Joint Topic-Semantic-aware Social Recommendation for Online Voting

Hongwei Wang[1,2], Jia Wang[2], Miao Zhao[2], Jiannong Cao[2], Minyi Guo[1*]

[1]Shanghai Jiao Tong University, [2]The Hong Kong Polytechnic University

wanghongwei55@gmail.com,{csjiawang,csmiaozhao,csjcao}@comp.polyu.edu.hk,guo-my@cs.sjtu.edu.cn

ABSTRACT

Online voting is an emerging feature in social networks, in which users can express their attitudes toward various issues and show their unique interest. Online voting imposes new challenges on recommendation, because the propagation of votings heavily depends on the structure of social networks as well as the content of votings. In this paper, we investigate how to utilize these two factors in a comprehensive manner when doing voting recommendation. First, due to the fact that existing text mining methods such as topic model and semantic model cannot well process the content of votings that is typically short and ambiguous, we propose a novel Topic-Enhanced Word Embedding (TEWE) method to learn word and document representation by jointly considering their topics and semantics. Then we propose our Joint Topic-Semantic-aware social Matrix Factorization (JTS-MF) model for voting recommendation. JTS-MF model calculates similarity among users and votings by combining their TEWE representation and structural information of social networks, and preserves this topic-semantic-social similarity during matrix factorization. To evaluate the performance of TEWE representation and JTS-MF model, we conduct extensive experiments on real online voting dataset. The results prove the efficacy of our approach against several state-of-the-art baselines.

KEYWORDS

Online voting; recommender systems; topic-enhanced word embedding; matrix factorization

1 INTRODUCTION

Online voting [31] has recently become a popular function on social platforms, through which a user can share his opinion towards various interested subjects, ranging from livelihood issues to entertainment news. More advanced than simple like-dislike type of votings, some social networks, such as Weibo[1], have empowered users to run their own voting campaigns. Users can freely initiate votings on any topics of their own interests and customize voting options. These votings are visible to the friends of initiator, who can then choose to participate to make the votings further seen by
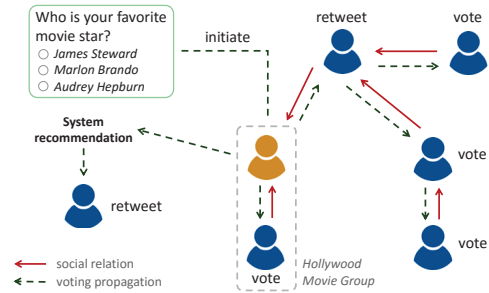
Fig. 1: Propagation scheme of online voting.

their friends or simply retweet the votings to their friends. In such a way, in addition to the system recommendation, a voting can widely propagate over the network along social paths. The voting propagation scheme is shown in Fig. 1.

Facing a large volume of diversified votings, a critical challenge is to present "right" votings to the "right" person. An effective recommender system is desired to be able to deal with information overload [2] by precisely locating what votings favor each user most, thus improves user experience and maximizes user engagement in votings. Such a recommender system can also benefit a variety of other online services such as personalized advertising [26], market research [11], public opinion analysis [17], etc.

Despite the great importance, there is little prior work considering recommending votings to users in social networks. The challenges are two-fold. First, the propagation of online votings relies heavily on the structure of social networks. A user can see the votings initiated, participated or retweeted by his followees, which implies that the user is more likely to be exposed to the votings that his friends are involved in. Moreover, in most social networks, a user can join different interest groups, which is another type of social structure that potentially affects users' voting behavior. Though several prior works [3, 6, 10, 12, 25, 29, 31, 32] have been proposed to leverage social network information in recommendation, it is still an open question how to comprehensively incorporate structural social information into the task of voting recommendation considering its propagation pattern. Second, users' interest in votings is strongly connected with voting content presented in question text (e.g., "Who is your favorite movie star?"). Topic model [1] is regarded as a possible approach to mine the voting interests through discovering the latent topic distribution of relevant voting text. However, the voting questions are typically short and lack sufficient topic information, leading to severe performance degradation of topic models. Alternatively, semantic analytics [21] can also possibly be used to mine voting interests through learning text representations. However, such semantic models typically represent each word using a single vector, making them indiscriminative

for homonymy and polysemy, which are especially common in voting questions (e.g., "Do you use apple products?" and "Do you peel apple before eating?"). In brief, these inherent defects of the above models limit their power in the scenario of social voting recommendation.

To address aforementioned challenges, in this paper, we propose a novel *Joint Topic-Semantic-aware Matrix Factorization* (JTS-MF) model for online voting recommendation. JTS-MF model considers social network structure and representation of voting content in a comprehensive manner. For social network structure, JTS-MF model fully encodes the information of social relationship and group affiliation into the objective function. We will further justify the usage of social network structure in Section 3. For representation of voting content, we propose a *Topic-Enhanced Word Embedding* (TEWE) method to build a multi-prototype word and document[2] representation, which jointly considers their topics and semantics. The key idea of TEWE is to enable each word to have different representations under different word topics and different documents. We will detail TEWE in Section 5. Once obtaining TEWE representation for each document, JTS-MF model combines them with the structural information of social networks to calculate the topic-semantic-social similarity among users and votings. The reason of calculating such similarity is that, inspired by Locally Linear Embedding [23], we try to preserve the similarity among users and votings during matrix factorization, as it contains abundant proximity information and can greatly benefit feature learning for users and votings. JTS-MF model is detailed in Section 6.

We conduct extensive investigation on JTS-MF with real online voting dataset. The experimental results in Section 7 demonstrate that JTS-MF model achieves substantial gains compared with baselines. The results also prove that TEWE is able to well combine topic and semantic information of texts and generates a better kind of document representation.

In summary, the contributions of this paper are as follows:

- We formally formulate the online voting recommendation problem, which has not been fully investigated yet.
- We indicate that user's voting behavior is highly correlated with social network structure by conducting thorough statistical measurements.
- We propose a novel Topic-Enhanced Word Embedding model to jointly consider topics and semantics of words and documents to learn their representation. TEWE takes advantages of both topic model and semantic model, and consequently learns more informative embeddings.
- We develop a novel matrix factorization based models, named JTS-MF, for online voting recommendation. JTS-MF is able to preserve the topic-semantic-social similarity among users and votings from original embedding space during learning process.
- We carry out extensive experiments on real online voting dataset, the results of which reveal that JTS-MF significantly outperforms baseline (variant) methods, say for example, surpassing basic matrix factorization model with

57%, 38% and 25% enhancement in terms of recall for top-1, top-5 and top-20 recommendation, respectively.

2 RELATED WORK

2.1 Recommender Systems

Roughly speaking, existing recommender systems can be categorized into three classes [2]: content-based, collaborative filtering, and hybrid methods. Content-based methods [14, 34] make use of user profiles or item descriptions as features for recommendation. Collaborative filtering methods [22, 25, 28, 31] use either explicit feedback (e.g., users' ratings on items) or implicit feedback (e.g., users' browsing records about items) data of user-item interactions to find user preference and make the recommendation. In addition, various models are incorporated into collaborative filtering, such as Support Vector Machine [30], Restricted Boltzmann Machine [24], and Stacked Denoising Auto Encoder [27]. Hybrid methods [9, 16] combine content-based and collaborative filtering models in many hybridization approaches, such as weighted, switching, cascade and feature combination or augmentation.

2.2 Social Recommendation

Traditional recommender systems are vulnerable to data sparsity problem and cold-start problem. To mitigate this issue, many approaches have been proposed to utilize social network information in recommender systems [3, 6, 10, 12, 25, 29, 31, 32]. For example, [12] represents a social network as a star-structured hybrid graph centered on a social domain which connects with other item domains to help improve the prediction accuracy. [10] investigates the seed selection problem for viral marketing that considers both effects of social influence and item inference for product recommendation. [29] studies the effects of strong and weak ties in social recommendation, and extends Bayesian Personalized Ranking model to incorporate the distinction of strong and weak ties. However, the above works only utilize users' social links without considering the topic and semantic information for mining the similarities among users and items, which we found quite helpful for voting recommendation tasks. Another difference between these works and ours is that we also take social group affiliation into consideration, which can further improve the performance of recommendation.

2.3 Topic and Semantic Language Models

Latent Dirichlet Allocation (LDA) [1] is a well-known generative topic model that learns the latent topic distributions for documents. LDA is widely used in sentiment analysis [20], aspects and opinions mining [33], and recommendation [5]. Word2vec [21] is generally recognized as a neural network model, which learns word representations that capture precise syntactic and semantic word relationships. Word2vec as well as associated Skip-Gram model are extensively used in document classification [15], dependency parser [4], and network embedding [8]. However, LDA and Word2vec are not directly applicable in the scenario of voting recommendation because the content of voting is usually short and ambiguous. As a combination, [18] tries to learn topical word embeddings based on both words and their topics. The difference between [18] and ours is that we also take topics of documents into consideration,

---

[2]In this paper, a "document" can be related to a voting, a user or a group. A voting document is the content of its question, a user document is formed by aggregating all the documents of votings he participates, and a group document is formed by aggregating all the documents of users who join the group.

Table 1: Basic Statistics of Weibo Dataset.

# users	1,011,389	# groups	299,077
# users with votings	525,589	# user-voting	3,908,024
# users with groups	723,913	# user-user	83,636,677
# votings	185,387	# user-group	5,643,534

which enables our model to learn a even more discriminative and informative representations for words and documents.

3 BACKGROUND AND DATA ANALYSIS

In this section, we briefly introduce the background of Weibo voting and present detailed analysis of Weibo voting dataset.

3.1 Background

Weibo is one of the most popular Chinese microblogging website launched by Sina corporation, which is akin to a hybrid of Twitter and Facebook platforms. Users on Weibo can follow each other, write tweets and share with his followers. Users can also join different groups based on their attributes (e.g., region) or interests of topics (e.g., career).

Voting[3] is one of the embedded features of Weibo. As of January 2013, more than 92 million users have participated in at least one voting and more than 2.2 million ongoing votings are available on Weibo every day. Any user can freely initiate, retweet and participate a voting campaign in Weibo. As shown in Fig. 1, votings can propagate in two ways. The first way is through social propagation: a user can see the voting initiated, retweeted or participated by his followees and potentially participates the voting. The second way is through Weibo voting recommendation list, which consists of popular votings and personalized recommendation for each user.

3.2 Data Measurements

Our Weibo voting dataset comes from the technical team of Sina Weibo, which contains detailed information about votings from November 2010 to January 2012, as well as other auxiliary information. Specifically, the dataset includes users' participation status on each voting[4], content of each voting, social connection between users, name and category of each group, and user-group affiliation.

3.2.1 Basic statistics. The basic statistics are summarized in Table 1. From Table 1 we can learn that, each user has 165.4 followers/followees, participates 3.9 votings, and joins 5.6 groups on average. If we only count users who participate at least one voting and users who join at least on group, the average number of votings and average number of joined groups of each user is 7.4 and 7.8, respectively. Fig. 2 depicts the distribution curves of the above statistics, where the meaning of each subfigure is given in the caption.

To get an intuitive understanding of whether user's voting behavior is correlated with his social relation and group affiliation, we conduct the following two sets of statistical experiments:

3.2.2 Correlation between the number of common votings of user pairs and the types of user pairs. We randomly select ten million user pairs from the set of all users, and count the average number
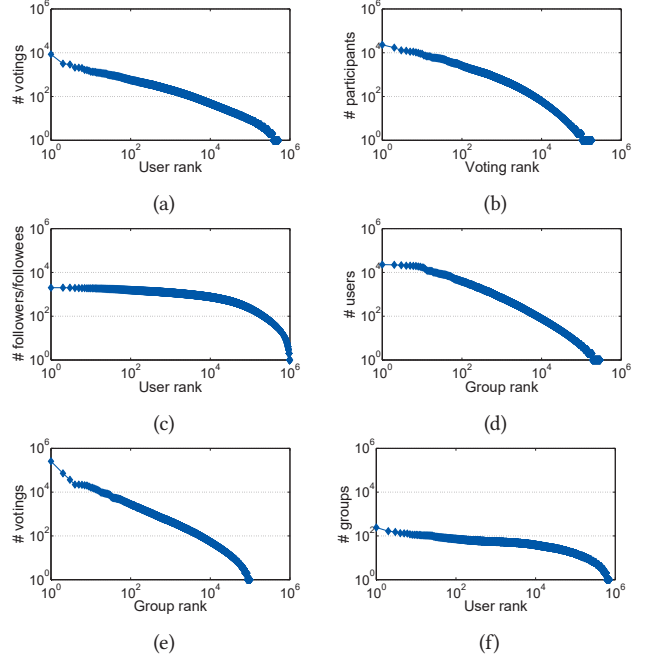
...

Fig. 2: (a) Distribution of the number of votings participated by a user; (b) Distribution of the number of participants of a voting; (c) Distribution of the number of followers/followees of a user; (d) Distribution of the number of users in a group; (e) Distribution of the number of votings (may contain duplicated votings) participated by all users in a group; (f) Distribution of the number of groups joined by a user.

of votings that the two users both participate under the following four circumstances: 1) one of the users follows the other in the pair, i.e., they are social-level friends; 2) the two users are in at least one common group, i.e., they are group-leven friends; 3) the two users are neither social-level friends nor group-level friends; 4) all cases. The results are plotted in Fig. 3a, which clearly shows the difference among these cases. In fact, the average number of common votings of social-level friends (3.54×10^{-4}) and group level friends (1.79×10^{-4}) are 17.4 and 8.8 times higher than that of "strangers" (2.04×10^{-5}). The results demonstrate that if two users are social-level or group-level friends, they are likely to participate more votings in common.

3.2.3 Correlation between the probability of two users being friends and whether they participate common voting. We first randomly select ten thousand votings from the set of all votings. For each sampled voting v_j, we calculate the probability that two of its participants are social-level or group-level friends, i.e., $p_j = \frac{\text{\# of social/group-level friends among participants of } v_j}{n_j \times (n_j-1)/2}$, where n_j is the number of v_j' participants. We calculate p_j over all sampled votings and plot the average result (blue bar) in Fig. 3b. For comparison, we also plot the result for randomly sampled set of users (green bar) in Fig. 3b. It is clear that if two users ever participated common voting, they are more likely to be social-level or group-level friends. In fact, probabilities of two users being social-level or group-level
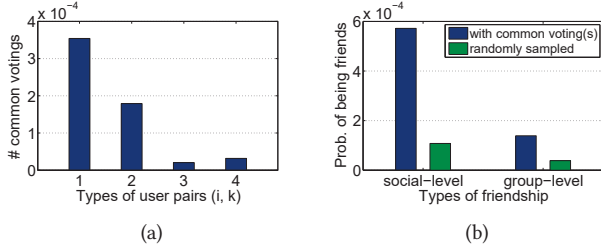
Fig. 3: (a) Average number of common votings participated by user u_i and u_k in four cases: 1. u_i follows/is followed by u_k; 2. u_i and u_k are in at least one common group; 3. u_i and u_k have no social-level and group-level relationship; 4. all cases; **(b)** Probability of two users being social-level or group-level friends in two cases: 1. they ever participated at least one common voting; 2. they are randomly sampled.

friends are raised by 5.3 and 3.6 times given the observation that they are with common voting.

The above two findings effectively prove the strong correlation between voting behavior and social network structure, which motivates us to take users' social relation and group affiliation into consideration when making voting recommendation.

4 PROBLEM FORMULATION

In this paper, we consider the problem of recommending Weibo votings to users. We denote the set of all users, the set of all votings, and the set of all groups by $\mathcal{U} = \{u_1, ..., u_N\}$, $\mathcal{V} = \{v_1, ..., v_M\}$, and $\mathcal{G} = \{G_1, ..., G_L\}$, respectively. Moreover, we model three types of relationship in Weibo platform: user-voting, user-user, and user-group relationship as follows:

(1) The user-voting relationship for u_i and v_j is defined as

$$I_{u_i, v_j} = \begin{cases} 1, & \text{if } u_i \text{ participates } v_j; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

(2) The user-user relationship for u_i and u_k is defined as

$$I_{u_i, u_k} = \begin{cases} 1, & \text{if } u_i \text{ follows } u_k; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

We further use $\mathcal{F}_i^+$ to denote the set of u_i's followees, and use $\mathcal{F}_i^-$ to denote the set of u_i's followers ("+" means "out" and "−" means "in").

(3) The user-group relationship for u_i and G_c is defined as

$$I_{u_i, G_c} = \begin{cases} 1, & \text{if } u_i \text{ joins } G_c; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Given the above sets of users and votings as well as three types of relationship, we aim to recommend a list of votings for each user, in which the votings are not participated by the user but may be interesting to him.

5 JOINT-TOPIC-SEMANTIC EMBEDDING

In this section, we explain how to learn the embeddings of users, votings, and groups in a joint topic and semantic way, and apply the embeddings to calculate similarities. We first introduce the methods of learning topic information and semantic information by LDA and

Skip-Gram models, respectively, and propose our method which combines these two models to learn more powerful embeddings.

5.1 Topic Distillation

In this subsection, we introduce how to profile users, votings, and groups in terms of topic interest distribution by performing topic distillation on the associated textual content information.

In general, LDA is a popular generative model to discover latent topic information from a collection of documents [1]. In LDA, each document d is represented as a multinomial distribution Θ_d over a set of topics, and each topic z is also represented as a multinomial distribution Φ_z over a number of words. Subsequently, each word position l in document d is assigned a topic $z_{d,l}$ according to Θ_d, and the word $w_{d,l}$ is finally generated according to $\Phi_{z_{d,l}}$. By LDA approach, the topic distribution for each document and the topic assignment for each word can be obtained, which would be utilized later in our proposed model.

Here, we discuss how to apply LDA in the scenario of Weibo voting. According to the Weibo voting dataset, each voting v_j associates a sentence of question, which can be regarded as document d_{v_j} [5]. The document d_{u_i} for user u_i can thus be formed by aggregating the content of all votings he participates, i.e., $d_{u_i} = \cup\{d_{v_j} | I_{u_i, v_j} = 1\}$, and the document d_{G_c} for group G_c is formed by aggregating documents of all its members, i.e., $d_{G_c} = \cup\{d_{u_i} | I_{u_i, G_c} = 1\}$. Note that though our target is to learn the topic distributions of all users, votings, and groups, it is inadvisable to train LDA model on d_{u_i}'s and d_{v_j}'s because: (1) the entitled sentence associated with a single voting is typically short-presented and topic-ambiguous; (2) even with user-level voting content aggregation, some documents of inactive users are not long enough to accurately extract the authentic topic distribution, yet showing relatively flat distribution over all the topics. Therefore, we choose to feed group-level aggregated documents d_{G_c}'s to LDA model as training samples. The process of group-level voting content aggregation will cover all the content the affiliated users are interested in and help better identify their interests in terms of voting topic.

Denote $\text{Dir}(\boldsymbol{\alpha})$ as the Dirichlet prior of Θ_d, and $\text{Dir}(\boldsymbol{\beta})$ as the Dirichlet prior of Φ_z. Given $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the joint distribution of document-topic distributions Θ, topic-word distributions Φ, topics of words $\boldsymbol{z}$, and a set of words $\boldsymbol{w}$ is

$$p(\Theta, \Phi, \boldsymbol{z}, \boldsymbol{w} | \boldsymbol{\alpha}, \boldsymbol{\beta})$$
$$= \prod_z p(\Phi_z | \boldsymbol{\beta}) \cdot \prod_d \left(p(\Theta_d | \boldsymbol{\alpha}) \prod_l \left(p(z_{d,l} | \Theta_d) p(w_{d,l} | \Phi_{z_{d,l}}) \right) \right), \quad (4)$$

where d traverses all group-level aggregated documents. In general, it is computationally intractable to directly maximize the joint likelihood in Eq. (4), thus Gibbs Sampling [7] is usually applied to estimate the posterior probability $p(\boldsymbol{z} | \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ and solve Θ, Φ iteratively. Denote $\theta_d^{(z)}$ the z-th component of Θ_d, and $\phi_z^{(w)}$ the w-th component of Φ_z. With the sampling results, Θ_d and Φ_z can be estimated as:

$$\theta_d^{(z)} = \left(n_d^{(z)} + \alpha^{(z)} \right) / \left(\sum_z (n_d^{(z)} + \alpha^{(z)}) \right), \qquad z = 1, ..., Z,$$
$$\phi_z^{(w)} = \left(n_z^{(w)} + \beta^{(w)} \right) / \left(\sum_w (n_z^{(w)} + \beta^{(w)}) \right), \qquad w = 1, ..., V, \quad (5)$$

---

[5]d_{v_j} is segmented by Jieba (https://github.com/fxsjy/jieba) and all stop words are removed.

where $\alpha^{(z)}$ is the z-th component of $\boldsymbol{\alpha}$, $\beta^{(w)}$ is the w-th component of $\boldsymbol{\beta}$, $n_d^{(z)}$ is the observation counts of topic z for document d, $n_z^{(w)}$ is the frequency of word w assigned as topic z, Z is the number of topics and V is vocabulary size.

So far, we have obtained the topic assignment for each word and topic distribution for each group. Topic distributions for users and votings can thus be inferred by using the learned model and Gibbs Sampling, which is similar to the calculation of $\theta_d^{(z)}$ in Eq. (5).

5.2 Semantic Distillation

In this subsection, we introduce how to profile users, votings, and groups in terms of semantic information. Word embedding, which represents each word using a vector, is widely used to capture semantic information of words. Skip-Gram model is a well-known framework for word embedding, which finds word representation that are useful for predicting surrounding words in a document given a target word in a sliding window [21]. More formally, given a word sequence $D = \{w_1, w_2, \ldots, w_T\}$, the objective function of Skip-Gram is to maximize the average log probability

$$\mathcal{L}(D) = \frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-k \leq c \leq k \\ c \neq 0}} \log p(w_{t+c}|w_t), \qquad (6)$$

where k is the training context size of the target word, which can be a function of the centered work w_t. The basic Skip-Gram formulation defines $p(w_i|w_t)$ using the softmax function as follows:

$$p(w_i|w_t) = \frac{\exp(\mathbf{w}_i^\top \mathbf{w}_t)}{\sum_{w \in V} \exp(\mathbf{w}^\top \mathbf{w}_t)}, \qquad (7)$$

where $\mathbf{w}_i$ and $\mathbf{w}_t$ are the vector representation of context word w_i and target word w_t, respectively, and V is the vocabulary. To avoid traversing the entire vocabulary, hierarchical softmax or negative sampling are used in general during learning process [21].

5.3 Topic-Enhanced Word Embedding

In this subsection, we propose a joint topic and semantic learning model, named *Topic-Enhanced Word Embedding* (TEWE), to analyze documents of users, votings, and groups. The motivation of proposed TEWE is based on the following two observations: (1) The voting content typically involves short texts. Even we infer the topic distribution for each voting based on the learned topic-word distribution from group-level aggregated documents, it is still topic-ambiguous to some extent. (2) The Skip-Gram model for word embedding assumes that each word always preserves a single vector, which sometimes is indiscriminate under different circumstances due to the homonymy and polysemy. Therefore, the basic idea of TEWE is to preserve topic information of documents and words when measuring the interaction between target word w_t and context word w_i. In this way, a word with different associated topics has different embeddings, and a word in documents with different topics has different embeddings, too.

Specifically, rather than solely using the target word w to predict context words in Skip-Gram, TEWE also jointly utilizes z_w, the topic of the word in a document, as well as z_w^d, the most likely topic of the document that the word belongs to. Recall that in Section 5.1, we have obtained the topic of each word z_w and topic
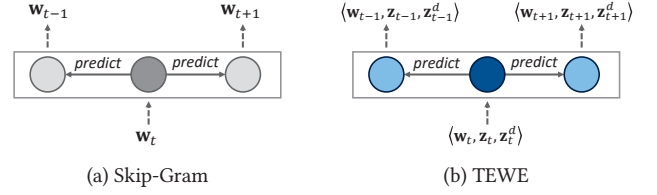

(a) Skip-Gram   (b) TEWE

Fig. 4: Comparison between Skip-Gram and TEWE. The gray circles in (a) indicate the embeddings of original words, while the blue circles in (b) indicate the TEWE representation of pseudo words, which preserves semantic and topic information of words and documents.

distributions of each document $\boldsymbol{\Theta}_d$, thus z_w^d can be calculated as $z_w^d = \arg\max_z \theta_d^{(z)}$, where $\theta_d^{(z)}$ is the probability that document d belongs to topic z, as introduced in Eq. (5). TEWE regards each word-topics triplet $\langle w, z_w, z_w^d \rangle$ as a pseudo word and learns a unique vector $\mathbf{w}^{z, z^d}$ for it. The objective function of TEWE is as follows:

$$\mathcal{L}(D) = \frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-k \leq c \leq k \\ c \neq 0}} \log p(\langle w_{t+c}, z_{t+c}, z_{t+c}^d \rangle | \langle w_t, z_t, z_t^d \rangle), \quad (8)$$

where $p(\langle w_i, z_i, z_i^d \rangle | \langle w_t, z_t, z_t^d \rangle)$ is a softmax function as

$$p(\langle w_i, z_i, z_i^d \rangle | \langle w_t, z_t, z_t^d \rangle) = \frac{\exp\left(\mathbf{w}_i^{z, z^d \top} \mathbf{w}_t^{z, z^d}\right)}{\sum_{\langle w, z, z^d \rangle \in \langle V, Z, Z \rangle} \exp\left(\mathbf{w}^{z, z^d \top} \mathbf{w}_t^{z, z^d}\right)}. \qquad (9)$$

The comparison between Skip-Gram and TEWE is shown in Fig. 4. Instead of solely utilizing the target and context words as in Skip-Gram, TEWE further preserves word topic and document topic along with these words, and incorporates both topic and semantic information in embedding learning.

Once obtaining TEWE representation for each pseudo word, the representation of each document can be correspondingly derived by aggregating the embeddings of its containing words weighted by *term frequency-inverse document frequency* (TF-IDF) coefficient. Specifically, for each document d, its TEWE can be calculated as

$$\mathbf{e}_d = \sum_{w \in d} \text{TF-IDF}(w, d) \cdot \mathbf{w}^{z, z^d}, \qquad (10)$$

where $\text{TF-IDF}(w, d)$ is the product of the raw count of w in d and the logarithmically scaled inverse fraction of the documents that contains w, i.e., $\text{TF-IDF}(w, d) = f_{w,d} \cdot \log \frac{|D|}{|d \in D: w \in d|}$ (D is the set of all documents). TEWE document representations can be used in measuring inter-document similarities. For example, the similarity of two user documents d_{u_i} and d_{u_k} can be calculated as the cosine similarity between their TEWE representations, i.e., $\frac{\mathbf{e}_{u_i}^\top \mathbf{e}_{u_k}}{\|\mathbf{e}_{u_i}\|_2 \|\mathbf{e}_{u_k}\|_2}$. This similarity encodes both topic and semantic proximity information of user documents, which implicitly reveals the similarity of voting interests between two users.

6 RECOMMENDATION MODEL

In this section, we present our *Joint Topic-Semantic-aware Matrix Factorization* (JTS-MF) model for online social votings, in which social relationship, group affiliation, and topic-semantic similarities
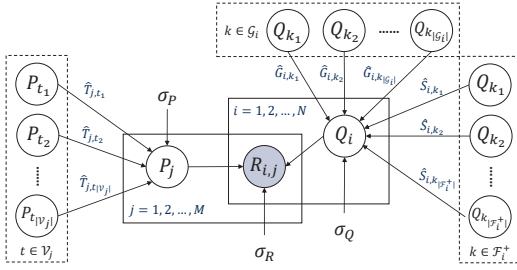
Fig. 5: Graphic Model of JTS-MF.

are combined and taken into account for voting recommendation in a comprehensive manner. Motivated by Locally Linear Embedding [23] which tries to preserve the local linear dependency among inputs in the low-dimensional embedding space, we expect to keep inter-user and inter-voting topic-semantic similarities in latent feature space as well. To this end, in JTS-MF model, while the rating $R_{i,j}$ is factorized as user latent feature Q_i and voting latent feature P_j, we deliberately enforce Q_i and P_j to be dependent on their social-topic-semantic similar counterparts, respectively. The graphic model of JTS-MF model is as shown in Figure 5.

6.1 Similarity Coefficients

In order to characterize the influence of inter-user common interests and inter-voting content relevance, we first introduce the following three similarity coefficients:

- Normalized social-level similarity coefficient of users: $\widehat{S}_{i,k}$, where u_k is the social-level friend of u_i;
- Normalized group-level similarity coefficient of users: $\widehat{G}_{i,k}$, where u_k is the group-level friend of u_i;
- Normalized similarity coefficient of voting: $\widehat{T}_{j,t}$, where v_j and v_t are two distinct votings.

Generally speaking, in JTS-MF, the latent feature Q_i for user u_i is tied up with the latent feature of his social-level and group-level friends who are weighted through $\widehat{S}_{i,k}$'s and $\widehat{G}_{i,k}$'s. Likewise, the latent feature P_j for voting v_j is tied up with the latent feature of its similar votings, which are weighted through $\widehat{T}_{j,t}$'s.

6.1.1 Normalized social-level similarity coefficient of users. Social-level similarity coefficient of users is represented by matrix $S^{N \times N}$, which incorporates both social relationship and user-user topic-semantic similarity. Specifically, for each u_i, the social-level similarity coefficient with respect to u_k is defined as

$$S_{i,k} = I_{u_i, u_k} \cdot \sqrt{\frac{d_k^- + d}{d_i^+ + d_k^- + d}} \cdot \frac{\mathbf{e}_{u_i}^\top \mathbf{e}_{u_k}}{\|\mathbf{e}_{u_i}\|_2 \|\mathbf{e}_{u_k}\|_2}, \qquad (11)$$

where I_{u_i, u_k} indicates whether u_i follows u_k as described in Eq. (2), d_i^+ is the out-degree of u_i in the social network (i.e., $d_i^+ = |\mathcal{F}_i^+|$), d_k^- is the in-degree of u_k in the social network (i.e., $d_k^- = |\mathcal{F}_k^-|$), d is the smoothing constant ($d = 1$ in this paper), and $\frac{\mathbf{e}_{u_i}^\top \mathbf{e}_{u_k}}{\|\mathbf{e}_{u_i}\|_2 \|\mathbf{e}_{u_k}\|_2}$ is the topic-semantic similarity between user u_i and user u_k as mentioned in Section 5.3. $\sqrt{\frac{d_k^- + d}{d_i^+ + d_k^- + d}}$ incorporates the information of local authority and local hub value to differentiate the importance of different users [19]. Essentially, $S_{i,k}$ counts the closeness

between two users from both topic-semantic interests and their social influence perspectives.

To avoid the impact of different numbers of followees, we use the normalized social-level similarity coefficient of users in JTS-MF, which is defined as

$$\widehat{S}_{i,k} = \frac{S_{i,k}}{\sum_{k \in \mathcal{F}_i^+} S_{i,k}}, \qquad (12)$$

where $\mathcal{F}_i^+$ denotes the set of u_i's followees in social network.

6.1.2 Normalized group-level similarity coefficient of users. Group-level similarity coefficient of users is represented by matrix $G^{N \times N}$, which actually measures the topic-semantic similarity among users from viewpoint of groups. For each u_i, the group-level similarity coefficient with respect to u_k is defined as

$$G_{i,k} = \sum_{G \in \mathcal{G}} I_{u_i, G} \cdot I_{u_k, G} \cdot \frac{\mathbf{e}_{u_i}^\top \mathbf{e}_G}{\|\mathbf{e}_{u_i}\|_2 \|\mathbf{e}_G\|_2}, \qquad (13)$$

where $\mathcal{G}$ represents the set of all groups, $I_{u_i, G}$ and $I_{u_k, G}$ indicate whether u_i and u_k join group G respectively as described in Eq. (3), and the last term is the topic-semantic similarity between user u_i and group G. Essentially speaking, $G_{i,k}$ reflects the interest closeness between user u_i and its group-level friend u_k by using u_i's topic-semantic engagement extent to the corresponding group. We also normalize the group-level similarity coefficient of users as

$$\widehat{G}_{i,k} = \frac{G_{i,k}}{\sum_{k \in \mathcal{G}_i} G_{i,k}}, \qquad (14)$$

where $\mathcal{G}_i$ is the set of u_i's group-level friends in social network.

6.1.3 Normalized similarity coefficient of votings. Similarity coefficient of votings is represented by matrix $T^{M \times M}$, which is directly defined as the topic-semantic similarity among votings, i.e.,

$$T_{j,t} = \frac{\mathbf{e}_{v_j}^\top \mathbf{e}_{v_t}}{\|\mathbf{e}_{v_j}\|_2 \|\mathbf{e}_{v_t}\|_2}. \qquad (15)$$

Since the number of votings is typically huge, we only consider the similarity between two votings with sufficiently high coefficient value. Specifically, for each voting v_j, we define a set of votings $\mathcal{V}_j$ containing those votings whose similarity coefficients with v_j exceed a threshold, i.e., $\mathcal{V}_j = \{v_t | T_{j,t} \geq threshold\}$. Correspondingly, the similarity coefficient of votings are normalized as

$$\widehat{T}_{j,t} = \frac{T_{j,t}}{\sum_{t \in \mathcal{V}_j} T_{j,t}}. \qquad (16)$$

6.2 Objective Function

Using the notations listed above, the objective function of JTS-MF can be written as

$$L = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I'_{i,j} \left(R_{i,j} - Q_i P_j^\top \right)^2 + \frac{\alpha}{2} \sum_{i=1}^{N} \left\| Q_i - \sum_{k \in \mathcal{F}_i^+} \widehat{S}_{i,k} Q_k \right\|_2^2$$
$$+ \frac{\beta}{2} \sum_{i=1}^{N} \left\| Q_i - \sum_{k \in \mathcal{G}_i} \widehat{G}_{i,k} Q_k \right\|_2^2 + \frac{\gamma}{2} \sum_{j=1}^{M} \left\| P_j - \sum_{t \in \mathcal{V}_j} \widehat{T}_{j,t} P_t \right\|_2^2 + \frac{\lambda}{2} \left(\|Q\|_F^2 + \|P\|_F^2 \right).$$
$$(17)$$

The basic idea of the objective function in Eq. (17) lies in that, besides considering explicit feedback between users and votings, we also impose penalties on the discrepancy among features of

similar users and similar votings. We give detailed explanation as follows. The first term of Eq. (17) measures the mean squared error between prediction and ground truth, where $I'_{i,j}$ is the training weights defined as

$$I'_{i,j} = \begin{cases} 1, & if\ u_i\ participates\ v_j \\ I_m, & otherwise \end{cases}. \tag{18}$$

The reason we do not directly use I_{u_i, v_j} defined in Eq. (1) as the training weights is because we found a small and positive I_m makes the training process more robust and can greatly improve the results. $R_{i,j}$ is the actual rating of user u_i on voting v_j, and $Q_i P_j^\top$ is the predicted value of $R_{i,j}$. Without loss of generality, in JTS-MF model, we set $R_{i,j} = 1$ if u_i participates v_j and $R_{i,j} = 0$ otherwise.

The second, third, and fourth terms of Eq. (17) measure the penalty of discrepancy among similar users and similar votings. In particular, the second term enforces user u_i's latent feature Q_i to be similar to the weighted average of his like-minded followees' profiles Q_k's. Weight $\widehat{S}_{i,k}$'s address both the followee u_k's social influence on u_i as well as the degree of common voting interests shared between u_k and u_i. The third term enables user u_i's latent feature Q_i to be similar to the weighted average of all his group peers' profiles Q_k's. Weight $\widehat{G}_{i,k}$'s emphasize both the same group affiliation of users u_i and u_k and also the tie strength between u_i and the associated group with respect to voting interests. This implies that, among all group-level friends, u_i would have more similar latent feature with the users who frequently join those groups u_i is interested in. The fourth term ensures voting v_j's latent feature P_j to be similar to the weighted average of votings that share similar topic-semantic information with v_j.

Finally, the last term of Eq. (17) is the regularizer to prevent over-fitting, and λ is the regularization weight.

The trade-off among user social-level similarities, user group-level similarities, and voting similarities is controlled by the parameters α, β, and γ, respectively. Obviously, users' social-level similarity, users' group-level similarity, or votings' similarity is/are ignored if α, β, or γ is/are set to 0, while increasing these values shifts the trade-off more towards their respective directions.

6.3 Learning Algorithm

To solve the optimization in Eq. (17), we apply batch gradient descent approach to minimize the objective function[6]. The gradients of loss function in Eq. (17) with respect to each variable Q_i and P_j are as follows:

$$\frac{\partial L}{\partial Q_i} = \sum_{j=1}^{M} -I'_{i,j}\left(R_{i,j} - Q_i P_j^\top\right) P_j$$
$$+ \alpha\left(\left(Q_i - \sum_{k \in \mathcal{F}_i^+} \widehat{S}_{i,k} Q_k\right) + \sum_{t \in \mathcal{F}_i} -\widehat{S}_{t,i}\left(Q_t - \sum_{k \in \mathcal{F}_t^+} \widehat{S}_{t,k} Q_k\right)\right)$$
$$+ \beta\left(\left(Q_i - \sum_{k \in \mathcal{G}_i} \widehat{G}_{i,k} Q_k\right) + \sum_{t \in \mathcal{U}} -\widehat{G}_{t,i}\left(Q_t - \sum_{k \in \mathcal{G}_i} \widehat{G}_{t,k} Q_k\right)\right) + \lambda Q_i,$$
$$\tag{19}$$

---

[6]Note that it is impractical to apply Alternating Least Squares (ALS) method here because it requires calculating the inverse of two matrices with extremely large size.

$$\frac{\partial L}{\partial P_j} = \sum_{i=1}^{N} -I'_{i,j}\left(R_{i,j} - Q_i P_j^\top\right) Q_i$$
$$+ \gamma\left(\left(P_j - \sum_{t \in \mathcal{V}_j} \widehat{T}_{j,t} P_t\right) + \sum_{k \in \mathcal{V}_j} -\widehat{T}_{k,j}\left(P_k - \sum_{t \in \mathcal{V}_k} \widehat{T}_{k,t} P_t\right)\right) + \lambda P_j.$$
$$\tag{20}$$

To clearly understand the gradients in Eq. (19) and (20), it is worth pointing out that Q_i appears not only in the i-th sub-term in the second and third lines of Eq. (17) explicitly, but also exists in other t-th sub-terms followed by $\widehat{S}_{t,i}$ or $\widehat{G}_{t,i}$, where u_i plays as one of the followees or group members of other users. The case is similar for P_j. Given the gradients in Eq. (19) and (20), we list the pseudo code of the learning algorithm for JTS-MF as follows:

(1) Randomly initialize Q and P;
(2) In each iteration of the algorithm, do:
  a) update each Q_i: $Q_i \leftarrow Q_i - \delta\frac{\partial L}{\partial Q_i}$;
  b) update each P_j: $P_j \leftarrow P_j - \delta\frac{\partial L}{\partial P_j}$;
  until convergence, where δ is an configurable learning rate.

7 EXPERIMENTS

In this section, we evaluate our proposed JTS-MF model on the aforementioned Weibo voting dataset[7]. We first introduce baselines and parameter settings used in the experiments, and then present the experimental results of JTS-MF and the comparison with baselines.

7.1 Baselines

We use the following seven methods as the baselines against JTS-MF model. Note that the first three baselines are reduced versions of JTS-MF, which only consider one particular type of similarity among users or votings.

- **JTS-MF(S)** only considers social-level similarity of users, i.e., sets $\beta, \gamma = 0$ in JTS-MF model.
- **JTS-MF(G)** only considers group-level similarity of users, i.e., sets $\alpha, \gamma = 0$ in JTS-MF model.
- **JTS-MF(V)** only considers similarity of votings, i.e., sets $\alpha, \beta = 0$ in JTS-MF model.
- **MostPop** recommends the most popular items to users, i.e., the votings that have been participated by the most numbers of users.
- **Basic-MF** [13] simply uses matrix factorization method to predict the user-voting matrix while ignores additional social relation, group affiliation and voting content information.
- **Topic-MF** [1] is similar to JTS-MF except that we substitute Θ_d for e_d when calculating similarities in Eq. (11), (13), and (15). Note that Θ_d can also be viewed as the embedding of document with respect to topics. Therefore, Topic-MF only considers the topic similarity among users and votings.
- **Semantic-MF** is similar to JTS-MF except that we use the Skip-Gram model in [21] directly to learn the word embeddings. Therefore, Semantic-MF only considers the semantic similarity among users and votings.
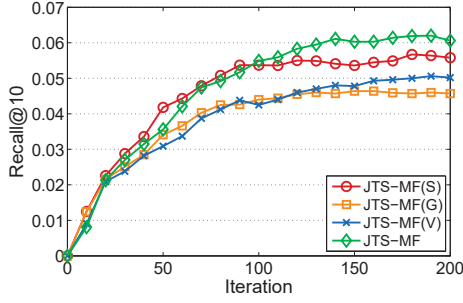
---

[7]Experiment code is provided at https://github.com/hwwang55/JTS-MF.

Fig. 6: Convergence of JTS-MF models with respect to *Recall@10*.

7.2 Parameter Settings

We use GibbsLDA++[8], an open-source implementation of LDA using Gibbs sampling, to calculate topic information of words and documents in JTS-MF and Topic-MF models. We set the number of topics to 50 and leave all other parameters in LDA as default values. For word embeddings in JTS-MF and Semantic-MF models, we use the same settings as follows: length of embedding dimension as 50, window size as 5, and number of negative samples as 3.

For all MF-based methods, we set the learning rate $\delta = 0.001$ and regularization weight $\lambda = 0.5$ by 10-fold cross validation. Typically, we set $I_m = 0.01$ in Eq. (18). Taking into consideration the balance of experimental results and time complexity, we run 200 iterations for each of the experiment cases. To conduct the recommendation task, we randomly select 20% of users' voting records in the dataset as test set and use the remaining data as the trainning examples for our JTS-MF model as well as all baselines. The choice of remaining hyper-parameters (trade-off parameters α, β, γ, and dimension of latent features *dim*) is discussed in Section 7.4.

To quantitatively analyze the performance of voting recommendation, in our experiment, we use *top-k recall* (*Recall@k*), *top-k precision* (*Precision@k*), and *top-k micro-F1* (*Micro-F1@k*) as the evaluation metrics.

7.3 Experiment Results

7.3.1 Study of convergence. To study the convergence of JTS-MF model, we run the learning algorithm up to 200 iterations for JTS-MF(S) with $\alpha = 10$, JTS-MF(G) with $\beta = 140$, JTS-MF(V) with $\gamma = 30$, JTS-MF with $\alpha = 10$, $\beta = 140$, $\gamma = 30$ (*dim* = 10 for Q_i and P_j in all models), then calculate *Recall@10* for every 10 iterations. The result of convergence of JTS-MF models is plotted in Fig. 6. From Fig. 6 we can see that, the recall of JTS-MF models rises rapidly before 100 iterations, and starts to oscillate slightly after around 150 iterations. The same changing pattern is observed for all four JTS-MF variants. Therefore, we set the number of learning iterations as 200 to achieve a balance between running time and performance of models.

7.3.2 Study of JTS-MF. To study the performance of JTS-MF model and the effectiveness of three types of similarities, we run JTS-MF model as well as its three reduced versions on Weibo voting dataset, and report the results of *Recall*, *Precision*, and *Micro-F1* in

[8]GibbsLDA++: http://gibbslda.sourceforge.net

Fig. 7. The parameter settings of α, β, γ, and *dim* are the same as in Section 7.3.1. Fig. 7a, 7b, and 7c consistently demonstrate that JTS-MF(S) performs best and JTS-MF(G) performs worst among three types of reduced versions of JTS-MF. Note that JTS-MF(S) only considers users' social-level similarity and JTS-MF(G) only considers users' group-level similarity. Therefore, it could be concluded that social-level friends are more helpful than group-level friends when determining users' voting interest. This is in accordance with our intuition, since a user typically has much more group-level friends than social-level friends, which inevitably dilutes its effectiveness and brings noises into group-level relationship. In addition, the result in Fig. 7 also demonstrates the effectiveness of the usage of votings' similarity. Furthermore, it can be evidently observed that JTS-MF model outperforms its three reduced versions in all cases, which proves that the three types of similarities are well combined in JTS-MF model to achieve much better results.

7.3.3 Comparison of Models. To further compare JTS-MF model with other baselines, we gradually increase k from 1 to 500 and report the results in Table 2 with the best performance highlighted in bold. The value of α, β, and γ for JTS-MF and its reduced models are the same as in Section 7.3.1. The parameter settings are $\alpha = 2$, $\beta = 60$, $\gamma = 15$ for Topic-MF, $\alpha = 8$, $\beta = 120$, $\gamma = 20$ for Semantic-MF, and *dim* = 10 for Q_i and P_j in all MF-based methods. The above parameter settings are the optimal results of fine tuning for given *dim*. In Table 2, several observations stand out:

- MostPop performs worst among all methods, because MostPop simply recommends the most popular votings to all users without considering users' specific interests.
- Topic-MF and Semantic-MF outperforms Basic-MF, which proves the usage of similarities with respect to topic and semantic helpful for recommending votings. Besides, Semantic-MF outperforms Topic-MF. This suggests that semantic information is more accurate than topic information when measuring similarities through mining short-length texts.
- JTS-MF outperforms Topic-MF and Semantic-MF. This is the most important observation from Table 2, since it justifies our aforementioned claim that joint-topic-semantic model can benefit from both topic and semantic aspects and achieve better performance.
- The significance of JTS-MF over other models is evident for small k. However, this margin becomes smaller when k gets larger, and JTS-MF is even slightly inferior to JTS-MF(S) when $k \geq 50$. This means that users' group-level similarities and votings' similarities "drag the feet" of JTS-MF model when k is large. However, JTS-MF is still preferred in practice, since a real recommender system would only recommend a small set of votings to users in general.

7.4 Parameter Sensitivity

We investigate parameter sensitivity in this subsection. Specifically, we evaluate how different value of trade-off parameters α, β, γ, and different numbers of latent feature dimensions *dim* can affect the performance.

7.4.1 Trade-off parameters. We fix *dim* = 10, keep two of the trade-off parameters as 0, and vary the value of the left trade-off
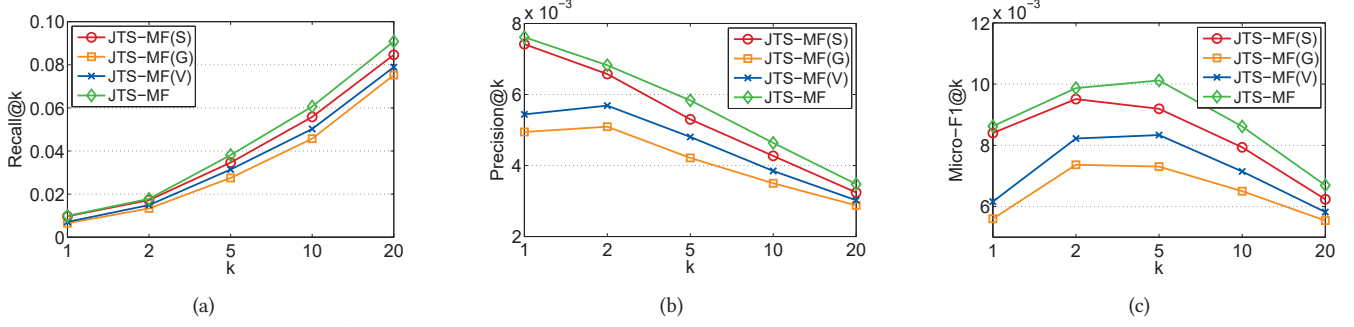
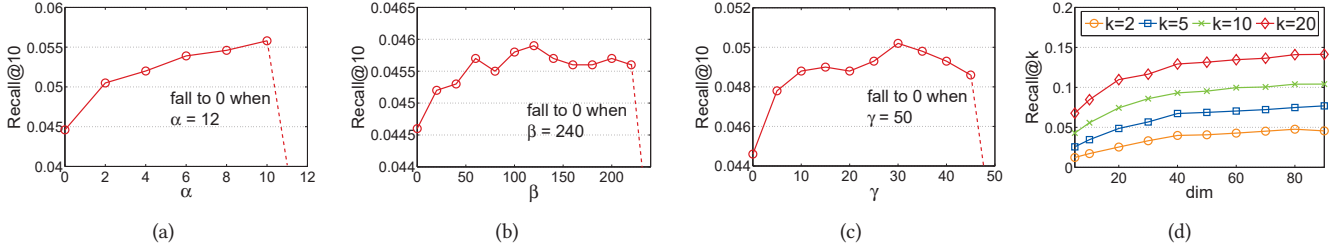Fig. 7: (a) *Recall@k*, (b) *Precision@k*, and (c) *Micro-F1@k* of JTS-MF models.

Fig. 8: Parameter sensitivity with respect to (a) α, (b) β, (c) γ, and (d) *dim*.

Table 2: Result of *Recall@k, Precision@k,* and *Micro-F1@k* for JTS-MF model and baselines.

Model	Metric	k							
		1	2	5	10	20	50	100	500
JTS-MF(S)	*Recall*	0.0097	0.0172	0.0346	0.0558	0.0846	**0.1529**	**0.2229**	**0.4392**
	Precision	0.007416	0.006575	0.005300	0.004271	0.003238	**0.002341**	**0.001707**	**0.000672**
	Micro-F1	0.008401	0.009511	0.009192	0.007935	0.006238	**0.004612**	**0.003387**	**0.001343**
JTS-MF(G)	*Recall*	0.0065	0.0133	0.0275	0.0457	0.0752	0.1360	0.2051	0.4216
	Precision	0.004944	0.005092	0.004212	0.003500	0.002877	0.002082	0.001570	0.000645
	Micro-F1	0.005601	0.007365	0.007306	0.006503	0.005542	0.004102	0.003116	0.001289
JTS-MF(V)	*Recall*	0.0071	0.0149	0.0314	0.0502	0.0789	0.1387	0.2049	0.4176
	Precision	0.005439	0.005685	0.004805	0.003846	0.003021	0.002124	0.001568	0.000639
	Micro-F1	0.006161	0.008223	0.008335	0.007145	0.005819	0.004184	0.003112	0.001277
JTS-MF	*Recall*	**0.0099**	**0.0178**	**0.0381**	**0.0606**	**0.0908**	0.1520	0.2187	0.4297
	Precision	**0.007614**	**0.006823**	**0.005834**	**0.004637**	**0.003475**	0.002327	0.001674	0.000658
	Micro-F1	**0.008625**	**0.009868**	**0.010118**	**0.008615**	**0.006695**	0.004585	0.003322	0.001314
MostPop	*Recall*	0.0042	0.0085	0.0191	0.0313	0.0517	0.0974	0.1455	0.3086
	Precision	0.003221	0.003261	0.002921	0.002403	0.001972	0.001482	0.001119	0.000469
	Micro-F1	0.003637	0.004721	0.005062	0.004468	0.003804	0.002925	0.002218	0.000937
Basic-MF	*Recall*	0.0063	0.0129	0.0274	0.0446	0.0727	0.1368	0.2050	0.4198
	Precision	0.004845	0.004944	0.004192	0.003411	0.002783	0.002094	0.001569	0.000643
	Micro-F1	0.005489	0.007151	0.007271	0.006337	0.005361	0.004125	0.003114	0.001283
Topic-MF	*Recall*	0.0076	0.0147	0.0311	0.0495	0.0781	0.1395	0.2076	0.4210
	Precision	0.005834	0.005636	0.004766	0.003787	0.002991	0.002136	0.001589	0.000644
	Micro-F1	0.006609	0.008152	0.008266	0.007035	0.005761	0.004207	0.003154	0.001287
Semantic-MF	*Recall*	0.0093	0.0169	0.0333	0.0545	0.0860	0.1471	0.2142	0.4293
	Precision	0.007120	0.006476	0.005102	0.004173	0.003293	0.002252	0.001639	0.000657
	Micro-F1	0.008065	0.009368	0.008849	0.007752	0.006342	0.004437	0.003254	0.001313

parameter. Then we report *Recall@10* in Fig. 8a, 8b, and 8c, respectively.

As shown in Fig. 8a, the *Recall@10* increases constantly as α gets larger and reaches a maximum of 0.0558 when $\alpha = 10$. This suggests that the usage of users' social-level similarity do help to improve the recommendation performance. However, when α is too large ($\alpha =$ 12), the learning algorithm of JTS-MF is misled to wrong direction when updating latent features of users and votings, resulting in performance deterioration. The similar phenomenon are observed in Fig. 8b and Fig. 8c, too. According to the results, when the other two trade-off parameters are set to 0, *Recall@10* reaches the maximum when $\alpha = 10$, $\beta = 140$, and $\gamma = 30$, respectively.

Therefore, in previous experiments we adopt these optimal settings for JTS-MF(S), JTS-MF(G), and JTS-MF(V), respectively, and use their combination as the parameter settings in JTS-MF.

7.4.2 Dimension of latent features. We fix $\alpha = 10$, $\beta = 0$, $\gamma = 0$ and tune the dimension of latent features of users and votings from 5 to 90. The result is shown in Fig. 8d. From the figure, we can see clearly that the recall is increasing when *dim* gets larger, this is because latent features with larger number of dimensions have more capacity to characterize users and votings. But a larger *dim* leads to more running time in experiments. Moreover, we notice that the improvement of performance stagnates after *dim* reaches 80. On balance, we set *dim* = 10 in our experiment scenarios to ensure the experiments can complete within rational time duration.

8 CONCLUSIONS

In this paper, we study the problem of recommending online votings to users in social networks. We first formalize the voting recommendation problem and justify the motivation of leveraging social structure and voting content information. To overcome the limitations of topic models and semantic models when learning representation of voting content, we propose Topic-Enhanced Word Embedding method to jointly consider topics and semantics of words and documents. We then propose our Joint-Topic-Semantic-aware social Matrix Factorization model to learn latent features of users and votings based on the social network structure and TEWE representation. We conduct extensive experiments to evaluate JTS-MF with Weibo voting dataset. The experimental results prove the competitiveness of JTS-MF against other state-of-the-art baselines and demonstrate the efficacy of TEWE representation.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. In *Journal of Machine Learning Research*. 993–1022.
[2] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (2013).
[3] Marco Bressan, Stefano Leucci, Alessandro Panconesi, Prabhakar Raghavan, and Erisa Terolli. 2016. The Limits of Popularity-Based Recommendations, and the Role of Social Ties. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 745–754.
[4] Danqi Chen and Christopher D Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks.. In *EMNLP*. 740–750.
[5] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 193–202.
[6] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-aware point of interest recommendation on location-based social networks.. In *AAAI*. 1721–1727.
[7] Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. In *Proceedings of National Academy of Sciences (PNAS)*. 5228–5235.
[8] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.
[9] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 595–606.
[10] Hui-Ju Hung, Hong-Han Shuai, De-Nian Yang, Liang-Hao Huang, Wang-Chien Lee, Jian Pei, and Ming-Syan Chen. 2016. When Social Influence Meets Item Inference. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 915–924.
[11] Janet Ilieva, Steve Baron, and Nigel M Healey. 2002. Online surveys in marketing research: Pros and cons. *International Journal of Market Research* 44, 3 (2002), 361.
[12] Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2015. Social recommendation with cross-domain transferable knowledge. *IEEE Transactions on Knowledge and Data Engineering* 27, 11 (2015), 3084–3097.
[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[14] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*. 331–339.
[15] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML*, Vol. 14. 1188–1196.
[16] Wu-Jun Li, Dit-Yan Yeung, and Zhihua Zhang. 2011. Generalized latent factor models for social network analysis. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain*. 1705.
[17] Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*. Springer, 415–463.
[18] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Sun Maosong. 2015. Topical word embeddings. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI)*. AAAI, 2418–2424.
[19] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 931–940.
[20] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 171–180.
[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*. 3111–3119.
[22] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 273–282.
[23] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.
[24] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*. ACM, 791–798.
[25] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. 2012. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 139–146.
[26] Shaojie Tang and Jing Yuan. 2016. Optimizing Ad Allocation in Social Advertising. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1383–1392.
[27] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.
[28] Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. 2016. Recommending groups to users using user-group engagement and time-dependent matrix factorization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 1331–1337.
[29] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social Recommendation with Strong and Weak Ties. In *Proceedings of the 25th ACM CIKM*. ACM, 5–14.
[30] Zhonghang Xia, Yulin Dong, and Guangming Xing. 2006. Support vector machines for collaborative filtering. In *Proceedings of the 44th annual Southeast regional conference*. ACM, 169–174.
[31] Xiwang Yang, Chao Liang, Miao Zhao, Hongwei Wang, Hao Ding, Yong Liu, Yang Li, and Junlin Zhang. 2017. Collaborative filtering-based recommendation of online social voting. *IEEE Transactions on Computational Social Systems* 4, 1 (2017), 1–13.
[32] Qin Zhang, Jia Wu, Peng Zhang, Guodong Long, Ivor W Tsang, and Chengqi Zhang. 2016. Inferring Latent Network from Cascade Data for Dynamic Social Recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 669–678.
[33] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 56–65.
[34] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong, and Jilei Tian. 2012. Mining personal context-aware preferences for mobile users. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 1212–1217.