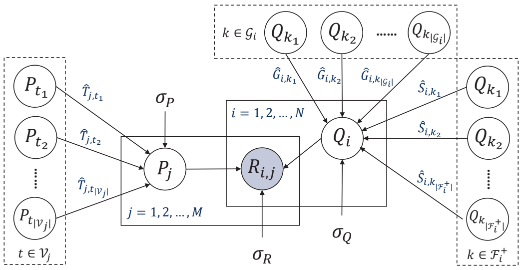


Fig. 5: Graphic Model of JTS-MF.

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

6.1 Similarity Coefficients

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

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

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

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

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

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

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

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

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

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

$$G_{i,k} = \sum\nolimits_{G\in\mathcal{G}} I_{u_i,G} \cdot I_{u_k,G} \cdot \frac{\mathbf{e}_{u_i}^\top \mathbf{e}_G}{\|\mathbf{e}_{u_i}\|_2 \|\mathbf{e}_G\|_2}, \tag{13}$$

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

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

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

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

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

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

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

6.2 Objective Function

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

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

The basic idea of the objective function in Eq. (17) lies in that, besides considering explicit feedback between users and votings, we also impose penalties on the discrepancy among features of similar users and similar votings. We give detailed explanation as follows. The first term of Eq. (17) measures the mean squared error between prediction and ground truth, where $I'_{i,j}$ is the training weights defined as

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

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

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

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

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

6.3 Learning Algorithm

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

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

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

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

(1) Randomly initialize $\boldsymbol{Q}$ 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 configurable learning rate.

7 EXPERIMENTS

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

7.1 Baselines

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

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

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

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

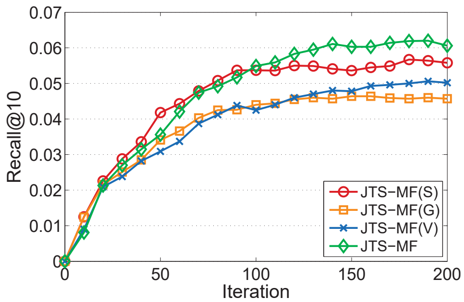


Fig. 6: Convergence of JTS-MF models with respect to *Recall@10.*

7.2 Parameter Settings

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

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

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

7.3 Experiment Results

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

7.3.2 Study of JTS-MF. To study the performance of JTS-MF model and the effectiveness of three types of similarities, we run JTS-MF model as well as its three reduced versions on Weibo voting dataset, and report the results of *Recall*, *Precision*, and *Micro-F1* in Fig. 7. The parameter settings of α, β, γ, and dim are the same as in Section 7.3.1. Fig. 7a, 7b, and 7c consistently demonstrate that JTS-MF(S) performs best and JTS-MF(G) performs worst among three types of reduced versions of JTS-MF. Note that JTS-MF(S) only considers users' social-level similarity and JTS-MF(G) only considers users' group-level similarity. Therefore, it could be concluded that social-level friends are more helpful than group-level friends when determining users' voting interest. This is in accordance with our intuition, since a user typically has much more group-level friends than social-level friends, which inevitably dilutes its effectiveness and brings noises into group-level relationship. In addition, the result in Fig. 7 also demonstrates the effectiveness of the usage of votings' similarity. Furthermore, it can be evidently observed that JTS-MF model outperforms its three reduced versions in all cases, which proves that the three types of similarities are well combined in JTS-MF model to achieve much better results.

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

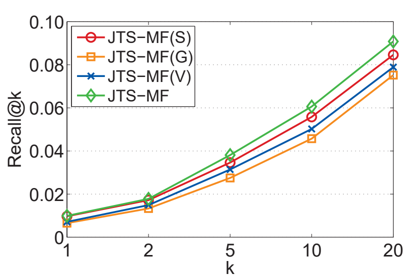
- MostPop performs worst among all methods, because MostPop simply recommends the most popular votings to all users without considering users' specific interests.
- Topic-MF and Semantic-MF outperforms Basic-MF, which proves the usage of similarities with respect to topic and semantic helpful for recommending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

7.4 Parameter Sensitivity

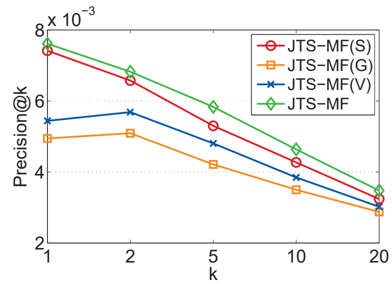
We investigate parameter sensitivity in this subsection. Specifically, we evaluate how different value of trade-off parameters α, β, γ, and different numbers of latent feature dimensions *dim* can affect the performance.

7.4.1 Trade-off parameters. We fix $dim = 10$, keep two of the trade-off parameters as 0, and vary the value of the left trade-off

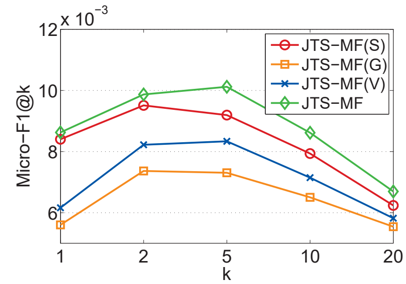
[8]GibbsLDA++: http://gibbslda.sourceforge.net



(a)

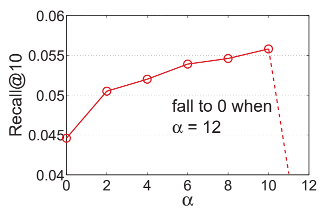


(b)

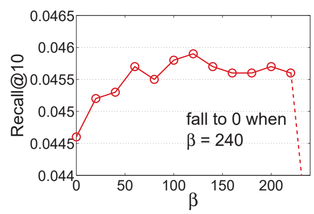


(c)

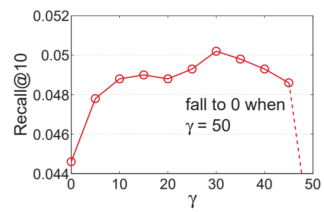
Fig. 7: (a) *Recall@k*, (b) *Precision@k*, and (c) *Micro-F1@k* of JTS-MF models.



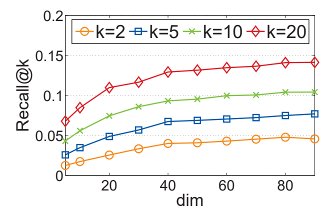
(a)



(b)



(c)



(d)

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

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

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

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

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

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

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

8 CONCLUSIONS

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

ACKNOWLEDGMENTS

This work was partially sponsored by the National Basic Research 973 Program of China under Grant 2015CB352403, the NSFC Key Grant (No. 61332004), PolyU Project of Strategic Importance 1-ZE26, and HK-PolyU Grant 1-ZVHZ.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. In *Journal of Machine Learning Research*. 993–1022.

[2] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (2013).

[3] Marco Bressan, Stefano Leucci, Alessandro Panconesi, Prabhakar Raghavan, and Erisa Terolli. 2016. The Limits of Popularity-Based Recommendations, and the Role of Social Ties. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 745–754.

[4] Danqi Chen and Christopher D Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks.. In *EMNLP*. 740–750.

[5] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 193–202.

[6] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2015. Content-aware point of interest recommendation on location-based social networks.. In *AAAI*. 1721–1727.

[7] Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. In *Proceedings of National Academy of Sciences (PNAS)*. 5228–5235.

[8] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[9] Liang Hu, Jian Cao, Guandong Xu, Longbing Cao, Zhiping Gu, and Can Zhu. 2013. Personalized recommendation via cross-domain triadic factorization. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 595–606.

[10] Hui-Ju Hung, Hong-Han Shuai, De-Nian Yang, Liang-Hao Huang, Wang-Chien Lee, Jian Pei, and Ming-Syan Chen. 2016. When Social Influence Meets Item Inference. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 915–924.

[11] Janet Ilieva, Steve Baron, and Nigel M Healey. 2002. Online surveys in marketing research: Pros and cons. *International Journal of Market Research* 44, 3 (2002), 361.

[12] Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2015. Social recommendation with cross-domain transferable knowledge. *IEEE Transactions on Knowledge and Data Engineering* 27, 11 (2015), 3084–3097.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[14] Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference on machine learning*. 331–339.

[15] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML*, Vol. 14. 1188–1196.

[16] Wu-Jun Li, Dit-Yan Yeung, and Zhihua Zhang. 2011. Generalized latent factor models for social network analysis. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain*. 1705.

[17] Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*. Springer, 415–463.

[18] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Sun Maosong. 2015. Topical word embeddings. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI)*. AAAI, 2418–2424.

[19] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 931–940.

[20] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 171–180.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*. 3111–3119.

[22] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 273–282.

[23] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[24] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*. ACM, 791–798.

[25] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. 2012. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 139–146.

[26] Shaojie Tang and Jing Yuan. 2016. Optimizing Ad Allocation in Social Advertising. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 1383–1392.

[27] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[28] Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. 2016. Recommending groups to users using user-group engagement and time-dependent matrix factorization. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 1331–1337.

[29] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social Recommendation with Strong and Weak Ties. In *Proceedings of the 25th ACM CIKM*. ACM, 5–14.

[30] Zhonghang Xia, Yulin Dong, and Guangming Xing. 2006. Support vector machines for collaborative filtering. In *Proceedings of the 44th annual Southeast regional conference*. ACM, 169–174.

[31] Xiwang Yang, Chao Liang, Miao Zhao, Hongwei Wang, Hao Ding, Yong Liu, Yang Li, and Junlin Zhang. 2017. Collaborative filtering-based recommendation of online social voting. *IEEE Transactions on Computational Social Systems* 4, 1 (2017), 1–13.

[32] Qin Zhang, Jia Wu, Peng Zhang, Guodong Long, Ivor W Tsang, and Chengqi Zhang. 2016. Inferring Latent Network from Cascade Data for Dynamic Social Recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 669–678.

[33] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 56–65.

[34] Hengshu Zhu, Enhong Chen, Kuifei Yu, Huanhuan Cao, Hui Xiong, and Jilei Tian. 2012. Mining personal context-aware preferences for mobile users. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 1212–1217.