fig. 11. Top-K followee recommendation with varying K: (a) recall-at-K; (b) precision-at-K; and (c) average hit-rank.

'video', 'Rob', 'MTV', 'movie' and 'Twilight'. He is also often affected by 'TwilightPoison' who is a fan of the movie. With these illustrative examples, it is confirmed that *TWILITE* captures the behaviors of users in Twitter quite well.

7.3. Qualities of our recommendation algorithms

We conducted our experiments with varying the number of recommendations K, the number of hidden topics $|Z|$, the constant λ in Eq. (1) to compute the probability that a user follows another user, and the constant τ in Eq. (17) which is used in the matrix factorization for regularization. We also varied three weights ω_{FOF}, ω_{PR} and ω_{TW} in the score functions for recommending top-K followees and tweets in Eqs. (20), (22) and (23) respectively. The default values of these parameters are $K=30$, $|Z|=50$, $\lambda=0.8$, $\omega_{FOF}=0.9$, $\omega_{PR}=0.1$ and $\omega_{TW}=1$. In all experiments, we calculated the model parameters using TRAIN-DATA. When we evaluate top-K followee and tweet recommendation algorithms, TEST-F-DATA and TEST-T-DATA were used respectively.

Quality measures: We evaluated three quality measures called *recall-at-K*, *precision-at-K* [35] and *average hit-rank* [12]. (The recall-at-K is also known as *hit-rate* [12].) In top-K followee recommendations, for a test user u, let h be the number of the true followees in the recommended top-K followees and $n_T(u)$ be the number of u's true followees in the test data set. The recall-at-K and precision-at-K for u are $h/n_T(u)$ and h/K respectively. The average hit-rank is to measure the effectiveness of ranking for each test user. When $p_1, p_2, \ldots, p_h$ are the position of the true followees in the recommended top-K followees, the average hit-rank for a test user is $(1/n_T(u)) \cdot \sum_{i=1}^{h}(1/p_i)$. As the true followees appear with high ranks in the recommended followees, this measure becomes larger.

In top-K tweet recommendations, for each test user u, we select top-K tweets among the tweets in the test data set and compute the recall-at-K and average hit-rank similarly.

Top-K followee recommendation: We varied the number of recommendations K from 1 to 50. Fig. 11(a), (b) and (c) show the averages of *recall-at-Ks*, *precision-at-Ks* and *average hit-ranks* with 400 test users respectively. As we recommend more followees, since we have more chance to answer the true followees correctly, both recall-at-K and average hit-rank grow gradually with increasing K. However, the precision-at-K measure decreases when K is increased because more false followees can be recommended with a larger K.

The graphs confirm that our new recommendation algorithm *TWILITE-PR-F* with every range of K outperforms the other ones in terms of recall-at-K, precision-at-K and average hit-rank. The second best performer is *TWILITE-FOF-F* and the third one is *TWITOBI-F* which is our previous work. As expected, *TFIDF-F* was the worst performer since it simply recommends the followees using the cosine similarity between the sets of friends for a pair of users.

Note that the precision-at-K of *TWILITE-PR-F* and *TWILITE-FOF-F* is significantly better with small values of K than the other algorithms. This is because our algorithms have good characteristics of producing correct answers to have higher ranks than the other algorithms. For the same reason, for the average hit-rank, our algorithms show

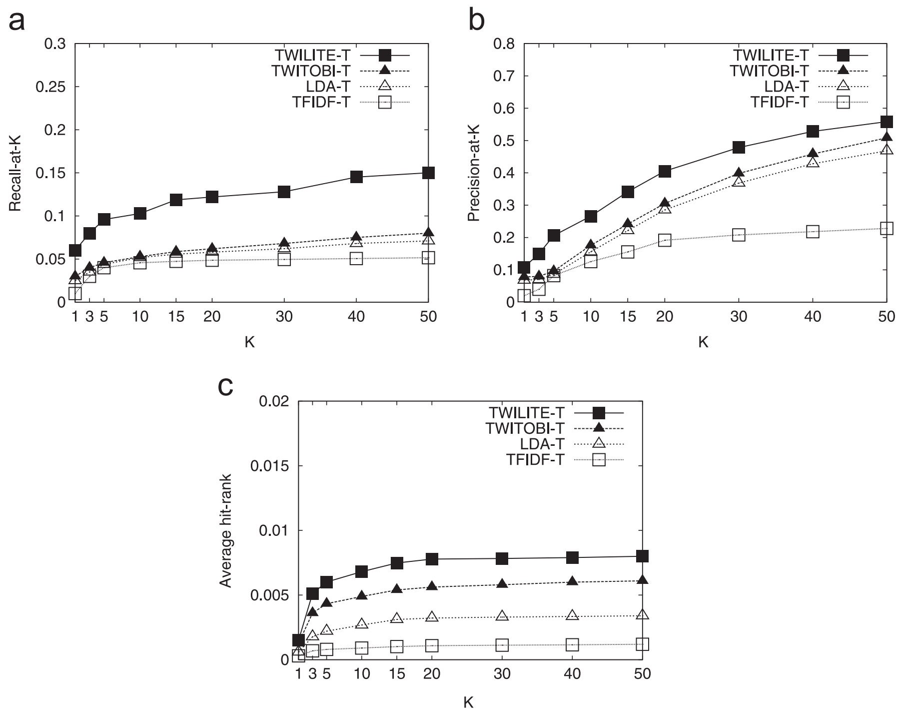

Fig. 12. Top-*K* tweet recommendation with varying *K*: (a) recall-at-K; (b) precision-at-K; and (c) average hit-rank.

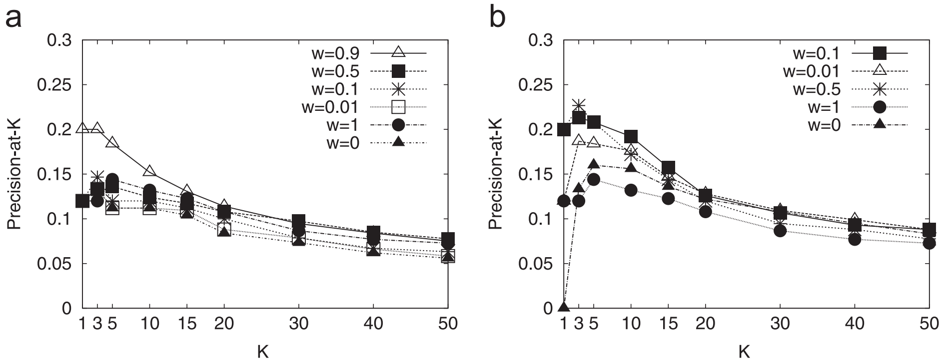

Fig. 13. The precision-at-K for top-*K* tweet recommendations with varying weights: (a) varying ω_{FOF} and (b) varying ω_{PR}.

much larger performance improvement than that improvement for recall-at-K or precision-at-K compared to the other algorithms.

Top-K tweet recommendation: With varying *K* from 1 to 50, we plotted the averages of *recall-at-Ks*, *precision-at-Ks* and *average hit-ranks* for 400 test users in Fig. 12(a), (b) and (c) respectively. The evaluated algorithms are *TWILITE-T*, *TWITOBI-T*, *LDA-T* and *TFIDF-T*. The graphs illustrate that *TWILITE-T* for every *K* outperforms the other recommendation algorithms in terms of every quality measure. Similar to the top-*K* followee recommendation, *TFIDF-T* was the worst performer.

Varying ω_{FOF}, ω_{PR} and ω_{TW}: To find the best setting of the parameters ω_{FOF} and ω_{PR}, we ran *TWILITE-FOF-F* and *TWILITE-PR-F* with varying ω_{FOF} and ω_{PR} from 0 to 1. The precision-at-Ks of those two algorithms are presented in Fig. 13. For both algorithms, setting those weights to 1 implies that we select the top-*K* followees using the latent factor vectors only obtained by matrix factorization without the first terms in both Eqs. (20) and (22). The performances of *TWILITE-FOF-F* and *TWILITE-PR-F* were the best when ω_{FOF} and ω_{PR} are set to 0.9 and 0.1 respectively. We also varied ω_{TW} with *TWILITE-T* and obtained the best performance with $\omega_{TW} = 0.5$.

Varying $|Z|$ and d: We varied the number of topics $|Z|$ from 30 to 70 while increasing *K* from 1 to 50 and plotted the precision-at-Ks of *TWILITE-PR-F* in Fig. 14(a). Similarly, we varied the number of latent factors *d* from 30 to 70 and plotted the precision-at-Ks of *TWILITE-PR-F* in Fig. 12(b). We can see that the precision of recommendations is the best when $|Z| = 50$ and $d = 50$ in both graphs. With more topics and latent factors, the accuracies of recommendations decrease due to the overfitting problem in the variational EM algorithm and matrix factorization that

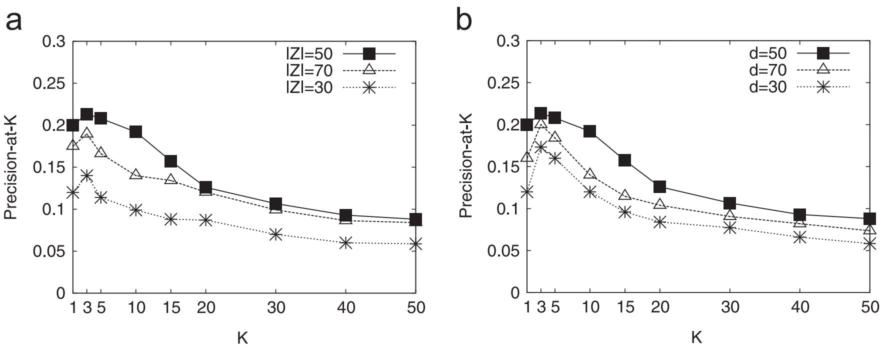

Fig. 14. The precision-at-K for top-K tweet recommendations with varying (a) $|Z|$ and (b) d.

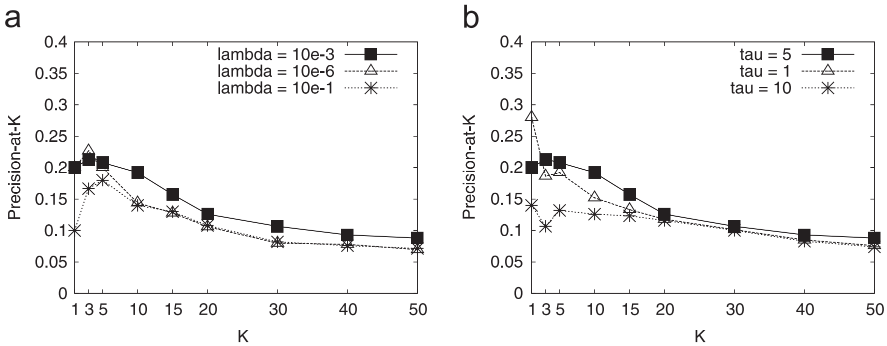

Fig. 15. The precision-at-K for top-K followee recommendation with varying (a) λ and (b) τ.

occurs with a large number of topics or latent factors [5]. Thus, we set the default value of $|Z|$ and d to 50 and 50 respectively.

Varying λ and τ: We next estimate the model parameters with *TWILITE-VEM* with varying λ and τ. Recall that λ in Eq. (1) is the constant used in the probability $p(f_{uv})$ that a user follows another user v and τ in Eq. (16) is the regularization constant to prevent from overfitting in matrix factorization. The tested values of each constant are 0.1, 10^{-3} and 10^{-6} for λ, and 1, 5 and 10 for τ. With varying K from 1 to 50, we plotted the precision-of-Ks of *TWILITE-PR-F* in Fig. 15 with each setting of λ and τ. Both graphs show that we can obtain the best recommendation by estimating the model parameters with $\lambda=10^{-3}$ and $\tau=5$.

8. Conclusion

In this paper, we proposed *TWILITE*, a recommendation system for Twitter using probabilistic modeling based on latent Dirichlet allocation which can recommend top-K users to follow and top-K tweets to read for a user. With our novel probabilistic model, we improved the performance of recommendations for Twitter by exploiting both the friend relationships between users and the contents of tweet messages posted by users. Our generative model can represent a realistic process of posting tweet messages by generalizing an LDA model as well as the process of connecting to friends by utilizing a matrix factorization. We next developed a variational EM algorithm *TWILITE-VEM* to learn our model parameters. We also presented the ranking algorithms to recommend top-K followees and top-K tweets to a user based on the estimated model parameters. Our performance study with real-life data sets showed the effectiveness of *TWILITE* and the accuracy of the personalized recommendation methods.

Acknowledgment

This research was supported by Basic Science Research Program (No. NRF-2009-0078828) and Next-Generation Information Computing Development Program (No. NRF-2012M3C4A7033342) which were funded by the National Research Foundation of Korea (NRF) of the Ministry of Science, ICT & Future Planning (MSIP). It was also supported by Information Technology Research Center (ITRC) Support Program through the National IT Industry Promotion Agency (NIPA) of MSIP (No. NIPA-2013-H0301-13-4009).

Appendix A. The lower bound $\mathbb{F}$ of log-likelihood log $\mathbb{L}$

Introducing the approximate parameters δ_{uj}, ϕ_{uj}, a_u, b_k, c_u and ρ in Eq. (6), we obtain the lower bound of log-likelihood $\mathbb{F}(\delta,\phi,a,b,c,\rho)$ in Eq. (5) as follows:

$$\mathbb{F}(\delta,\phi,a,b,c,\rho) = \sum_{u=1}^{|U|}\left\{\log\Gamma(\alpha_\bullet)-\sum_{z=1}^{|Z|}\log\Gamma(\alpha_z)+\sum_{z=1}^{|Z|}(\alpha_z-1)(\Psi(a_{uz})-\Psi(a_{u\bullet}))\right\}$$

$$+\sum_{z=1}^{|Z|}\left\{\log\Gamma(\beta_\bullet)-\sum_{t=1}^{|V|}\log\Gamma(\beta_t)+\sum_{t=1}^{|V|}(\beta_t-1)(\Psi(b_{zt})-\Psi(b_{z\bullet}))\right\}$$
$$+\sum_{u=1}^{|U|}\left\{\log\Gamma(\gamma_{u\bullet})-\sum_{\ell=1}^{|F_u|}\log\Gamma(\gamma_{u\ell})+\sum_{\ell=1}^{|F_u|}(\gamma_{u\ell}-1)(\Psi(c_{u\ell})-\Psi(c_{u\bullet}))\right\}$$
$$+\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|}\sum_{z=1}^{|Z|}\sum_{v=1}^{|U|}I(f_{u\ell}=v)\phi_{uj}(z)\delta_{uj}(\ell)(\Psi(a_{vz})-\Psi(a_{v\bullet}))$$
$$+\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|}\sum_{t=1}^{|V|}\sum_{z=1}^{|Z|}I(w_{uj}=v_t)\phi_{uj}(z)(\Psi(b_{kt})-\Psi(b_{z\bullet}))$$
$$+\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|}\delta_{uj}(\ell)(\Psi(c_{u\ell})-\Psi(c_{u\bullet}))$$
$$+\sum_{u=1}^{|U|}\sum_{\ell=1}^{|F_u|}\sum_{v=1}^{|U|}I(f_{u\ell}=v)\rho_{uv}\{\Psi(c_{u\ell})-\Psi(c_{u\bullet})\}+\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}I(v\notin F_u)\rho_{uv}\log\lambda$$
$$-\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}\rho_{uv}(f_{uv}-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2-\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}\rho_{uv}\log\zeta_u-\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}\rho_{uv}\log\rho_{uv}$$
$$-\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|}\sum_{z=1}^{|Z|}\phi_{uj}(z)\log\phi_{uj}(z)-\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|}\delta_{uj}(\ell)\log\delta_{uj}(\ell)$$
$$-\sum_{u=1}^{|U|}\left\{\log\Gamma(a_{u\bullet})-\sum_{z=1}^{|Z|}\log\Gamma(a_{uz})+\sum_{z=1}^{|Z|}(a_{uz}-1)(\Psi(a_{uz})-\Psi(a_{u\bullet}))\right\}$$
$$-\sum_{z=1}^{|Z|}\left\{\log\Gamma(b_{z\bullet})-\sum_{t=1}^{|V|}\log\Gamma(b_{zt})+\sum_{t=1}^{|V|}(b_{zt}-1)(\Psi(b_{zt})-\Psi(b_{z\bullet}))\right\}$$
$$-\sum_{u=1}^{|U|}\left\{\log\Gamma(c_{u\bullet})-\sum_{\ell=1}^{|F_u|}\log\Gamma(c_{u\ell})+\sum_{\ell=1}^{|F_u|}(c_{u\ell}-1)(\Psi(c_{u\ell})-\Psi(c_{u\bullet}))\right\} \tag{A.1}$$

Appendix B. Derivations of Eqs. (9)–(14)

In the variational E-step, we calculate the variational parameters maximizing the lower bound of $\mathbb{F}$ in Eq. (A.1). In this section, we derive the update equations for the variational parameters $\phi_{uj}(z)$, $\delta_{uj}(\ell)$, a_{uz}, b_{zt}, $c_{u\ell}$ and ρ_{uv} by using the method of Lagrange multipliers [3].

Eq. (9): We first compute the derivative $\mathbb{F}$ with $\phi_{uj}(z)$ as follows:

$$\frac{\partial\mathbb{F}(\phi_{uj}(z))}{\partial\phi_{uj}(z)}=\sum_{v=1}^{|U|}\sum_{\ell=1}^{|F_u|}I(f_{u\ell}=v)\delta_{uj}(\ell)(\Psi(a_{vz})-\Psi(a_{v\bullet}))+\left(\Psi\left(b_{zw_{uj}}\right)-\Psi(b_{z\bullet})\right)-\log\phi_{uj}(k)-1-\xi_{\phi_{uj}}$$

where $\xi_{\phi_{uj}}$ represents the Lagrange multiplier for the constraint $\sum_{z=1}^{|Z|}\phi_{uj}(z)=1$. Then, by setting the above derivative to zero, we obtain the following formula for $\phi_{uj}(z)$ in Eq. (9):

$$\phi_{uj}(z)\propto\exp\left[\sum_{v=1}^{|U|}\sum_{\ell=1}^{|F_u|}I(f_{u\ell}=v)\delta_{uj}(\ell)(\Psi(a_{vk})-\Psi(a_{v}\bullet))+(\Psi(b_{k,w_{uj}})-\Psi(b_k\bullet))\right]$$

Eq. (10): Similarly, we next calculate the derivative of $\mathbb{F}$ with $\delta_{uj}(\ell)$ as shown below:

$$\frac{\partial\mathbb{F}(\delta_{uj}(\ell))}{\partial\delta_{uj}(\ell)}=\sum_{v=1}^{|U|}I(f_{u\ell}=v)\sum_{z=1}^{|Z|}\phi_{vj}(z)(\Psi(a_{vz})-\Psi(a_{v\bullet}))+(\Psi(c_{u\ell})-\Psi(c_{u\bullet}))-\log\delta_{uj}(\ell)-1-\xi_{\delta_{uj}}$$

where $\xi_{\delta_{uj}}$ denotes the Lagrange multiplier for the constraint $\sum_{\ell=1}^{|F_u|}\delta_{uj}(\ell)=1$. Setting the derivative equal to zero leads to the following equation for $\delta_{uj}(\ell)$:

$$\delta_{uj}(\ell)\propto\exp\left[\sum_{v=1}^{|U|}\sum_{\ell=1}^{|F_u|}I(f_{u\ell}=v)\sum_{z=1}^{|Z|}\phi_{vj}(z)(\Psi(a_{vz})-\Psi(a_{v\bullet}))+(\Psi(c_{u\ell})-\Psi(c_{u\bullet}))\right]$$

Eq. (11): We compute the derivative of $\mathbb{F}$ with ρ_{uv}, which is a variational parameter for the posterior distribution $p(f_{uv})$ in Eq. (1), as follows:

$$\frac{\partial\mathbb{F}(\rho_{uv})}{\partial\rho_{uv}}=\begin{cases}\sum_{\ell=1}^{|F_u|}I(f_{u\ell}=v)(\Psi(c_{u\ell})-\Psi(c_{u\bullet}))-(1-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2-\log\zeta_u-\log\rho_{uv}-1 & \text{if } v\in F_u\\ \log\lambda-(0-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2-\log\zeta_u-\log\rho_{uv}-1 & \text{otherwise}\end{cases}$$

We set it to zero and obtain the equation to update ρ_{uv} in Eq. (11) as

$$\rho_{uv} \propto \begin{cases} \exp\left[\sum_{\ell=1}^{|F_u|} I(f_{u\ell}=v)(\Psi(c_{u\ell})-\Psi(c_{u\bullet}))-(1-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2\right] & \text{if } v\in F_u \\ \exp[\log\lambda-(0-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2] & \text{otherwise} \end{cases}$$

Eq. (12): We next take the derivative of $\mathbb{F}$ with regard to each of a_{uz}, which is a variational parameter for the posterior distribution $\overrightarrow{\theta_u}$, as shown below:

$$\begin{aligned}\frac{\partial \mathbb{F}(a_{uz})}{\partial a_{uz}} &= \Psi'(a_{uz})\left(\alpha_{uz}+\sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|} I(f_{u\ell}=u)\phi_{uj}(z)\delta_{uj}(\ell)+\sum_{v\in R_u}\sum_{j=1}^{|D_v|}\sum_{\ell=1}^{|F_v|} I(f_{v\ell}=u)\phi_{vj}(z)\delta_{vj}(\ell)-a_{uz}\right)\\ &+\Psi'(a_{u\bullet})\sum_{z'=1}^{|Z|}\left(\alpha_{uz'}+\sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|} I(f_{u\ell}=u)\phi_{uj}(z')\delta_{uj}(\ell)+\sum_{v\in R_u}\sum_{j=1}^{|D_v|}\sum_{\ell=1}^{|F_v|} I(f_{v\ell}=u)\phi_{vj}(z')\delta_{vj}(\ell)-a_{uz'}\right)\end{aligned}$$

where $\Psi'(x)$ denotes the first derivative of the function $\Psi(x)$ and R_u is the set of follower's IDs of u. We set the derivative equal to zero and then, obtain

$$a_{uz}=\alpha_{uz}+\sum_{j=1}^{|D_u|}\sum_{\ell=1}^{|F_u|} I(f_{u\ell}=u)\phi_{uj}(z)\delta_{uj}(\ell)+\sum_{v\in R_u}\sum_{j=1}^{|D_v|}\sum_{\ell=1}^{|F_v|} I(f_{v\ell}=u)\phi_{vj}(z)\delta_{vj}(\ell)$$

Eq. (13): The derivative of $\mathbb{F}$ with b_{zt}, which is a variational parameter for the posterior distribution $\overrightarrow{\varphi_z}$, can be calculated as

$$\frac{\partial \mathbb{F}(b_{zt})}{\partial b_{zt}} = \Psi'(b_{zt})\left(\beta_{zt}+\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|} I(w_{uj}=t)\phi_{uj}(z)\delta_{uj}(\ell)-b_{zt}\right) \quad +\Psi'(b_{z\bullet})\sum_{t'=1}^{|V|}\left(\beta_{zt'}+\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|} I(w_{uj}=t')\phi_{uj}(z)\delta_{uj}(\ell)-b_{zt'}\right)$$

We set the above derivative to zero and get the following equation:

$$b_{zt}=\beta_{zt}+\sum_{u=1}^{|U|}\sum_{j=1}^{|D_u|} I(w_{uj}=t)\phi_{uj}(z)\delta_{uj}(\ell)$$

Eq. (14): We finally calculate the derivative of $\mathbb{F}$ with $c_{u\ell}$, which is a variational parameter for the posterior distribution $\overrightarrow{\pi_u}$, as

$$\frac{\partial \mathbb{F}(c_{u\ell})}{\partial c_{u\ell}} = \Psi'(c_{u\ell})\left(\gamma_{u\ell}+\sum_{j=1}^{|D_u|}\delta_{uj}(\ell)-c_{u\ell}\right)+\Psi'(c_{u\bullet})\sum_{\ell'=1}^{|F_u|}\left(\gamma_{u\ell'}+\sum_{j=1}^{|D_u|}\delta_{uj}(\ell')-c_{u\ell'}\right)$$

By setting the derivative equal to zero, we obtain the equation for updating $c_{u\ell}$ as $c_{u\ell}=\gamma_{u\ell}+\sum_{j=1}^{|D_u|}\delta_{uj}(\ell)$.

Appendix C. Derivations of Eqs. (17) and (18)

In the variational M-step of *TWITOBI-LDA-VEM*, we compute the model parameters $\mathbb{X}$ and $\mathbb{Y}$ which maximize $\mathbb{F}$ in Eq. (A.1). In $\mathbb{F}$, the terms related to $\mathbb{X}$ and $\mathbb{Y}$ are

$$\begin{aligned}\mathbb{F}(\mathbb{X},\mathbb{Y}) &= -\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}\rho_{uv}(f_{uv}-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2-\sum_{u=1}^{|U|}S(\rho_u)\log\zeta_u \quad = -\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}\rho_{uv}(f_{uv}-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2\\ &-\sum_{u=1}^{|U|}S(\rho_u)\log\sum_{v=1}^{|U|}[I(v\in F_u)\pi_{uv}e^{-(1-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2}+I(v\notin F_u)\lambda e^{-(0-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2}]\end{aligned} \tag{C.1}$$

where $S(\rho_u)$ is $\sum_{v=1}^{|U|}\rho_{uv}$.

Given ρ_{uv} and π_{uv}, we utilize an EM algorithm to calculate $\mathbb{X}$ and $\mathbb{Y}$ that maximize $\mathbb{F}$. We first approximate $\mathbb{F}$ using Jensen's inequality by introducing a free distribution q_{uv}, satisfying $\sum_{v=1}^{|U|}q_{uv}=1$, as

$$\begin{aligned}\mathbb{F}(\mathbb{X},\mathbb{Y}) &= -\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}\rho_{uv}(f_{uv}-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2\\ &-\sum_{u=1}^{|U|}S(\rho_u)\log\sum_{v=1}^{|U|}\frac{q_{uv}\left[I(v\in F_u)\pi_{uv}e^{-(1-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2}+I(v\notin F_u)\lambda e^{-(0-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2}\right]}{q_{uv}}\\ &\le -\sum_{u=1}^{|U|}\sum_{v=1}^{|U|}\rho_{uv}(f_{uv}-\overrightarrow{x_u}^T\cdot\overrightarrow{y_v})^2\end{aligned} \tag{C.2}$$

$$-\sum_{u=1}^{|U|} S(\rho_u) \sum_{v=1}^{|U|} q_{uv} \left[I(v \in F_u) \log \frac{\pi_{uv} e^{-(1-\vec{x_u}^T \cdot \vec{y_v})^2}}{q_{uv}} + I(v \notin F_u) \log \frac{\lambda e^{-(0-\vec{x_u}^T \cdot \vec{y_v})^2}}{q_{uv}} \right] \tag{C.3}$$

$$= -\sum_{u=1}^{|U|} \sum_{v=1}^{|U|} (\rho_{uv} - S(\rho_u) q_{uv})(f_{uv} - \vec{x_u}^T \cdot \vec{y_v})^2 - \sum_{u=1}^{|U|} S(\rho_u) \sum_{v=1}^{|U|} q_{uv}[I(\in F_u) \log \pi_{uv} + I(v \notin F_u) \log \lambda - \log q_{uv}] \tag{C.4}$$

E-step: E-step is to minimize the difference between $\mathbb{F}$ in Eq. (C.2) and the approximate one in Eq. (C.3). If we take the derivative of $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ and set it to zero, we can obtain the following equation for updating q_{uv}:

$$q_{uv} = \frac{1}{\xi_u} \left[I(v \in F_u) \log \pi_{uv} e^{-(1-\vec{x_u}^T \cdot \vec{y_v})^2} + I(v \notin F_u) \log \lambda \, e^{-(0-\vec{x_u}^T \cdot \vec{y_v})^2} \right] \tag{C.5}$$

where ξ_u is $\sum_{v=1}^{|U|}[I(v \in F_u) \log \pi_{uv} e^{-(1-\vec{x_u}^T \cdot \vec{y_v})^2} + I(v \notin F_u) \log \lambda e^{-(0-\vec{x_u}^T \cdot \vec{y_v})^2}]$. Note that the expression of q_{uv} in Eq. (C.5) is exactly the same as that of $p(f_{uv} | \vec{\pi_u}, \vec{x_u}, \vec{y_v})$ in Eq. (1).

M-step: Given free distributions q_{uv}, we now maximize $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ with respect to $\vec{x_u}$ and $\vec{y_u}$. Note that computing $\mathbb{X}$ and $\mathbb{Y}$ maximizing $\mathbb{F}(\mathbb{X}, \mathbb{Y})$ in Eq. (C.4) is exactly equivalent to the minimization in weighted matrix factorization proposed in [18]. Thus, we borrow its equations for updating $\vec{x_u}$ and $\vec{y_u}$ from [18] as follows:

$$\vec{x_u} = (Y^T \cdot P_u \cdot Y + \omega I)^{-1} \cdot Y^T \cdot P_u \cdot \vec{f_u} \tag{C.6}$$

$$\vec{y_u} = (X^T \cdot Q_u \cdot X + \omega I)^{-1} \cdot X^T \cdot Q_u \cdot \vec{f_u} \tag{C.7}$$

where X and Y are the $|U| \times |U|$ matrices such that the uth column of each matrix is $\vec{x_u}$ and $\vec{y_u}$ respectively. Furthermore, in the above equations, P_u is a $|U| \times |U|$ diagonal matrix such that the value of the vth row and the vth column is $\rho_{uv} - S(\rho_u) q_{uv}$ with a user v whom u follows (i.e., $v \in F_u$). Similarly, Q_u is also a $|U| \times |U|$ diagonal matrix such that the value of the vth row and the vth column is $\rho_{uv} - S(\rho_u) q_{uv}$ with a user v who follows u (i.e., $v \in R_u$). Since q_{uv} in Eqs. (C.6) and (C.7) is $p(f_{uv})$, we can obtain Eqs. (17) and (18) in Section 5.

References

[1] E.M. Airoldi, D.M. Blei, S.E. Fienberg, E.P. Xing, Mixed membership stochastic blockmodels, J. Mach. Learn. Res. 9 (2008) 1981–2014.
[2] C.G. Akcora, B. Carminati, E. Ferrari, Privacy in social networks: how risky is your social graph? in: ICDE, 2012, pp. 9–19.
[3] D.P. Bertsekas, Nonlinear Programming, second ed., Cambridge, 1999.
[4] M. Bianchini, M. Gori, F. Scarselli, Inside pagerank, ACM Trans. Internet Technol. 5 (1) (2005) 92–128.
[5] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[6] D. Cai, Q. Mei, J. Han, C. Zhai, Modeling hidden topics on document manifold, in: CIKM, 2008, pp. 911–920.
[7] X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y.S. Kim, P. Compton, A. Mahidadia, Learning collaborative filtering and its application to people to people recommendation in social networks, in: ICDM, 2010, pp. 743–748.
[8] J. Chang, D.M. Blei, Relational topic models for document networks, J. Mach. Learn. Res. Proc. Track 5 (2009) 81–88.
[9] J. Chen, R. Nairn, L. Nelson, M.S. Bernstein, E.H. Chi, Short and tweet: experiments on recommending content from information streams, in: CHI, 2010.
[10] CiteULike, ⟨http://www.citeulike.org⟩.
[11] H. Deng, M.R. Lyu, I. King, A generalized Co-HITS algorithm and its application to bipartite graphs, in: KDD, 2009, pp. 239–248.
[12] M. Deshpande, G. Karypis, Item-based top-recommendation algorithms, ACM Trans. Inf. Syst. 22 (1) (2004) 143–177.
[13] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proc. Natl. Acad. Sci. USA 101 (Suppl. 1) (2004) 5228–5235.
[14] J. Hannon, M. Bennett, B. Smyth, Recommending twitter users to follow using content and collaborative filtering approaches, in: RecSys, 2010.
[15] T. Haveliwala, S. Kamvar, G. Jeh, An analytical comparison of approaches to personalizing pagerank, in: Preprint, 2003.
[16] T. Hofmann, Probabilistic latent semantic indexing, in: SIGIR, 1999.
[17] T. Hofmann, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004).
[18] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: ICDM, 2008, pp. 263–272.
[19] K. Humphreys, D. Titterington, Approximate Bayesian inference for simple mixtures, in: COMPSTAT, Physica-Verlag HD, 2000, pp. 331–336.
[20] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: RecSys, 2010, pp. 135–142.
[21] M.I. Jordan, Z. Ghahramani, T. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models, Mach. Learn. 37 (2) (1999) 183–233.
[22] Y. Kim, K. Shim, TWITOBI: a recommendation system for Twitter using probabilistic modeling, in: ICDM, 2011, pp. 340–349.
[23] Y. Kim, Y. Park, K. Shim, DIGTOBI: a recommendation system for Digg articles using probabilistic modeling, in: WWW, 2013, pp. 691–702.
[24] I. Konstas, V. Stathopoulos, J.M. Jose, On social networks and collaborative recommendation, in: SIGIR, 2009, pp. 195–202.
[25] Y. Koren, R.M. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, IEEE Comput. 42 (8) (2009) 30–37.
[26] Y. Liu, A. Niculescu-Mizil, W. Gryc, Topic-link LDA: joint models of topic and author community, in: ICML, 2009, p. 84.
[27] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: WSDM, 2011, pp. 287–296.
[28] B. Mehta, T. Hofmann, W. Nejdl, Robust collaborative filtering, in: RecSys, 2007.
[29] A. Popescul, L.H. Ungar, D.M. Pennock, S. Lawrence, Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, in: UAI, 2001, pp. 437–444.
[30] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: NIPS, 2007.
[31] G. Shani, D. Heckerman, R.I. Brafman, An MDP-based recommender system, J. Mach. Learn. Res. 6 (2005) 1265–1295.
[32] L. Si, R. Jin, Flexible mixture model for collaborative filtering, in: ICML, 2003, pp. 704–711.
[33] G. Strang, Linear Algebra and its Applications, Brooks Cole, Philadelphia, PA, 1988.
[34] Twitter, Api document, ⟨http://dev.twitter.com/doc⟩, 2011.
[35] B.-Q. Vuong, E.-P. Lim, A. Sun, M.-T. Le, H.W. Lauw, On ranking controversies in Wikipedia: models and evaluation, in: WSDM, 2008.
[36] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: KDD, 2011, pp. 448–456.
[37] Wikipedia, Variational algorithms for approximate Bayesian inference (Ph.D. thesis), University of Cambridge, 2003.

[38] Wikipedia, Dirichlet Distribution, ⟨http://en.wikipedia.org/wiki/Dirichlet_distribution⟩, 2013.
[39] Wikipedia, Jensen's Inequality, ⟨http://en.wikipedia.org/wiki/Jensen's_inequality⟩, 2013.
[40] S.-H. Yang, A.J. Smola, B. Long, H. Zha, Y. Chang, Friend or frenemy? Predicting signed ties in social networks, in: SIGIR, 2012, pp. 555–564.
[41] H. Yildirim, M.S. Krishnamoorthy, A random walk method for alleviating the sparsity problem in collaborative filtering, in: RecSys, 2008.
[42] W. Zeng, L. Chen, Heterogeneous data fusion via matrix factorization for augmenting item, group and friend recommendations, in: SAC, 2013, pp. 237–244.