Knowledge-Based Systems 210 (2020) 106433

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Joint Topic-Semantic-aware Social Matrix Factorization for online voting recommendation[☆]

Jia Wang [a,*,1], Hongwei Wang [a,b,1], Miao Zhao [a], Jiannong Cao [a], Zhuo Li [a], Minyi Guo [b]

[a] The Hong Kong Polytechnic University, Hong Kong
[b] Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Article history:
Received 28 November 2019
Received in revised form 9 August 2020
Accepted 12 September 2020
Available online 19 September 2020

Keywords:
Online voting
Recommender systems
Topic-enhanced word embedding
Matrix factorization

ABSTRACT

Social voting is an emerging new feature in online social platforms, through which users can express their attitudes and opinions towards various interested subjects. Since both social relations and textual content decide the votes propagation, the diverse sources present opportunities and challenges for recommender systems. In this paper, we jointly consider these two factors for the online voting recommendation. First, we conduct feature learning on the vote content. Note that the vote questions are usually short and contain informal expressions, existing text mining methods cannot handle it well. We propose a novel topic-enhanced word embedding (TEWE) method, which learns the word vectors by considering both token-level semantics and document-level mixture topics. Second, we propose two Joint Topic-Semantic-aware Social Matrix Factorization (JTS-MF) models, which fuse social relations and textual content for the vote recommendation. Specifically, JTS-MF1 directly identifies the interaction strength to calculate the similarity among users and votes, while JTS-MF2 aims to preserve inter-user and inter-vote similarities during matrix factorization. Extensive experimental results on real online voting dataset show the effectiveness of our approaches against several state-of-the-art baselines. JTS-MF1 and JTS-MF2 models surpass the matrix factorization based method, with 25.4% and 57.1% improvements in the top-1 recall, and 59.12% and 25.1% improvements in the top-10 recall.

1. Introduction

1.1. Background of online voting

With the rapid growth of information available on social media, recommendation systems (RSs) have become important to help users quickly find personalized information that suits their needs. Social networks like Twitter and Weibo are popular platforms for users to connect with each other, participate in online activities, and share opinions. Many social platforms recently have launched a voting function [1], in which users can express their various attitudes towards various issues and show their unique interest, e.g., like or dislike, on various subjects, ranging from livelihood issues to entertainment news.

To enrich the functionality of online voting, some online social platforms, such as Weibo,[2] have empowered users to run their own voting campaigns on any topic of interest. Users can connect to each other and freely initiate votes and customize vote options. The friends of a voting initiator can participate the voting and retweet the votes to their followers. In this way, a vote can widely propagate over the network along social paths. Fig. 1 illustrates an example of a vote propagation scheme. In this example, a user (in orange) initiates a voting campaign with the title "Who is your favorite movie star?". This vote is directly visible to his social friends (i.e., followers) and users in the group named "movie group" ((i.e., the initiator has joined). After a friend participates or simply retweets the vote, the vote can be further seen by all of his followers. Such propagation continues to spread out based on social connections so that the two-hop and three-hop followers of the initiator can also involve the voting.

In fact, we can have different level of "documents" related to a vote, a user or a group, which reflects the interested voting topics that users may participate. Typically, a vote document

☆ This paper is an extended version of a conference paper entitled "Joint-Topic-Semantic-aware Social Recommendation for Online Voting" that is included in Proceedings of the 26th ACM Conference on Information and Knowledge Management (CIKM'17).

* Corresponding author.
E-mail addresses: csjiawang@comp.polyu.edu.hk (J. Wang), hongweiw@cs.stanford.edu (H. Wang), csmiaozhao@comp.polyu.edu.hk (M. Zhao), csjcao@comp.polyu.edu.hk (J. Cao), cszhuoli@comp.polyu.edu.hk (Z. Li), guo-my@cs.sjtu.edu.cn (M. Guo).

1 Jia Wang and Hongwei Wang contribute equally and share the co-first authorship.

2 http://weibo.com.

https://doi.org/10.1016/j.knosys.2020.106433
0950-7051/© 2020 Elsevier B.V. All rights reserved.

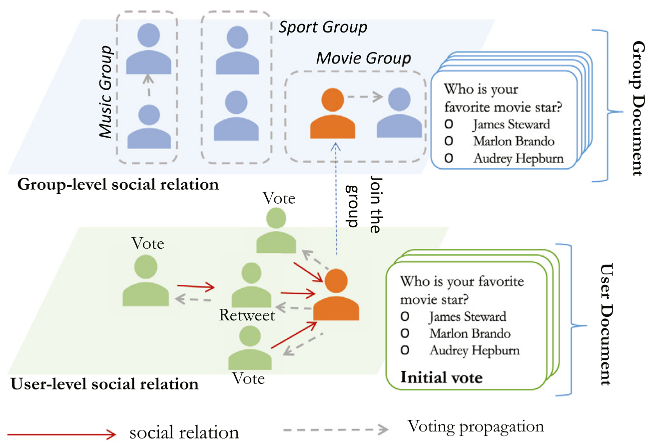


Fig. 1. Propagation scheme of online votes.

is the content of vote question, a user document is formed by aggregating all the documents of votes he participates, and a group document is formed by aggregating all the documents of users who join the group.

Usually, users are more willing to participate in the votings that social friends have joined or that match their interests. Facing a large volume of diversified votes, an effective recommender system should be instrumental, which helps precisely determining what votes favor each user most, thus allows users to quickly find the personalized information that maximizes user engagement in votes. Such a recommender system can also have many potential commercial values and benefit a variety of other online services such as personalized advertising [2], market research [3], and public opinion analysis [4].

1.2. Challenges in vote recommendation

Considering that both social relations and vote content affect the recommendation results, there is little prior work considering recommending votes to users in social networks. The challenges are two-fold:

(1) The propagation of online votes relies heavily on social network structure and it follows bootstrap percolation phenomenon [5]. A user can see the votes initiated, participated or retweeted by his followees, which implies that a user is more likely to be exposed to the votes 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. Several prior works from the computer science community [6–10] propose to leverage social network information in recommendation. Also, in the field of statistical physics [11,12], many diffusion-based algorithms are proposed by introducing physical dynamics (i.e., heat conduction mass diffusion [13] and heat conduction [14]) into the recommender systems. [15,16] argue that relying on the similarity in social patterns may lead to accurate yet less-diverse recommendations. Hence, it is still an open question on how to comprehensively incorporate structural social information into the task of vote recommendation considering its unique propagation pattern.

(2) We notice that the content of votes in question texts (e.g., "What color is the trend of this season?") also affects the user's engagement in voting. The existing text mining methods can be categorized into two classes: Topic model [17] and semantic model [18]. Topic model discovery users' voting interests through discovering the latent topic distribution of relevant vote text. However, voting questions are usually short and contain informal Internet slang (e.g., "A/S/L?" stands for "age, sex, location"), leading to severe degradation of topic model performance. Alternatively, semantic model [18] can discover users' voting interests through learning text representations. However, such semantic methods typically represent each word using a single vector, making them indiscriminate for homonym and polysemy that are especially common in vote questions (e.g., "sauce" stands for "source" when asking for the online source of an image or other posted material".). In brief, these inherent defects of the above models limit their power to characterize the vote contents that are usually short and informally expressed.

1.3. Our proposed method: JTS-MF

To address aforementioned challenges, in this paper, we propose a novel *Joint Topic-Semantic-aware Matrix Factorization* (JTS-MF) framework for online vote recommendation. JTS-MF considers social network structure and vote content in a comprehensive manner. For social network structure, JTS-MF fully encodes the information of social relationship and group affiliation into the objective function. We will further explain the motivation for using social network structure in Section 3. For vote content, we propose a *Topic-Enhanced Word Embedding* (TEWE) method which build multi-prototype word and document representations by considering both token-level semantics and document-level mixture topics. The key idea of TEWE is to enable each word to have different representations in the same context of the document but under different topics. The detailed part of TEWE will be introduced in Section 5.

After obtaining TEWE representation for each document, we propose two models under JTS-MF framework for vote recommendation, namely JTS-MF1 and JTS-MF2. In JTS-MF1, we characterize the interaction strength between each user and each vote directly by introducing an influence index, which measures the potential enthusiasm of a user on a vote. The influence index consists of three components: the topic-semantic similarity between a user and a vote with respect to their TEWE representations, the popularity of a vote among user's social-level friends, and the popularity of a vote among user's group-level friends. The influence index is incorporated into JTS-MF1 model as an impact factor when factorizing user–vote interactions.

Similar to JTS-MF1, JTS-MF2 also takes vote content and social structure into consideration simultaneously. Specifically, rather than measuring the interaction strength between users and votes directly as in JTS-MF1, in JTS-MF2, we consider user–user similarity and vote–vote similarity with respect to their TEWE representations and structural information of social networks. The reason of calculating such similarity is that, inspired by Locally Linear Embedding (LLE) [19], we try to preserve the similarity among latent features of users and votes during matrix factorization process, as it contains abundant proximity information and can greatly benefit representation learning for users and votes. We conduct extensive experiments on both JTS-MF models with a real online vote dataset from Weibo. The experiment results in Section 7 demonstrate that JTS-MF models achieve substantial gains compared with baselines. The results also show that TEWE is able to well combine topic and semantic information of texts and generates better document representation.

1.4. Contributions

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

- We formulate the online vote recommendation problem, and find 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 representations. TEWE takes advantages of both topic/semantic models and learns more informative embeddings.
- We develop two novel matrix factorization based models, named JTS-MF1 and JTS-MF2, for an online vote recommendation. Both JTS-MF models are able to utilize the topic-semantic similarity among users and votes as well as the structural information of social networks during the learning process.
- We conduct extensive experiments on real online vote dataset, and the results reveal that JTS-MF models significantly outperforms baseline methods. For example, JTS-MF1 and JTS-MF2 surpass basic matrix factorization model with 25.4% and 57.1% improvements in top-1 recall, and with 49.2% and 58.3% improvements in top-1 precision, respectively.

2. Related work

2.1. Recommender systems

Roughly speaking, existing recommender systems can be categorized into three classes [12]: content-based, collaborative filtering, and hybrid methods. Content-based methods [20,21] make use of user profile or item description as features for recommendation. Collaborative filtering methods [1,22] 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 recommendation. Hybrid methods [23,24] combine content-based and collaborative filtering models in many hybridization approaches, such as weighted, switching, cascade and feature combination or augmentation. A variety of models are incorporated into recommender systems, such as support vector machine [25], restricted Boltzmann machine [26] , heterogeneous information network [27], and attention model [28]. In addition, recently, some recent works have employed deep learning to design recommendation systems. Collaborative Memory Networks methods [29–31] combine memory networks and neural attention mechanisms for recommendations. Meta-path based methods [32,33] leverage auxiliary information like movie genres to make top-n recommendations. Conditional VAE based methods [34–36] leverage both items content and ratings to learn implicit relationships between items and users. Collaborative deep learning methods [37–39] incorporate the deep learning extracted content information into collaborative filtering for recommendations. However, all these neural approaches require the carefully tuned deep model, and the repeatability and interpretability of the model results are weak.

2.2. Social recommendation

A number of advanced recommendation algorithms have been proposed by researchers both in physics and computer science domains. Traditional recommender systems are vulnerable to data sparsity and cold start problem. To mitigate these issues, many approaches have been proposed to utilize social network information in recommender systems [1,40,41]. For example, [42] 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. [6] investigates the seed selection problem for viral marketing that considers both effects of social influence and item inference for the product recommendation. [8] studies the effects of strong and weak ties in a social recommendation, and extends Bayesian Personalized Ranking model to incorporate the distinction of strong and weak ties. DSCF [43] captures social network information to consider the influence of distant neighbors on personalized recommendation. NGCF [44] models the high-order connectivity in user–item graph to extract implicit collaborative signal. Furthermore, the physics community has considered physical dynamics and proposed a diffusion-based method [45,46] for personalized suggestions. However, the above works only utilize users' social links without considering the topic and semantic information to mine the similarities among users and items, which we found quite helpful for vote recommendation tasks. Another difference between these work and ours is that we also take hierarchical social structure (i.e, user-level following information and group membership) into consideration, which can further improve the performance of recommendation.

2.3. Short texts analysis

Short texts are common in a variety of Internet applications, such as product reviews, news feeds, forum posts, Twitter messages, and blogs. Because of the length limitation, short texts do not contain sufficient statistical information to support state-of-the-art NLP methods to achieve desirable performance. To tackle this issue, researchers proposed special methods to analyze short texts, such as lexicon-based short texts analysis [47,48], short texts enrichment [49,50], and sentence structure analysis [51,52]. For example, [47] uses lexical semantic knowledge from a well-known semantic network by identifying the lexical relationship between terms in a sentence for short texts understanding. To enrich short texts information, [50] regard hashtags as self-annotation labels, [53] deploys the named-entity recognition techniques, and [54,55] leverages emojis as an extra type of information. Sentence structure analysis method [52] introduces a sentiment Treebank to represent short texts based on its syntactic structure. In addition, to model different meanings of a word in different contexts, researchers proposed several deep learning based methods. For example, ELMo [56] learns word embeddings of short text from language models representations. They pretrained a deep word representations model on the large corpus dataset and then fine-tuned it on down-stream tasks.

2.4. Topic and semantic language models

Latent Dirichlet Allocation (LDA) [17] is a well-known generative topic model used to learn the latent topic distribution of documents. Topic modeling methods are widely used in aspect and opinion mining [57], sentiment analysis [58] and recommendation [59].

Word2vec [18] is a typical example of semantic modeling method. It learns word representations by capturing precise syntactic and semantic word relationships. The Skip-Gram model associated with the semantic modeling method is widely used for document classification [60], dependency parser [61] and web embedding [62]. However, LDA and Word2vec are usually used for long texts. They are not directly applicable to the case of recommended voting, because the content of the voting is usually short and ambiguous. As a combination, [63] tries to learn topical word embeddings based on both words and their topics. The difference between [63] and ours is that we also take topics of documents into consideration, which enables our model to learn an even more discriminative and informative representations of words and documents.

3. Background and data analysis

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

3.1. Background

Sina Weibo[3] is a Chinese microblogging (weibo) website, Launched by Sina Corporation on 14 August 2009. It is like a hybrid of Twitter and Facebook platforms, in which user can follow each other, write posts (tweets) and share them with his/her followers. Weibo is one of the most popular Chinese microblogging platform based on fostering user relationships to share, disseminate and receive information. Users can also join different interest groups based on their locations (e.g., Los Anglos) or self-interested topics (e.g., music).

Weibo has an embedded function called Voting.[4] At least 92 million users have participated in at least one vote as of January 2013, and more than 2.2 million ongoing votes are running on Weibo every day. As shown in Fig. 1, any user can freely initiate, a voting campaign. Other users can see and potentially participate in the voting through two ways. The first way is social propagation: after a user initiates a vote, all his/her social friends (i.e., followers or users in the interest group) can see the vote. Then users can participate in the voting campaign or simply retweet it to their social friends. The second way is through the voting recommendation list of Weibo, which include popular votes and personalized recommendations. The mechanism of the recommendation list is commonly without public disclosure.

3.2. Data analysis

We directly obtained user voting-logs from November 2010 to January 2012 from the technical team of Sina Weibo. The dataset covers information about user's voting participation records,[5] voting contents, and the statues of each vote. In addition, the dataset also includes social connections between users (e.g., A follows B), title and topic of each group, and user-group affiliation.

Basic statistics. The basic statistics of the dataset are summarized in Table 1. On average, each user has participated in 3.9 votes, has 165.4 followers/followees, and has joined 5.6 groups. If we only consider users who participate in at least one vote and users who join more than one group, a user commonly has participated in 7.4 votes and joined 7.8 groups on average. Fig. 2 illustrates the distribution curves of the above statistics, where the meaning of each subfigure is given in the caption. One can observe that the average number of participants of a voting is 21.1, the average number of users joining a same group is 18.9, and the average number of votings participated by a group is about 56.7. 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:

The influence of social relations on the voting participation. With the ten million user pairs randomly selected from the set of all users, we count the average number of votes that both users 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-level friends; (3) the two users are neither social-level friends nor group-level friends; (4) all cases.

Table 1
Basic statistics of Weibo vote dataset.

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

The results are presented in Fig. 3(a), which clearly shows the difference between these given cases. In fact, the average number of common votes 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 with no social relations (2.04×10^{-5}). The results demonstrate that if two users are social-level or group-level friends, they are likely to participate more votes in common.

The probability that two users participating in a common vote are friends. With ten thousand votes randomly selected from the set of all votes, we calculate the probability whether users participating in the same vote (e.g., v_j) are social or group-level friends as follows.

$$p_j = \frac{\text{\# of social/group-level friends among participants of } v_j}{n_j \times (n_j - 1)/2}, \tag{1}$$

where n_j is the number of v_j' participants. We calculate p_j over all sampled votes. For comparison, we plot the average result (blue bar) in Fig. 3(b) and the result of randomly sampled set of users (green bar) in Fig. 3(b). It is clear that if two users have ever participated in a common vote, they are more likely to be social-level or group-level friends. In fact, the probabilities of two users being social-level or group-level friends are increased by 5.3 and 3.6 if they have joined the same voting campaign.

The above two findings effectively prove that both user-level and group-level social relationships strongly influence voting participation, which motivates us to consider these unique propagation patters when doing vote recommendation.

4. Problem formulation

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

1. The user–vote relationship for u_i and v_j is defined as

$$I_{u_i,v_j} = \begin{cases} 1, & \text{if } u_i \text{ participates } v_j; \\ 0, & otherwise. \end{cases} \tag{2}$$

2. The user–user relationship for u_i and u_k is defined as

$$I_{u_i,u_k} = \begin{cases} 1, & \text{if } u_i \text{ follows } u_k; \\ 0, & otherwise. \end{cases} \tag{3}$$

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_l is defined as

$$I_{u_i,G_l} = \begin{cases} 1, & \text{if } u_i \text{ joins } G_l; \\ 0, & otherwise. \end{cases} \tag{4}$$

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

[3] http://www.weibo.com.

[4] http://www.weibo.com/vote?is_all=1.

[5] We only know whether a user participated a vote or not, rather than user voting results, i.e., we do not know which vote option a user chose.

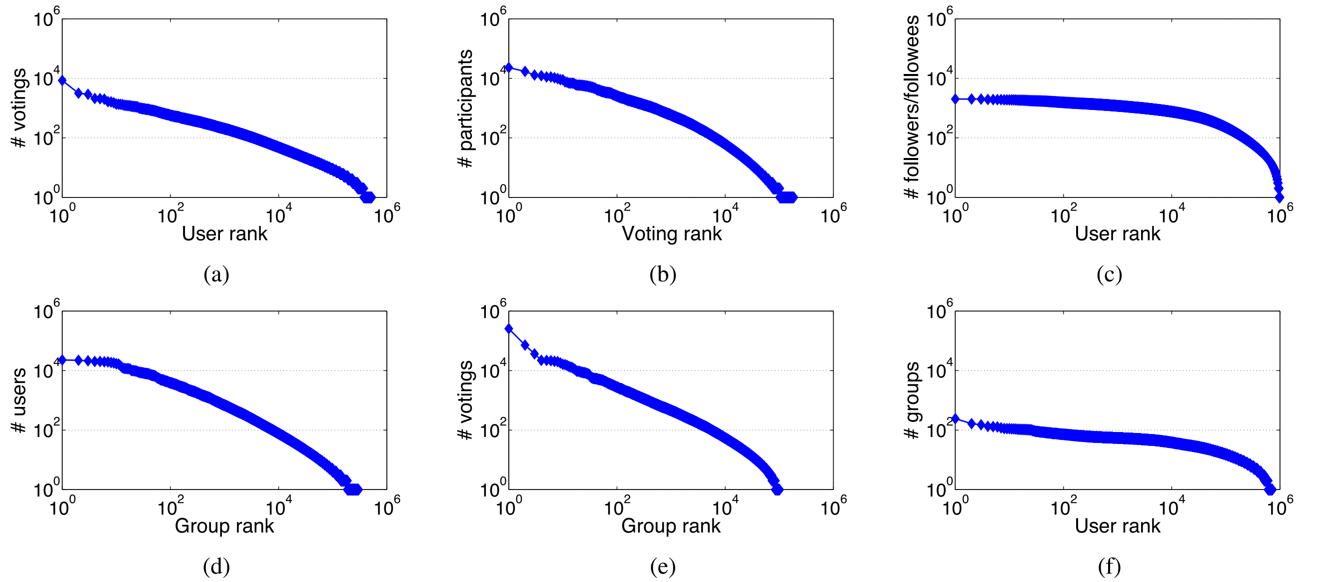


Fig. 2. (a) Distribution of the number of votes participated by a user; (b) Distribution of the number of participants of a vote; (c) Distribution of the total number of followers and followees of a user; (d) Distribution of the number of users in a group; (e) Distribution of the number of votes (may contain duplicated votes) participated by all users in a group; (f) Distribution of the number of groups joined by a user.

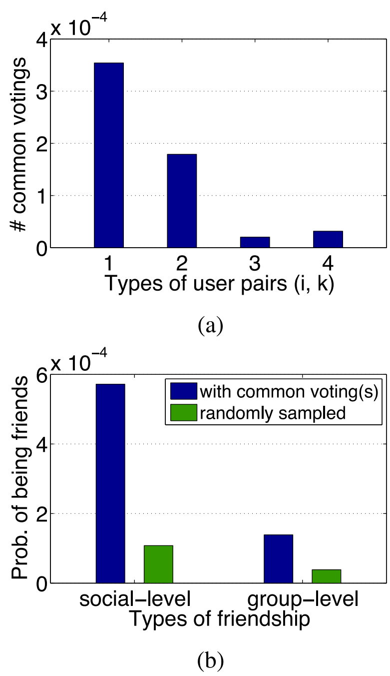


Fig. 3. (a) Average number of common votes participated by user u_i and u_k in four cases: 1. u_i follows/is followed by u_k; 2. u_i and u_k are in at least one common group; 3. u_i and u_k have no social-level and group-level relationship; 4. all cases; (b) Probability of two users being social-level or group-level friends in two cases: 1. they ever participated at least one common vote; 2. they are randomly sampled.

5. Joint-topic-semantic embedding

In this section, we explain how to learn the embeddings of users, votes, 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 that combines these two models to learn more powerful embeddings.

5.1. Topic distillation

In this subsection, we introduce how to profile documents of votes, users and groups in terms of topic distillation. LDA is a popular generative model to perform topic distillation on the text documents [17]. In LDA, each document l is represented as a multinomial distribution $\boldsymbol{\Theta}_l$ over a set of topics, and each topic z is also represented as a multinomial distribution $\boldsymbol{\Phi}_z$ over a number of words. We denote $\mathrm{Dir}(\alpha)$ and $\mathrm{Dir}(\beta)$ as the Dirichlet prior of $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$. Given α and β, by traversing all the documents l, the document-topic distributions $\boldsymbol{\Theta}$, topic-word distributions $\boldsymbol{\Phi}$ can be iteratively calculated as follows.

$$p(\boldsymbol{\Theta}, \boldsymbol{\Phi}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_z p(\boldsymbol{\Phi}_z|\boldsymbol{\beta}) \cdot \prod_l \left(p(\boldsymbol{\Theta}_l|\boldsymbol{\alpha}) \prod_t \left(p(z_{l,t}|\boldsymbol{\Theta}_l) p(w_{l,t}|\boldsymbol{\Phi}_{z_{l,t}}) \right) \right), \tag{5}$$

where $z_{l,t}$ is a topic assigned to each word at position t in document l according to $\boldsymbol{\Theta}_l$, and $w_{l,t}$ is a word generated at position t in document l according to $\boldsymbol{\Phi}_{z_{l,t}}$.

Here, we discuss how to apply LDA in the scenario of Weibo vote. According to the Weibo vote dataset, each vote v_j associates a sentence of question, which can be regarded as document d_{v_j}.[6] The document d_{u_i} for user u_i can thus be formed by aggregating the content of all votes he participates, i.e., $d_{u_i} = \cup\{d_{v_j}|I_{u_i,v_j} = 1\}$, and the document d_{G_l} for group G_l is formed by aggregating documents of all its members, i.e., $d_{G_l} = \cup\{d_{u_i}|I_{u_i,G_l} = 1\}$. Note that though our target is to learn the topic distributions of all users, votes, 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 vote is typically short-presented and topic-ambiguous; (2) even with user-level vote 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_l}'s to LDA model as training samples. Group-level vote content aggregation will cover all the content from affiliated users and help better

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

identify their interests in terms of vote topic. Essentially, we iteratively update document-topic distribution $\boldsymbol{\Theta}$ and topic-word distribution $\boldsymbol{\Phi}$ until convergence. By LDA, 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.

5.2. Semantic distillation

In this subsection, we introduce how to analyze users, votes, and groups in terms of semantic information. Word embedding is a language modeling technique, which maps words (or phrases) from the vocabulary into vectors of real numbers. Word embedding allows words with similar meanings to have closed representations. The Skip-Gram model is a well-known word embedding framework that finds word representations by predicting surrounding words in a document when there is a central word in a sliding window [18]. Typically, 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.

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

where k is (half) window size of the context words. The basic Skip-Gram model defines $p(w_i|w_t)$ using the softmax function as follows:

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

where $\mathbf{w}_i$ and $\mathbf{w}_t$ are the vector representation of context word w_i and center word w_t, respectively, and V is the vocabulary. In addition, techniques of hierarchical softmax or negative sampling are used during learning process [18] to improve the efficiency of computation.

5.3. Topic-enhanced word embedding

We noticed that the text of voting content is usually short and contains informal Internet lang. Even if we aggregate the votes into group-level documents, topic modeling methods like LDA still cannot infer the topic distribution of each vote. In fact, the Skip-Gram model for word embedding assumes that each word always preserves a single vector, so they are indiscriminate for homonym and polysemy that are typically common in vote questions. Considering these unique characteristics of vote texts, in this section, we propose a joint topic and semantic learning model, named *Topic-Enhanced Word Embedding* (TEWE). The basic idea of TEWE is that the prediction result of the central word w_t should contain the context words w_i as well as the topic information of the words z_i. In this way, words with different related topics and words in documents with different topics can have different embeddings.

The comparison between Skip-Gram and TEWE is shown in Fig. 4. Different from Skip-Gram that solely uses the center word w to predict context words, TEWE also jointly considers z_w^d, the most likely topic d that the document belongs to and z_w, the topic of the word in a document. Since the topic of each word z_w and topic distribution of each document $\boldsymbol{\Theta}_d$ have been derived in Section 5.1, we can calculate z_w^d as $z_w^d = \arg\max_z \theta_d^{(z)}$, where $\theta_d^{(z)}$ is the probability that document d belongs to topic z. Thus, TEWE considers 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:

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

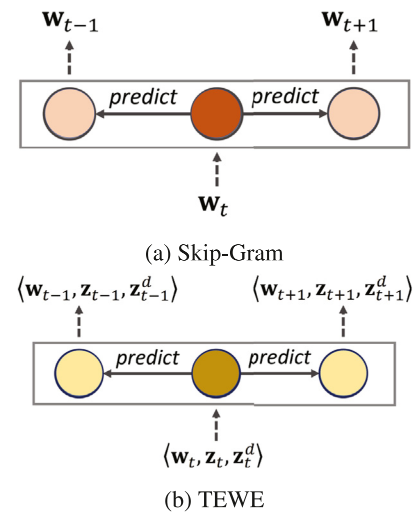


Fig. 4. Comparison between Skip-Gram and TEWE. The dark orange circles in (a) indicate the embeddings of original words, while the dark yellow circles in (b) indicate the TEWE representation of pseudo words, which preserves semantic and topic information of words and documents.

where $p(\langle w_i, z_i, z_i^d\rangle|\langle w_t, z_t, z_t^d\rangle)$ outputs a vector that represents the probability distributions over a list of predicted context words as follow:

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

Similar to the Skip-Gram model, we use the negative sampling technique to train the model. The objective function is to distinguish between the context word and the negative words drawn from noise distribution as follows:

$$\mathcal{O}(D) = \frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-k\le c\le k\\ c\neq 0}}\Bigg(\log\sigma(\mathbf{w}_{t+c}^{z,z^d\top}\mathbf{w}_t^{z,z^d}) + \sum_{i=1}^{d}\mathbb{E}_{n_i\sim P(w)}[\log\sigma(-\mathbf{w}_{n_i}^{z,z^d\top}\mathbf{w}_t^{z,z^d})]\Bigg), \tag{10}$$

where $\mathbf{w}_{t+c}^{z,z^d}$ is the context word and $\mathbf{w}_{n_i}^{z,z^d}$ is the negative words drawn from noise distribution $P(w)$. Note that compared with the softmax in Eq. (9) which involves all words in the vocabulary, each term in Eq. (10) only depends on the center word $\mathbf{w}_t^{z,z^d}$, the context word $\mathbf{w}_{t+c}^{z,z^d}$, and d negative samples, which greatly reduces the complexity for optimization. Given the objective in Eq. (10), the representation vector $\mathbf{w}^{z,z^d}$ for each pseudo word $\langle w, z_w, z_w^d\rangle$ can be learned by stochastic gradient ascent method:

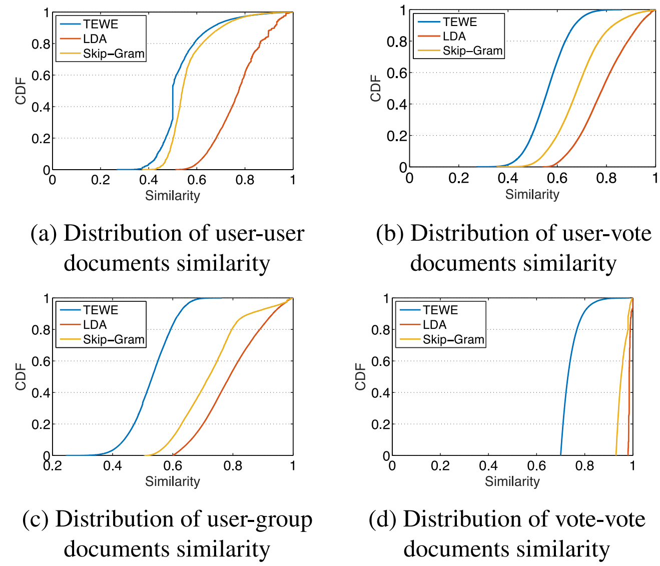
$$\mathbf{w}^{z,z^d} \leftarrow \mathbf{w}^{z,z^d} + \eta\cdot\frac{\partial\mathcal{O}(D)}{\partial\mathbf{w}^{z,z^d}}, \tag{11}$$

where η is the learning rate.

Once the TEWE representation of each pseudo-word is obtained, the representation of each document can be obtained by aggregating 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

$$\mathbf{e}_d = \sum_{w\in d}\text{TF-IDF}(w,d)\cdot\mathbf{w}^{z,z^d}, \tag{12}$$

Input : vote documents $\{d_{v_j}\}$, user-level documents $\{d_{u_i}\}$, group-level documents $\{d_{G_l}\}$.
Output: TEWE representation $\mathbf{e}_d$ for each document d.
for $d \in \{d_{v_j}\} \cup \{d_{u_i}\} \cup \{d_{G_l}\}$ **do**
    for $w \in d$ **do**
        Obtain w's topic assignment z_w according to Algorithm 1;
        Obtain d's topic distribution $\boldsymbol{\Theta}_d$ according to Algorithm 1;
        Calculate d's most likely topic $z_w^d = \arg\max_z \theta_d^{(z)}$;
        Create pseudo word $\langle w, z_w, z_w^d \rangle$;
    end
end
repeat
    for *each pseudo word* $\langle w, z_w, z_w^d \rangle$ **do**
        Update its TEWE representation $\mathbf{w}^{z,z^d}$ according to Eq. (11);
    end
until *convergence*;
for $d \in \{d_{v_j}\} \cup \{d_{u_i}\} \cup \{d_{G_l}\}$ **do**
    Calculate its TEWE representation $\mathbf{e}_d$ according to Eq. (12);
end
Algorithm 1: Calculating TEWE representation for each document



(a) Distribution of user-user documents similarity
(b) Distribution of user-vote documents similarity
(c) Distribution of user-group documents similarity
(d) Distribution of vote-vote documents similarity

Fig. 5. Comparison of the different level documents similarities based on the representations learned by TEWE, LDA, and Skip-Gram models.

where 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., TF-IDF$(w, d) = f_{w,d} \cdot \log \frac{|D|}{|d \in D: w \in d|}$ (D is the set of all documents). With TEWE document representations, we can measure inter-document similarities. For example, the similarity between 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 the topic and semantic proximity information of user documents, thereby implicitly revealing the similarity of voting interests between two users.

The procedure of calculating TEWE representation for each document is summarized in Algorithm 1. For each word w in each document d, we first obtain the topic assignment of this word z_w and the most likely topic of the document that the word belongs to z_w^d, according to Algorithm 1. Then, we learn the TEWE representation $\mathbf{w}^{z,z^d}$ for each pseudo word $\langle w, z_w, z_w^d \rangle$ according to Eq. (11). Finally, the TEWE representation $\mathbf{e}_d$ for each document d is calculated according to Eq. (12).

5.4. Voting content embedding methods analysis

To demonstrate the difference of learned document embeddings through TEWE, LDA, and Skip-Gram models, we calculate the pair-wise similarity (i.e., cosine distance) between user, vote and group documents. The similarity distance is ranged from 0 to 1, the smaller the value, the closer the two representations are. The similarity cumulative distribution is illustrated in Fig. 5. One can observe that the similarity of the document representations learned by TEWE is much lower than that learned by LDA and Skip-Gram. Typically, for vote documents, one can see that the cosine distance of representations learned through LDA, Skip-Gram, and TEWE, are started from 0.72, 0.91, and 0.96, respectively. This suggests that TEWE is more effective to characterize and differentiate the vote contents. Therefore, compared with LDA and Skip-Gram, through TEWE we can better capture voting topics and user interest in participation.

6. Recommendation models

As illustrated in 6, the framework of JTS-MF models mainly consist of two parts: the study of voting content analysis in the blue box and the voting recommendation in the orange box. The methods for vote content analysis have been thoroughly discussed in Section 5. In this section, we will introduce the detailed design of voting recommendations. Specifically, we propose two Joint Topic-Semantic-aware social Matrix Factorization (JTS-MF) models for online vote recommendation, named JTS-MF1 and JTS-MF2, which incorporate both hierarchical social relationship and topic-semantic similarities into vote recommendation in an explicit and implicit manner.

6.1. JTS-MF1 model

In this subsection, we present the details of JTS-MF1. JTS-MF1 model characterizes the interaction strength between each user and each vote directly. Specifically, in JTS-MF1, we first construct an influence index $E_{i,j}$ which describe user u_i's potential enthusiasm on vote v_j. To fairly evaluate $E_{i,j}$, we take the following three different factors into consideration: the similarity between u_i and v_j, the popularity of v_j among u_i's social-level friends, and the popularity of v_j among u_i's group-level friends. Next, we will introduce the definitions of these three factors in detail, respectively.

User–vote similarity. The similarity between user u_i and vote v_j is defined as the cosine distance of their TEWE representations:

$$S_{u_i,v_j} = \frac{\mathbf{e}_{u_i}^\top \mathbf{e}_{v_j}}{\|\mathbf{e}_{u_i}\|_2 \|\mathbf{e}_{v_j}\|_2}. \tag{13}$$

User–vote similarity measures the topic-semantic closeness of particular user and vote.

Social-level popularity factor. The social-level popularity factor of vote v_j with respect to user u_i is defined as

$$F_{u_i,v_j} = \sum_{u_k \in \mathcal{F}_i^+} I_{u_k,v_j} \sqrt{\frac{d_k^- + d}{d_i^+ + d_k^- + d}}, \tag{14}$$

where I_{u_i,u_k} indicates whether u_i follows u_k as described in Eq. (3), 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., $d_k^- = |\mathcal{F}_k^-|$), and d is the smoothing constant ($d = 1$ in this paper). $\sqrt{\frac{d_k^- + d}{d_i^+ + d_k^- + d}}$

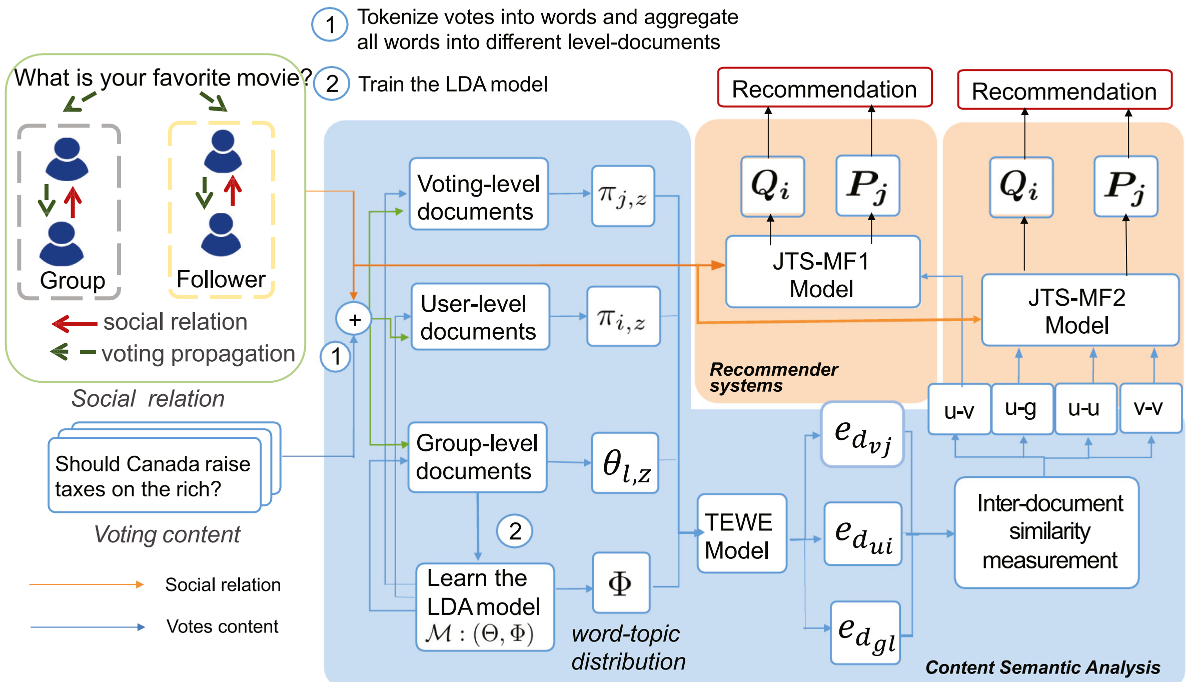


Fig. 6. The framework of joint-topic-semantic embedding and recommendation model. We propose two methods regarding how to utilize social relation hierarchical structure and vote content for online vote recommendation.

incorporates the information of local authority and local hub value to differentiate the importance of different users [64]. Essentially, F_{u_i,v_j} measures the popularity of vote v_j among all followees of user u_i.

Group-level popularity factor. The group-level popularity factor for vote v_j with respect to user u_i is defined as

$$G_{u_i,v_j} = \sum_{G_l \in \mathcal{G}_i} \sum_{u_k \in G_l, k \neq i} I_{u_k,v_j}, \tag{15}$$

where $\mathcal{G}_i$ is the set of groups that user u_i joins. G_{u_i,v_j} measures the popularity of vote v_j among group-level friends of user u_i.

Influence index $E_{i,j}$. We define the influence index between user u_i and vote v_j as

$$E_{i,j} = \mu S_{u_i,v_j} + \nu F_{u_i,v_j} + (1-\mu-\nu) G_{u_i,v_j}, \tag{16}$$

where μ and ν are the mixing parameters for balancing the personalized vote preference, social-level popularity and group-level popularity. Note that the influence index $E_{i,j}$ encodes both the topic-semantic similarity between user u_i and vote v_j as well as the popularity of vote v_j in user v_i's social structure.

6.1.1. Objective function

Taking the influence index into consideration, we aim to minimize the following objective function in JTS-MF1 model:

$$L = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I'_{i,j}(R_{i,j} - E_{i,j}\boldsymbol{Q_i}\boldsymbol{P_j}^\top)^2 + \frac{\lambda}{2}(\|\boldsymbol{Q}\|_F^2 + \|\boldsymbol{P}\|_F^2), \tag{17}$$

where $\boldsymbol{Q_i}$ and $\boldsymbol{P_j}$ are the latent feature of user u_i and vote v_j, respectively, and $I'_{i,j}$ is the training weights defined as

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

The reason we do not directly use I_{u_i,v_j} defined in Eq. (2) 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 vote v_j, and $E_{i,j}\boldsymbol{Q_i}\boldsymbol{P_j}^\top$ is the predicted value of $R_{i,j}$. Without loss of generality, in JTS-MF1 model (as well as the following JTS-MF2 model), we set $R_{i,j} = 1$ if u_i participates v_j and $R_{i,j} = 0$ otherwise. The second term of Eq. (17) is the regularization term to prevent overfitting, where λ is the regularization weight and $\|\cdot\|_F^2$ denotes the Frobenius norm. It should be noted that, different from traditional matrix factorization, we explicitly distinguish each user–vote pair (u_i, v_j) by attaching the influence index $E_{i,j}$, which could provide a finer guidance when optimizing the objective function and learning the latent representation of users and votes.

6.1.2. Learning algorithm

We perform the Alternating Least Square (ALS) method to update $\boldsymbol{Q}$ and $\boldsymbol{P}$ in Eq. (17). ALS fixes one of the unknowns variables ($\boldsymbol{Q}$ or $\boldsymbol{P}$) in each iteration, and recomputes the other by solving the least-squares problem. The gradient of the loss function in Eq. (17) with respect to $\boldsymbol{Q_i}$ and $\boldsymbol{P_j}$ are as follows:

$$\frac{\partial L}{\partial \boldsymbol{Q_i}} = \lambda \boldsymbol{Q_i} - \sum_{j=1}^{M} I'_{i,j}E_{i,j}\left(R_{i,j} - E_{i,j}\boldsymbol{Q_i}\boldsymbol{P_j}^\top\right)\boldsymbol{P_j}, \tag{19}$$

$$\frac{\partial L}{\partial \boldsymbol{P_j}} = \lambda \boldsymbol{P_j} - \sum_{i=1}^{N} I'_{i,j}E_{i,j}\left(R_{i,j} - E_{i,j}\boldsymbol{Q_i}\boldsymbol{P_j}^\top\right)\boldsymbol{Q_i}. \tag{20}$$

Setting the partial derivatives in Eqs. (19) and (20) to zero, we have

$$\boldsymbol{Q_i} = \frac{\sum_{j=1}^{M} I'_{i,j}E_{i,j}R_{i,j}\boldsymbol{P_j}}{\lambda + \sum_{j=1}^{M} I'_{i,j}E_{i,j}^2\boldsymbol{P_j}^\top\boldsymbol{P_j}}, \tag{21}$$

$$\boldsymbol{P_j} = \frac{\sum_{i=1}^{N} I'_{i,j}E_{i,j}R_{i,j}\boldsymbol{Q_i}}{\lambda + \sum_{i=1}^{N} I'_{i,j}E_{i,j}^2\boldsymbol{Q_i}^\top\boldsymbol{Q_i}}. \tag{22}$$

We list the pseudo code of the learning algorithm for JTS-MF1 as follows:

1. Initialize $\boldsymbol{Q}$ and $\boldsymbol{P}$ randomly;
2. In each iteration of the algorithm, do:
   (a) fix $\boldsymbol{P}$ and update every $\boldsymbol{Q_i}$ according to Eq. (21);
   (b) fix $\boldsymbol{Q}$ and update every $\boldsymbol{P_i}$ according to Eq. (22);
   until convergence.

6.2. JTS-MF2 model

JTS-MF1 model models the interaction strength between users and votes as the influence index directly. Different from JTS-MF1, in this subsection, we propose JTS-MF2 model, in which the inter-user similarities and inter-vote similarities are taken into consideration for vote recommendation. Motivated by Locally Linear Embedding [19], which tries to preserve the local linear dependency among inputs in the low-dimensional embedding space, we aim to keep inter-user and inter-vote similarities in latent feature space as well. To this end, in JTS-MF2 model, while the rating $R_{i,j}$ is factorized as user latent feature $\boldsymbol{Q_i}$ and vote latent feature $\boldsymbol{P_j}$, we deliberately enforce $\boldsymbol{Q_i}$ and $\boldsymbol{P_j}$ to be dependent on their social-topic-semantic similar counterparts, respectively.

6.2.1. Similarity coefficients

In order to characterize the influence of inter-user common interests and inter-vote 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 vote: $\widehat{T}_{j,t}$, where v_j and v_t are two distinct votes.

Generally speaking, in JTS-MF2, the latent feature $\boldsymbol{Q_i}$ for user u_i is tied up with the latent feature of his social-level friends and group-level friends who are weighted through the similarity coefficients $\widehat{S}_{i,k}$'s and $\widehat{G}_{i,k}$'s. Likewise, the latent feature $\boldsymbol{P_j}$ for vote v_j is tied up with the latent feature of its similar votes, which are weighted through the similarity coefficient $\widehat{T}_{j,t}$'s.

Normalized social-level similarity coefficient of users. Social-level similarity coefficient of users is represented by matrix $\boldsymbol{S}^{N\times N}$, 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

$$S_{i,k} = I_{u_i,u_k}\sqrt{\frac{d_k^- + d}{d_i^+ + d_k^- + d}} \cdot \frac{\mathbf{e}_{u_i}^\top \mathbf{e}_{u_k}}{\|\mathbf{e}_{u_i}\|_2\|\mathbf{e}_{u_k}\|_2}, \tag{23}$$

where $\frac{\mathbf{e}_{u_i}^\top \mathbf{e}_{u_k}}{\|\mathbf{e}_{u_i}\|_2\|\mathbf{e}_{u_k}\|_2}$ is the topic-semantic similarity between user u_i and user u_k, and all other notations are with the same meaning as in Eq. (14). Note that $S_{i,k}$ 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-MF2, which is defined as

$$\widehat{S}_{i,k} = \frac{S_{i,k}}{\sum_{k\in\mathcal{F}_i^+} S_{i,k}}, \tag{24}$$

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

Normalized group-level similarity coefficient of users. Group-level similarity coefficient of users is represented by matrix $\boldsymbol{G}^{N\times N}$, 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

$$G_{i,k} = \sum_{G_l\in\mathcal{G}} I_{u_i,G_l} I_{u_k,G_l} \cdot \frac{\mathbf{e}_{u_i}^\top \mathbf{e}_{G_l}}{\|\mathbf{e}_{u_i}\|_2\|\mathbf{e}_{G_l}\|_2}, \tag{25}$$

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, and the last term is the topic-semantic similarity between user u_i and group G_c. 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

$$\widehat{G}_{i,k} = \frac{G_{i,k}}{\sum_{k\in\mathcal{G}_i} G_{i,k}}, \tag{26}$$

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

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

$$T_{j,t} = \frac{\mathbf{e}_{v_j}^\top \mathbf{e}_{v_t}}{\|\mathbf{e}_{v_j}\|_2\|\mathbf{e}_{v_t}\|_2}. \tag{27}$$

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

$$\widehat{T}_{j,t} = \frac{T_{j,t}}{\sum_{t\in\mathcal{V}_j} T_{j,t}}. \tag{28}$$

6.2.2. Objective function

Using the notations listed above, the objective function of JTS-MF2 can be written as

$$\begin{aligned} L = {} & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I'_{i,j}\left(R_{i,j} - r_m - \boldsymbol{Q_i}\boldsymbol{P_j}^\top\right)^2 \\ & + \frac{\lambda}{2}\left(\|\boldsymbol{Q}\|_F^2 + \|\boldsymbol{P}\|_F^2\right) \\ & + \frac{\alpha}{2}\sum_{i=1}^{N}\left\|\boldsymbol{Q}_i - \sum_{k\in\mathcal{F}_i^+}\widehat{S}_{i,k}\boldsymbol{Q}_k\right\|^2 \\ & + \frac{\beta}{2}\sum_{i=1}^{N}\left\|\boldsymbol{Q}_i - \sum_{k\in\mathcal{G}_i}\widehat{G}_{i,k}\boldsymbol{Q}_k\right\|^2 \\ & + \frac{\gamma}{2}\sum_{j=1}^{M}\left\|\boldsymbol{P}_j - \sum_{t\in\mathcal{T}_j}\widehat{T}_{j,t}\boldsymbol{P}_t\right\|^2. \end{aligned} \tag{29}$$

The basic idea of the objective function in Eq. (29) lies in that, in addition to considering explicit feedback between users and votes, we also impose penalties on the discrepancy among features of similar users and similar votes. We give detailed explanation as follows.

The first term of Eq. (29) measures the mean squared error between prediction and ground truth, where $I'_{i,j}$ is the same as in Eq. (18). $R_{i,j}$ is the actual rating of user u_i on vote v_j, and $\boldsymbol{Q_i}\boldsymbol{P_j}^\top$ is the predicted value of $R_{i,j}$.

The second, third, and fourth terms of Eq. (29) measure the penalty of discrepancy among similar users and similar votes. In particular, the second term enforces user u_i's latent feature $\boldsymbol{Q_i}$ to be similar to the weighted average of his like-minded followees' profiles $\boldsymbol{Q_k}$. Weight $\widehat{S}_{i,k}$ addresses 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 $\boldsymbol{Q_i}$ to be similar to the weighted average of all his group peers' profiles $\boldsymbol{Q_k}$. Weight $\widehat{G}_{i,k}$ emphasizes 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 vote v_j's latent feature $\boldsymbol{P_j}$ to be similar to the weighted average of votes that share similar topic-semantic information with v_j.

The last term of Eq. (29) 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 vote similarities is controlled by the parameters α, β, and γ, respectively. Obviously, users' social-level similarity, users' group-level similarity, or votes' 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.

6.2.3. Learning algorithm

To solve the optimization in Eq. (29), we apply batch gradient descent approach to minimize the objective function.[7] The gradients of loss function in Eq. (29) with respect to each variable $\boldsymbol{Q_i}$ and $\boldsymbol{P_j}$ are as follows:

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{Q_i}} &= \sum_{j=1}^{M} -I'_{i,j}\left(R_{i,j} - \boldsymbol{Q_i}\boldsymbol{P_j}^{\top}\right)\boldsymbol{P_j} \\
&\quad + \alpha\Bigg(\Big(\boldsymbol{Q_i} - \sum_{k\in\mathcal{F}_i^+}\widehat{S}_{i,k}\boldsymbol{Q_k}\Big) + \sum_{t\in\mathcal{F}_i^-} -\widehat{S}_{t,i}\Big(\boldsymbol{Q_t} - \sum_{k\in\mathcal{F}_t^+}\widehat{S}_{t,k}\boldsymbol{Q_k}\Big)\Bigg) \\
&\quad + \beta\Bigg(\Big(\boldsymbol{Q_i} - \sum_{k\in\mathcal{G}_i}\widehat{G}_{i,k}\boldsymbol{Q_k}\Big) + \sum_{t\in\mathcal{U}} -\widehat{G}_{t,i}\Big(\boldsymbol{Q_t} - \sum_{k\in\mathcal{G}_i}\widehat{G}_{t,k}\boldsymbol{Q_k}\Big)\Bigg) + \lambda\boldsymbol{Q_i}, \\
\frac{\partial L}{\partial \boldsymbol{P_j}} &= \sum_{i=1}^{N} -I'_{i,j}\left(R_{i,j} - \boldsymbol{Q_i}\boldsymbol{P_j}^{\top}\right)\boldsymbol{Q_i} \\
&\quad + \gamma\Bigg(\Big(\boldsymbol{P_j} - \sum_{t\in\mathcal{V}_j}\widehat{T}_{j,t}\boldsymbol{P_t}\Big) + \sum_{k\in\mathcal{V}_j} -\widehat{T}_{k,j}\Big(\boldsymbol{P_k} - \sum_{t\in\mathcal{V}_k}\widehat{T}_{k,t}\boldsymbol{P_t}\Big)\Bigg) + \lambda\boldsymbol{P_j}.
\end{aligned} \tag{30}$$

To clearly understand the gradients in Eq. (30), it is worth pointing out that $\boldsymbol{Q_i}$ appears not only in the ith sub-term in the second and third terms of Eq. (29) explicitly, but also exists in other tth 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 $\boldsymbol{P_j}$. Given the gradients in Eq. (30), we list the pseudo code of the learning algorithm for JTS-MF2 as follows:

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

[7] Note that it is impractical to apply ALS method here because it requires calculating the inverse of two matrices with extremely large size.

7. Experiments

In this section, we evaluate our proposed JTS-MF1 and JTS-MF2 models on the aforementioned Weibo vote dataset.[8] We first introduce the evaluation metrics, baselines, and experiment setup in the experiments, and then present two case studies and the experiment results of JTS-MF1 and JTS-MF2.

7.1. Evaluation metrics

To quantitatively analyze the performance of vote 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. Their definition are as follows:

- *Recall@k* measures the coverage of relevant votes in recommendation result:
$$Recall@k = \frac{\sum_u \left|\{v|I_{u,v}=1, index_u(v)\le k\}\right|}{\sum_u \left|\{v|I_{u,v}=1\}\right|}, \tag{31}$$
where $index_u(v)$ is the rank index of vote v in the recommended vote set for user u.
- *Precision@k* measures the precision of relevant votes in recommendation result:
$$Precision@k = \frac{\sum_u \left|\{v|I_{u,v}=1, index_u(v)\le k\}\right|}{\sum_u k}. \tag{32}$$
- *micro-F1@k* is the harmonic mean of *Recall@k* and *Precision@k*:
$$micro\text{-}F1@k = \frac{2\times Recall@k \times Precision@k}{Recall@k + Precision@k}. \tag{33}$$

7.2. Baselines

We use the following methods as the baselines against JST-MF models. Note that the first three baselines are reduced versions of JST-MF1, which only consider one particular type of factors in the influence index $E_{i,j}$. The fourth to sixth baselines are reduced versions of JST-MF2, which only consider one particular type of similarity of users or votes.

- **JTS-MF1(C)** only considers user's similarity with vote content in the influence index, i.e., $\mu = 1, \nu = 0$ in JST-MF1 model.
- **JTS-MF1(S)** only considers social-level popularity of votes in the influence index, i.e., $\mu = 0, \nu = 1$ in JST-MF1 model.
- **JTS-MF1(G)** only considers group-level popularity of votes in the influence index, i.e., $\mu = 0, \nu = 0$ in JST-MF1 model.
- **ELMo-MF1** uses the same parameter settings as JTS-MF1 model except that it applies the ELMo model [56] to learn vote embeddings.
- **JTS-MF2(S)** only considers social-level similarity of users, i.e., $\beta, \gamma = 0$ in JTS-MF2 model.
- **JTS-MF2(G)** only considers group-level similarity of users, i.e., $\alpha, \gamma = 0$ in JTS-MF2 model.
- **JTS-MF2(V)** only considers similarity of votes, i.e., $\alpha, \beta = 0$ in JTS-MF2 model.
- **ELMo-MF2** uses the same parameter settings as JTS-MF2 model except that it applies the ELMo model to learn vote embeddings.
- **Basic-MF** [65] simply uses matrix factorization method to predict the user–vote matrix while ignores additional social relation, group affiliation and vote content information.

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

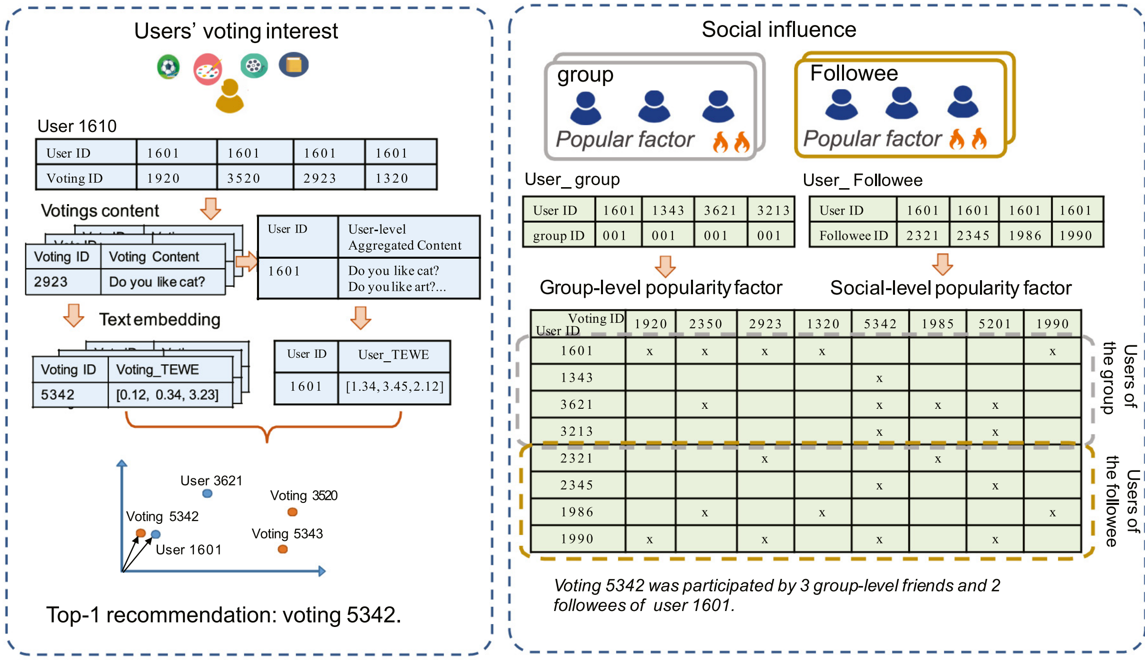


Fig. 7. The case study of JTS-MF1 model. JTS-MF1 characterizes the interaction strength between each user and each vote directly.

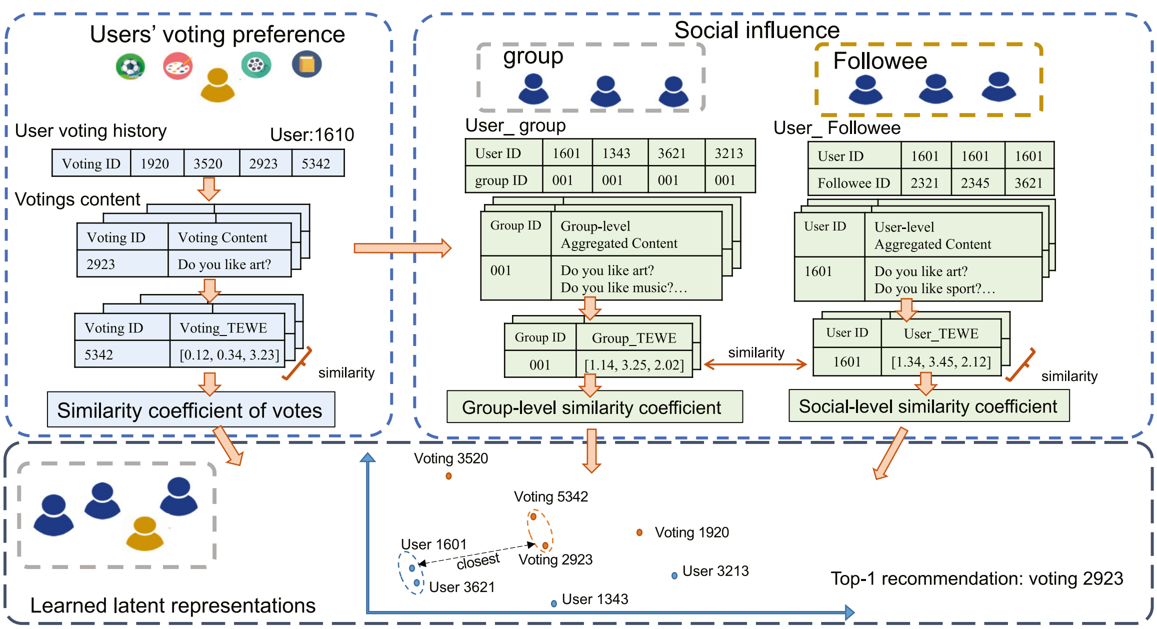


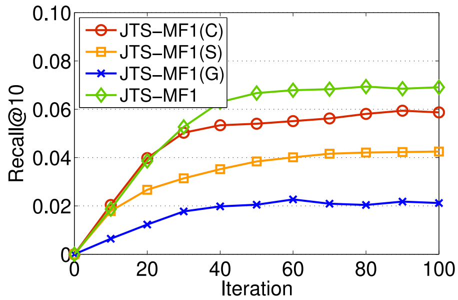
Fig. 8. The case study of JTS-MF2 model. JTS-MF2 preserves inter-user and inter-vote similarities during matrix factorization.

- **Topic-MF** [17] is similar as JTS-MF except that we substitute Θ_d for $\mathbf{e}_d$ when calculating similarities in Eq. (23), (25), and (27). Note that Θ_d can be viewed as the embedding of the document with respect to topics. Therefore, Topic-MF only considers the topic similarity among users and votes.
- **Semantic-MF** is similar as JTS-MF except that we use the Skip-Gram model in [18] directly to learn the word embeddings. Therefore, Semantic-MF only considers the semantic similarity among users and votes.
- **Neu-MF** [66] is a state-of-the-art neural network-based method, which replaces the inner product in matrix factorization by a neural architecture to model the interaction between users and items.
- **SRCMF** [67] is a state-of-the-art social and semantic-aware recommendation system that integrates the social network, item's reviews, and user's reviews in a unified convolutional neural network (CNN) based matrix factorization framework [68] for the recommendation.

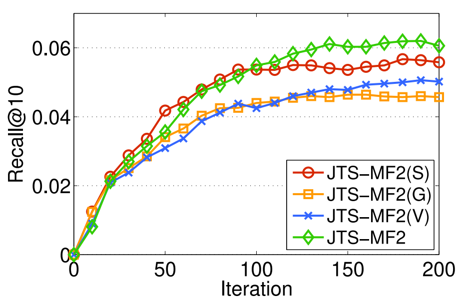
7.3. Experiment setup

The pre-processing steps of the vote dataset are as follows. We first split the text of each voting question into individual tokens. Then we use the Jieba package[9] to remove punctuations,

9 https://github.com/fxsjy/jieba.



(a) JTS-MF1



(b) JTS-MF2

Fig. 9. Convergence of JTS-MF models with respect to Recall@10.

numbers, and stop-words within the tokens. The vote-level, user-level and group-level documents can be obtained by aggregating related corpus from votes accordingly.

We use GibbsLDA++,[10] 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 JTS-MF1-based methods, we set regularization weight $\lambda = 0.5$. For all JTS-MF2-based methods, we set the learning rate $\delta = 0.001$ and $\lambda = 0.5$. Typically, we set $I_m = 0.01$ in Eq. (18). We run 100 iterations for each of the experiment cases for JTS-MF1-based methods, and 200 iterations for JTS-MF2-based methods. To conduct the recommendation task, we randomly select 20% of users' vote records in the dataset as test set and use the remaining data as the training examples for our JTS-MF model as well as all baselines. The choice of remaining hyper-parameters (mixing parameters μ, ν in JTS-MF1 method, trade-off parameters α, β, γ in JTS-MF2 method, and dimension of latent features *dim* in both JTS-MF methods) is discussed in Section 7.6.

7.4. Case studies

To intuitively understand the principles of JTS-MF1 and JTS-MF2 models, we carry out two corresponding case studies. Through these case studies, we demonstrate how the JTS-MF models characterize the essence of hierarchical social relationship and votes content in the vote recommendation.

The illustration of the case study for JTS-MF1 is shown Fig. 7. In this scenario, we aim to recommend a vote to user '1601'. To this end, we wish to characterize the strength between a user and a vote to provide an effective guidance for further matrix factorization. Specifically, in the lower right part of Fig. 7, we can observe that vote '5342' has been participated by 3 users in the same group and two followees of the user '1601'. Thus, the vote '5342' has a strong social connection with the user '1601'. In addition, we can find that the content similarity between the user '1601' and the vote '5342' is also high, since their representations are close in TEWE spaces as shown in the left part of Fig. 7. Therefore, JTS-MF1 model could infer that the user '1601' has a possibility to participate the vote '5342' in the future.

The case study illustration for JTS-MF2 is shown Fig. 8. In the upper right part of Fig. 8, one can observe that the user '1601' and the user '3621' are in the same group (i.e., '001'), and the user '1601' follows the user '3621'. Therefore, the social-level and group-level similarity coefficients between '1601' and '3621' are stronger than other user pairs. In fact, one can find that the user '1601' and the user '3621' like to participate in the same vote. JTS-MF2 captures the above social influence similarity and enforces the users with a strong social relationship to have similar representations in the learned latent space. Besides, by analyzing the vote texts, one can realize that the content of the vote '5342' and the vote '2923' are similar, JTS-MF2 thus preserves this information in the learned vote representations. Finally, JTS-MF2 recommends the vote '5342' to the user '1601', as they are the closest pair in the learned latent representation space.

7.5. Experiment results

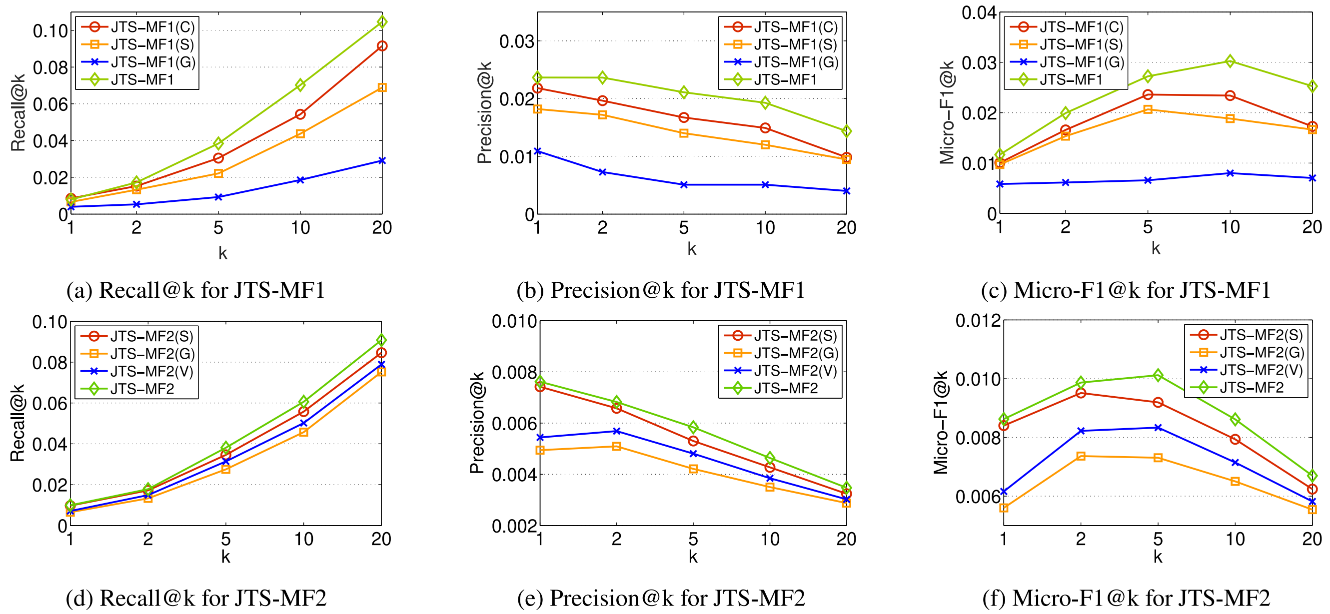
7.5.1. Study of convergence

To study the convergence of JTS-MF models, we set $\mu = 0.45$, $\nu = 0.5$ for JTS-MF1, and run the learning algorithm up to 100 iterations for JTS-MF1(C), JTS-MF1(S), JTS-MF1(G), and JTS-MF1, respectively. We also run up to 200 iterations for JTS-MF2(S) with $\alpha = 10$, JTS-MF2(G) with $\beta = 140$, JTS-MF2(V) with $\gamma = 30$, and JTS-MF2 with $\alpha = 10$, $\beta = 140$, $\gamma = 30$, respectively ($dim = 10$ for all JTS-MF models). The results of Recall@10 for JTS-MF1 and JTS-MF2 are presented in Figs. 9(a) and 9(b), respectively. From the results we can see that, the recall of JTS-MF1 rises rapidly in 40 iterations, and begins to oscillate slightly after around 80 iterations. The case is similar for JTS-MF2, except that the improvement of performance stagnates after around 150 iterations. Therefore, we set the number of learning iterations as 100 for JTS-MF1 model and 200 for JTS-MF2 model to achieve a balance between running time and performance of models.

7.5.2. Experiment results of JTS-MF1

In this subsection, we compare a JTS-MF1 model with its reduced versions to evaluate the efficacy of usage of vote content and social structure in the JTS-MF1 model. The parameter settings of μ, ν, and *dim* are the same as in Section 7.3. The results of recall@k, precision@k, and micro-F1@k are presented in Fig. 10(a), 10(b), and 10(c), respectively. From the results, we can conclude that JTS-MF1(C) achieves the best performance whereas JTS-MF1(G) performs worst among the three reduced versions. As aforementioned in Section 7.2, JTS-MF1(C) only considers the preference of users on vote contents, while JTS-MF1(S) or JTS-MF1(G) only considers the social-level or group-level popularity of votes. Therefore, these findings indicate that (1) user's voting interests are more influenced by the content of votes; (2) user's social-level structural information is more helpful than group-level structural information when determining users' voting interest. This is following 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 the group-level relationship. In addition, the results also show that JTS-MF1 model outperforms its three reduced versions, which

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



(a) Recall@k for JTS-MF1 (b) Precision@k for JTS-MF1 (c) Micro-F1@k for JTS-MF1

(d) Recall@k for JTS-MF2 (e) Precision@k for JTS-MF2 (f) Micro-F1@k for JTS-MF2

Fig. 10. Recall@k, Precision@k, and Micro-F1@k for JTS-MF1 and JTS-MF2 models.

demonstrates the ability of JTS-MF1 model in combining these three types of information.

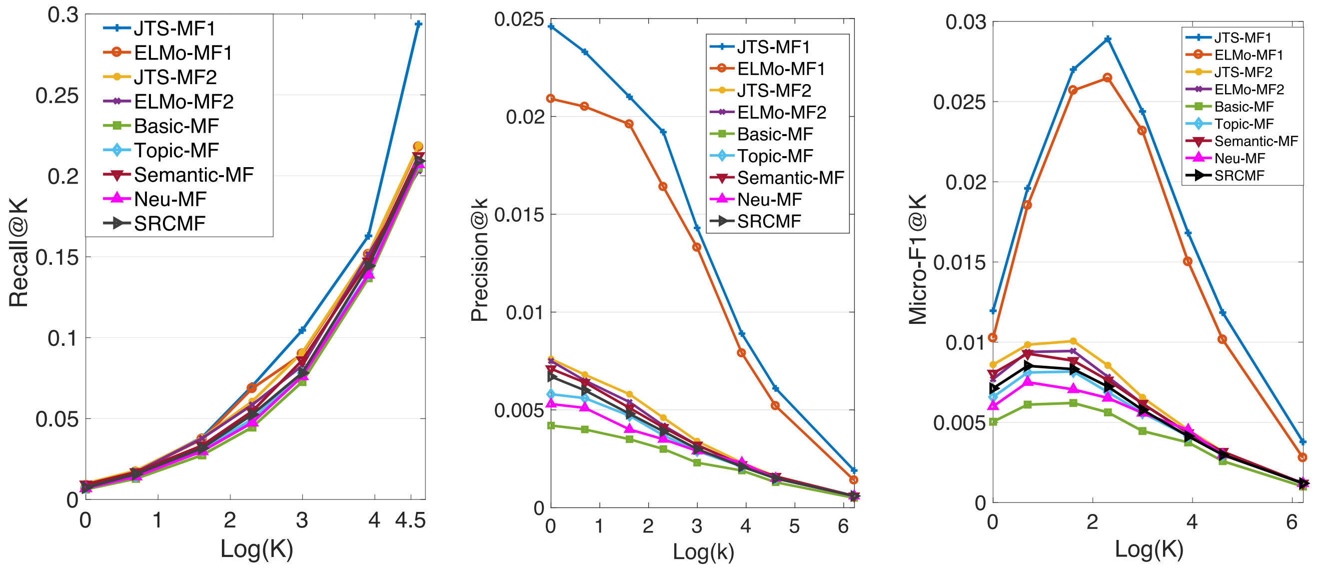
7.5.3. Experiment results of JTS-MF2

To study the performance of JTS-MF2 model and the effectiveness of three types of similarities, we run JTS-MF2 model as well as its corresponding three reduced versions and report the results in Fig. 10. The parameter settings of α, β, γ, and *dim* are the same as in Section 7.3. Fig. 10(d), 10(e), and 10(f) consistently demonstrate that JTS-MF2(S) performs best and JTS-MF2(G) performs worst among three types of reduced versions of JTS-MF2. Note that JTS-MF2(S) only considers users' social-level similarity and JTS-MF2(G) only considers users' group-level similarity. Therefore, it could be concluded that, in JTS-MF2 model, social-level friends are more helpful than group-level friends when determining users' voting interests, which is similar to the case in JTS-MF1 model. In addition, the result also demonstrates the effectiveness of the usage of votes' similarity. Finally, we can observe that JTS-MF2 model outperforms its three reduced versions in all cases, which proves that the three types of similarity can be well combined by JTS-MF2 model to achieve an even better result.

7.5.4. Comparison of models

To further compare JTS-MF1 and JTS-MF2 models with other baselines, we gradually increase the k from 1 to 500 and report the results in Fig. 11 with the best performance highlighted in bold. The value of μ, ν, α, β, and γ for JTS-MF and its reduced models are the same as in Section 7.3. 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 $\boldsymbol{Q_i}$ and $\boldsymbol{P_j}$ in all MF-based methods. The above parameter settings are the optimal results of fine tuning for given *dim*. From Fig. 11, we have several observations:

- MostPop performs worst among all methods, because MostPop simply recommends the most popular votes 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 votes. 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.
- ELMo is a dynamic word embedding model that generates embedding for each word considering its context. As shown in Fig. 11, ELMo-MF outperforms Topic-MF and Semantic-MF in terms of precision, recall, F1 scores, since ELMo model is more effective in learning representations for short texts. However, our proposed JST-MF still performs better than ELMo-MF. These results demonstrate that TEWE is able to distillate richer information from votes than ELMo.
- Neu-MF outperforming the basis-MF is especially evident (i.e., 30.19% improvement in Recall@1, 17% improvement in Recall@10). However, Neu-MF is inferior to the models which consider the auxiliary information (i.e., social network structure and vote content) into modeling. This phenomenon can be observed in the case of JTS-MF, ElMo-MF, Topic-MF and Semantic-MF. This is following our intuition since Neu-MF purely models the single-channel of user–item interaction information with deep learning method for the recommendation. Hence, this observation reveals that jointly considering users' social similarities and votes' similarities are pivotal in vote recommendation.
- Compared with SRCM, JTS-MF gains at least 0.0023 and 0.0111 improvement in Recall@1 and Recall@10 respectively. The main reason is that SRCM uses CNN to generate latent representations from item descriptions where words are represented by pre-trained word embeddings. This limits SRCM to characterize the voting content, since word embeddings do not distinguish between the homonyms and polysemous words that commonly appear in the voting questions. In addition, unlike JTS-MF, SRCM only considers user-level social relations (i.e., A follows B). This further proves the importance of properly incorporating unique patterns of vote propagation (i.e., structural social information) when making vote recommendation.
- JTS-MF1 and JTS-MF2 outperforms Topic-MF and Semantic-MF. This is the most important observation from Fig. 11,

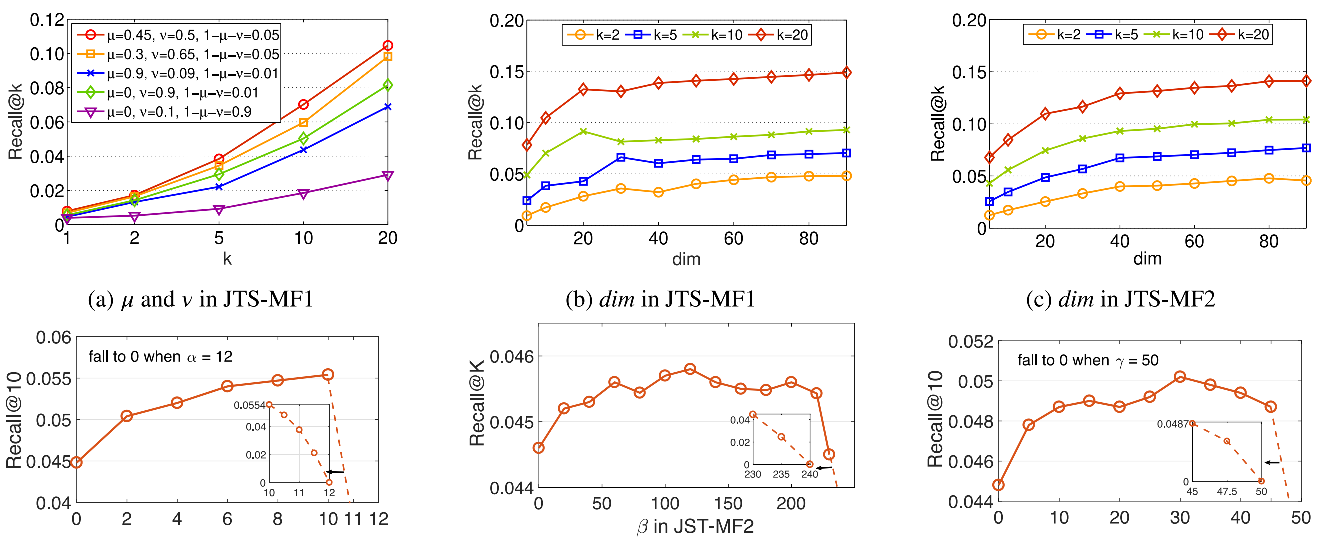


(a) Result of Recall@k for JTS-MF models and baselines.

(b) Result of Precision@k for JTS-MF models and baselines.

(c) Result of Micro-F1@k for JTS-MF models and baselines.

Fig. 11. Result of Recall@k, Precision@k, and Micro-F1@k for JTS-MF models and baselines.



(a) μ and ν in JTS-MF1

(b) *dim* in JTS-MF1

(c) *dim* in JTS-MF2

(d) α in JTS-MF2

(e) β in JTS-MF2

(f) γ in JTS-MF2

Fig. 12. Parameter sensitivity of JTS-MF1 and JTS-MF2.

since it justifies our aforementioned claim that joint-topic-semantic model can benefit from both topic and semantic aspects and achieve better performance.

- Basically, according to the results in Fig. 11, JTS-MF2 performs best among all the methods. Note that JTS-MF1 concentrates more on the interaction between users and votes, which proves that modeling user–vote interaction directly is more helpful than modeling user–user or vote–vote relations in Weibo vote scenario. However, it is worth noticing that, though JTS-MF1 outperforms JTS-MF2 in most cases, there is no much difference between these two models when measured by recall@k, especially when k is small (JTS-MF1 even achieves higher recall@1 than JTS-MF2). This finding demonstrates that both JTS-MF1 and JTS-MF2 have practical significance in real-world vote recommendation scenario.

7.6. Parameter sensitive analysis

We investigate the parameter sensitivity in this subsection. Specifically, we evaluate how different values of mixing parameters μ, ν, trade-off parameters α, β, γ, and latent feature dimensions *dim* can affect the results.

Parameters analysis in JST-MF1. We fix $dim = 10$, vary the value of μ and ν and report Recall@k in Fig. 12(a). From the result, we can see that the choices of mixing parameters can significantly affect the performance of JTS-MF1. For example, when $1 - \mu - \nu$ is relatively large ($1 - \mu - \nu = 0.05$ in Fig. 12(a)), increasing the value of μ can improve the performance of JTS-MF1 (from the yellow line to the red line). However, the case is opposite when $1-\mu-\nu$ is relatively small ($1-\mu-\nu = 0.01$ in Fig. 12(a)). Besides, the performance degrades seriously when $\mu = 0$, i.e., ignoring the similarity between users and votes totally in JTS-MF1. This study shows that all the three factors in the influence index $E_{i,j}$ play important roles in characterizing the relevance between users and votes, and the choices of mixing parameters should be carefully made since the interaction among these three factors is complicated. According to the result, Recall@10 reaches the maximum when $\mu = 0.45$ and $\nu = 0.5$. Therefore, we use these values as the parameter settings of JTS-MF1 in previous experiments.

Parameters analysis in JST-MF2. We fix $dim = 10$, keep two of the trade-off parameters as 0, and vary the value of the left trade-off parameter. Then we report *Recall@10* in Fig. 12(d), Fig. 12(e), and Fig. 12(f), respectively. As shown in Fig. 12(d), 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 votes, resulting in performance deterioration. The similar phenomenon are observed in Figs. 12(e) and 12(f), 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-MF2(S), JTS-MF2(G), and JTS-MF2(V), respectively, and use their combination as the parameter settings in JTS-MF2.

Impact of dimensionality. We fix $\mu = 0.45$, $\nu = 0.5$ in JTS-MF1, $\alpha = 10$, $\beta = 0$, $\gamma = 0$ in JTS-MF2, and tune the dimension of latent features of users and votes. The results are shown in Figs. 12(b) and 12(c), respectively. From the figures, we can see clearly that the performance is increasing when *dim* gets larger, this is because latent features with a larger number of dimensions have more capacity to encode users and votes. However, the performance drops when *dim* further increases, as too large dimension will introduce noises which mislead the subsequent recommendation. Moreover, a larger *dim* leads to more running time in experiments. On balance, we set $dim = 10$ in our experiment scenarios for both JTS-MF1 and JTS-MF2 to ensure the experiments can complete within reasonable time.

7.7. Experimental results discussion

Although many recent works have employed matrix factorization for the recommendation, they either separately leverage auxiliary information (e.g., textual descriptions of items and inner users relationships), or purely resort to deep learning to exact the user and item features from their interaction. To our knowledge, our proposed JTS-MF is the first one which studies how to jointly incorporate both **hierarchical social relationship** and **short text content** into matrix factorization in an explicit (i.e., JTS-MF1) and implicit (i.e., JTS-MF2) manner. We conduct experiments on the real application of Weibo voting which contains the user level follower/followee relations, group affiliation information and content of votes. From the experiment results, one can observe that, by properly leveraging auxiliary information, both JTS-MF1 and JTS-MF2 outperform the neural network-based method (i.e., NeuMF), as the latter one purely models the single-channel of user–item interaction information for the recommendation. This reveals that jointly considering users' social similarities and votes' similarities are pivotal in vote recommendation. In addition, we compare JTS-MF with a general social-semantic aware recommendation system (i.e. SRCM), the experiment results demonstrate that by considering the unique property of online voting, the domain specific recommendation system can achieve better performance.To study effective semantic analysis methods of voting content, we compare the performance of JST-MF with ELMo-MF, Topic-MF and Semantic-MF methods. The experimental results justify that our joint-topic semantic-model can benefit from both the topic and semantics aspect when processing short text, thereby achieving better performance. Finally, by analyzing the parameters sensitivities and comparing JTS-MF models with their reduced versions, one can realize that our proposed approaches have an ability to effectively combine hierarchical social relationship and short text content into vote recommendation.

8. Conclusions and future work

In this paper, we study the problem of recommending online votes to users in social networks. We justify the motivation and study how to jointly incorporate hierarchical social relationship and short text content into matrix factorization for vote recommendation. To overcome the limitations of the topic model and the semantic model when learning representations of vote contents, we propose Topic-Enhanced Word Embedding method to consider topics and semantics of words and documents jointly. Then we propose two Joint-Topic-Semantic-aware social Matrix Factorization models, i.e., JTS-MF1 and JTS-MF2, to learn latent features of users and votes based on the social network structure and TEWE representations. Specifically, JTS-MF1 model focuses on characterizing the interaction strength between users and votes directly, while JTS-MF2 model aims to preserve the inter-user and inter-vote similarities during matrix factorization. We conduct extensive experiments to evaluate JTS-MF models on Weibo vote dataset. The experimental results demonstrate the effectiveness of TEWE representations and validate the competitiveness of JTS-MF models against other strong state-of-the-art baselines.

In future, we are particularly interested in improving the representations of voting content by capturing sentiments in votes. Besides, we will study the extension of the JTS-MF model based on deep learning technology to replace inner matrix factorization with neural architectures that can learn arbitrary functions from user–vote interaction data. To promote the application of deep learning-based recommendation models on the real production lines, another emerging direction is to explore the potential of model interpretability to explain why a recommendation has been made.

CRediT authorship contribution statement

Jia Wang: Developed the theory, Conducted the experiments, Verified the analytical methods, Investigate the related works of Social Recommendation variants. **Hongwei Wang:** Developed the theory, Conducted the experiments, Verified the analytical methods, Investigate the related works of Social Recommendation variants. **Miao Zhao:** Developed the theory, Conducted the experiments, Verified the analytical methods, Investigate the related works of Social Recommendation variants, Supervised the findings of this work, Supervised the rest experiments. **Jiannong Cao:** Verified the analytical methods, Investigate the related works of Social Recommendation variants, Supervised the findings of this work, Supervised the first two experiments. **Zhuo Li:** Verified the analytical methods, Investigate the related works of Social Recommendation variants. **Minyi Guo:** Investigate the related works of Social Recommendation variants, Supervised the findings of this work, Supervised the first two experiments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by HK RGC Collaborative Research Fund Group Research Grant, Hong Kong (RGC No. C6030-18G) and HK RGC Collaborative Research Fund Group Research Grant, Hong Kong (RGC No. C5026-18G).

Jia Wang made major contributions to the journal version by introducing a new JTS-MF model (JTS-MF1) that is parallel to JTS-MF2 model proposed in CIKM'17 paper. All authors discussed the results and contributed to each version of manuscript.

References

[1] Y. Xiwang, L. Chao, Z. Miao, W. Hongwei, D. Hao, L. Yong, L. Yang, Z. Junlin, Collaborative filtering-based recommendation of online social voting, IEEE Trans. Comput. Soc. Syst. 4 (1) (2017) 1–13.
[2] S. Tang, J. Yuan, Optimizing ad allocation in social advertising, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 1383–1392.
[3] J. Ilieva, S. Baron, N.M. Healey, Online surveys in marketing research: Pros and cons, Int. J. Mark. Res. 44 (3) (2002) 361.
[4] B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, 2012, pp. 415–463.
[5] J. Gao, T. Zhou, Y. Hu, Bootstrap percolation on spatial networks, Sci. Rep. 5 (2015) 14662.
[6] H.-J. Hung, H.-H. Shuai, D.-N. Yang, L.-H. Huang, W.-C. Lee, J. Pei, M.-S. Chen, When social influence meets item inference, in: Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining, 2016, pp. 915–924.
[7] Q. Zhang, J. Wu, P. Zhang, G. Long, I.W. Tsang, C. Zhang, Inferring latent network from cascade data for dynamic social recommendation, in: Proceedings of the 16th IEEE ICDM International Conference on Data Mining, 2016, pp. 669–678.
[8] X. Wang, W. Lu, M. Ester, C. Wang, C. Chen, Social recommendation with strong and weak ties, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 5–14.
[9] H. Wang, M. Zhao, X. Xie, W. Li, M. Guo, Knowledge graph convolutional networks for recommender systems, in: The World Wide Web Conference, 2019, pp. 3307–3313.
[10] H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, Z. Wang, Knowledge-aware graph neural networks with label smoothness regularization for recommender systems, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 968–977.
[11] L. Lü, M. Medo, C.H. Yeung, Y.-C. Zhang, Z.-K. Zhang, T. Zhou, Recommender systems, Phys. Rep. 519 (1) (2012) 1–49.
[12] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.
[13] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, F. Yang, Recommendation model based on opinion diffusion, Europhys. Lett. 80 (6) (2007) 680–693.
[14] Y.-C. Zhang, M. Blattner, Y.-K. Yu, Heat conduction process on community networks as a recommendation model, Phys. Rev. Lett. 99 (15) (2007) 154301.
[15] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J.R. Wakeling, Y.-C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. 107 (10) (2010) 4511–4515.
[16] L.-J. Chen, Z.-K. Zhang, J.-H. Liu, J. Gao, T. Zhou, A vertex similarity index for better personalized recommendation, Phys. A 466 (2017) 607–615.
[17] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[18] M. Tomas, S. Ilya, C. Kai, C. Greg, D. Jeffrey, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013, pp. 3111–3119.
[19] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, science 290 (5500) (2000) 2323–2326.
[20] A. Van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in: Proceedings of the 27th Annual Conference on Neural Information Processing Systems, 2013, pp. 2643–2651.
[21] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, J. Wang, Dynamic attention deep model for article recommendation by learning human editors' demonstration, in: Proceedings of the 23rd ACM International Conference on Knowledge Discovery and Data Mining, 2017, pp. 2051–2059.
[22] X. Wang, R. Donaldson, C. Nell, P. Gorniak, M. Ester, J. Bu, Recommending groups to users using user-group engagement and time-dependent matrix factorization, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016, pp. 1331–1337.
[23] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining, 2016, pp. 353–362.
[24] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, F. Zhang, A hybrid collaborative filtering model with deep structure for recommender systems, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence, 2017, pp. 1309–1315.
[25] Z. Xia, Y. Dong, G. Xing, Support vector machines for collaborative filtering, in: Proceedings of the 44th Annual Southeast Regional Conference, 2006, pp. 169–174.
[26] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann machines for collaborative filtering, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 791–798.
[27] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, J. Han, Personalized entity recommendation: A heterogeneous information network approach, in: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, 2014, pp. 283–292.
[28] G. Zhou, C. Song, X. Zhu, X. Ma, Y. Yan, X. Dai, H. Zhu, J. Jin, H. Li, K. Gai, Deep interest network for click-through rate prediction, in: Proceedings of the 24th ACM International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1059–1068.
[29] T. Ebesu, B. Shen, Y. Fang, Collaborative memory network for recommendation systems, in: Proceedings of the 41st International ACM Conference on Research and Development in Information Retrieval, 2018, pp. 515–524.
[30] X. Jiang, B. Hu, Y. Fang, C. Shi, Multiplex memory network for collaborative filtering, in: Proceedings of the 20th International Conference on Data Mining, 2020, pp. 91–99.
[31] X. Zhou, C. Mascolo, Z. Zhao, Topic-enhanced memory networks for personalised point-of-interest recommendation, in: Proceedings of the 25th ACM International Conference on Knowledge Discovery and Data Mining, 2019, pp. 3018–3028.
[32] B. Hu, C. Shi, W.X. Zhao, P.S. Yu, Leveraging meta-path based context for top-n recommendation with a neural co-attention model, in: Proceedings of the 24th ACM International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1531–1540.
[33] J. Li, B. Ge, D. Zhao, J. Jiang, Q. Zhao, Meta-path-based weapon-target recommendation in heterogeneous combat network, IEEE Syst. J. 13 (4) (2019) 4433–4441.
[34] X. Li, J. She, Collaborative variational autoencoder for recommender systems, in: Proceedings of the 23rd ACM International Conference on Knowledge Discovery and Data Mining, 2017, pp. 305–314.
[35] F. Zhou, Y. Mo, G. Trajcevski, K. Zhang, J. Wu, T. Zhong, Recommendation via collaborative autoregressive flows, Neural Netw. 126 (2020) 52–64.
[36] B. Pang, M. Yang, C. Wang, A novel top-N recommendation approach based on conditional variational auto-encoder, in: Proceedings of the 23th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2019, pp. 357–368.
[37] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1235–1244.
[38] F. Xue, X. He, X. Wang, J. Xu, K. Liu, R. Hong, Deep item-based collaborative filtering for top-n recommendation, ACM Trans. Inf. Syst. 37 (3) (2019) 1–25.
[39] F. Fessahaye, L. Perez, T. Zhan, R. Zhang, C. Fossier, R. Markarian, C. Chiu, J. Zhan, L. Gewali, P. Oh, T-recsys: A novel music recommendation system using deep learning, in: Proceedings of the 2019 IEEE International Conference on Consumer Electronics, 2019, pp. 1–6.
[40] Q. Zhang, J. Lu, D. Wu, G. Zhang, A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities, IEEE Trans. Neural Netw. Learn. Syst. 30 (7) (2018) 1998–2012.
[41] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 417–426.
[42] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, S. Yang, Social recommendation with cross-domain transferable knowledge, IEEE Trans. Knowl. Data Eng. 27 (11) (2015) 3084–3097.
[43] W. Fan, Y. Ma, D. Yin, J. Wang, J. Tang, Q. Li, Deep social collaborative filtering, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 305–313.
[44] X. Wang, X. He, M. Wang, F. Feng, T. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd International ACM Conference on Research and Development in Information Retrieval, SIGIR, 2019, pp. 165–174.
[45] X. Wang, Y. Liu, G. Zhang, F. Xiong, J. Lu, Diffusion-based recommendation with trust relations on tripartite graphs, J. Stat. Mech. Theory Exp. 2017 (8) (2017) 43–55.
[46] L.-J. Chen, J. Gao, A trust-based recommendation method using network diffusion processes, Physica A 506 (2018) 679–691.

[47] W. Hua, Z. Wang, H. Wang, K. Zheng, X. Zhou, Short text understanding through lexical-semantic analysis, in: Proceedings of the 31st International Conference on Data Engineering, 2015, pp. 495–506.
[48] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, Q. Liu, Shine: Signed heterogeneous information network embedding for sentiment link prediction, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 592–600.
[49] G. Guibon, M. Ochs, P. Bellot, From emojis to sentiment analysis, in: Proceedings of 2016 Workshop Affect Compagnon Artificiel Interaction, 2016.
[50] N. Colnerič, J. Demsar, Emotion recognition on twitter: Comparative study and training a unison model, IEEE Trans. Affect. Comput. (2018).
[51] J. Oliva, J.I. Serrano, M.D. del Castillo, Á. Iglesias, Symss: A syntax-based measure for short-text semantic similarity, Data Knowl. Eng. 70 (4) (2011) 390–405.
[52] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, ACL, 2013, pp. 1631–1642.
[53] R. Florian, A. Ittycheriah, H. Jing, T. Zhang, Named entity recognition through classifier combination, in: Proceedings of the 7th Conference on Natural Language Learning At Human Language Technologies, 2003, pp. 168–171.
[54] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bošnjak, S. Riedel, emoji2vec: Learning emoji representations from their description, in: Proceedings of the 4th International Workshop on Natural Language Processing for Social Media, ACL, 2016, pp. 48–54.
[55] F. Barbieri, F. Ronzano, H. Saggion, What does this emoji mean? A vector space skip-gram model for twitter emojis, in: Proceedings of the 10th International Conference on Language Resources and Evaluation, ELRA, 2016.
[56] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 2227–2237.
[57] W.X. Zhao, J. Jiang, H. Yan, X. Li, Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid, in: Proceedings of the 10th Conference on Empirical Methods in Natural Language Processing, 2010, pp. 56–65.
[58] Q. Mei, X. Ling, M. Wondra, H. Su, C. Zhai, Topic sentiment mixture: Modeling facets and opinions in weblogs, in: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 171–180.
[59] Q. Diao, M. Qiu, C.-Y. Wu, A.J. Smola, J. Jiang, C. Wang, Jointly modeling aspects, ratings and sentiments for movie recommendation, in: Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining, 2014, pp. 193–202.
[60] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 1188–1196.
[61] D. Chen, C.D. Manning, A fast and accurate dependency parser using neural networks, in: Proceedings of the 14th Conference on Empirical Methods in Natural Language Processing EMNLP, 2014, pp. 740–750.
[62] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.
[63] Y. Liu, Z. Liu, T.-S. Chua, S. Maosong, Topical word embeddings, in: Proceedings of the 29th Conference on Artificial Intelligence, 2015, pp. 2418–2424.
[64] H. Ma, H. Yang, M.R. Lyu, I. King, Sorec: Social recommendation using probabilistic matrix factorization, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, 2008, pp. 931–940.
[65] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[66] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[67] X. Wang, X. Yang, L. Guo, Y. Han, F. Liu, B. Gao, Exploiting social review-enhanced convolutional matrix factorization for social recommendation, IEEE Access 7 (2019) 82826–82837.
[68] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 233–240.