Celebrity Recommendation with Collaborative Social Topic Regression

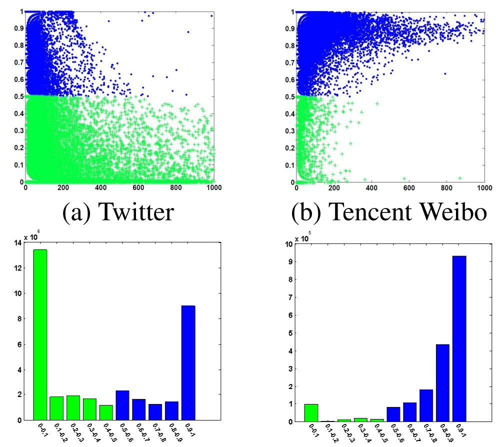
Xuetao Ding[†] **Xiaoming Jin**[†] **Yujia Li**[‡] **Lianghao Li**[§]
[†]School of Software, Tsinghua University, Beijing, China
[‡]Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
[§]Hong Kong University of Science and Technology, Hong Kong, China
helius.thu@gmail.com xmjin@tsinghua.edu.cn yujiali@cs.toronto.edu lianghaoli@yahoo.com

Abstract

Recently how to recommend celebrities to the public becomes an interesting problem on the social network websites, such as Twitter and Tencent Weibo. In this paper, we proposed a unified hierarchical Bayesian model to recommend celebrities to the general users. Specifically, we proposed to leverage both social network and descriptions of celebrities to improve the prediction ability and recommendation interpretability. In our model, we combine topic model with matrix factorization for both social network of celebrities and user following action matrix. It works by regularizing celebrity factors through celebrity's social network and descriptive words associated with each celebrity. We also proposed to incorporate different confidences for different dyadic contexts to handle the situation that only positive observations exist. We conducted experiments on two real-world datasets from Twitter and Tencent Weibo, which are the largest and second largest microblog websites in USA and China, respectively. The experiment results show that our model achieves a higher performance and provide more effective results than the state-of-art methods especially when recommending new celebrities. We also show that our model captures user intertests more precisely and gives better recommendation interpretability.

1 Introduction

Social networks have become one of the symbols of today's Internet [Ahmad, 2011]. With the explosive growth of the number of users in social network websites, the social network itself becomes a new media to propagate news, ideas and opinions. Nowadays, more and more users regard social networks as a powerful media in which they can gain the newest ideas and opinions from authorities and elites [Friedkin, 2006; Kwak *et al.*, 2010]. The public are increasingly interested in the celebrities and famous organizations. From figure 1, which is analyzed statistically using the dataset from



(a) Twitter (b) Tencent Weibo

(c) Twitter (d) Tencent Weibo

Figure 1: For (a)(c), 200,000 users were sampled, each point (x,y) represents a general user, where x is the number of followees and y is the percentage of celebrities in this user's followees. For (b)(d), all users are used, the height of each bar represents the number of users with corresponding percentage of celebrities in their followees.

Twitter[1] and Tencent Weibo[2], we can observe that more and more people are following celebrities and using social networks as a media. Though user behaviors of these two microblogs are different, we observe that a large number of users tend to follow celebrities in common, where nearly 40% users in Twitter and 90% users in Tencent Weibo prefer following celebrities. Celebrities and organizations have already recognized the eagerness of public to gain newest information which they are interested in and signed up their own accounts to broadcast their opinions in succession.

Though those celebrities are a small part of users in a social network, the quantity of them can still reach a relatively large order of magnitude. General users now face a big problem: how to find those celebrities they truly interested in to follow from massive candidates [Zimmerman *et al.*, 2002]. To solve this problem, the door to the employment of recommendation methods is opened. However, s-

[1]http://an.kaist.ac.kr/traces/WWW2010.html
[2]http://www.kddcup2012.org/c/kddcup2012-track1

ince we can only observe the user follow actions from microblogs, and unfollow does not mean not interested, conventional collaborative filtering techniques [Koren *et al.*, 2009; Su and Khoshgoftaar, 2009] can not be directly applied to address such problem. Inspired by one-class collaborative filtering techniques [Hu *et al.*, 2008], we utilize user following actions as positive samples with high confidence and no following actions as negative samples with low confidence.

Why are so many users interested in celebrities? The reason may be that most users are not only interested in those celebrities themselves, but also the interests behind those celebrities. In this paper, we propose to use the relationships between celebrities as a relationship network of interests from users. And we utilize the relationships between celebrities to boost the performance of recommendation. For example, the number of connections between singers is much larger than that between singers and entrepreneurs. That means one singer may have more common interests with another singer than entrepreneurs. Moreover, the number of connections between different groups varies largely, e.g., number of connections between sports stars and singers is far more than the number between singers and entrepreneurs. So it is necessary to analyze the structure of the social network among celebrities. Unlike analysis of common user's social network, which devotes giant efficiency cost due to the huge number of users, analyzing celebrity's social network will not suffer from such problem because of the relatively small quantity of celebrities. The two kinds of social networks have several great differences: (1) General user's social network are huge and sparse, but celebrity's social network is small and compact; (2) To some extent, the social network of celebrities can be viewed as the relationships among different kinds of interests, general user's social network may mainly represents friendship; and (3) Celebrities are the mainly followed groups on social network, but general users are not. To the best of our knowledge, this is the first work which aims at exploring the celebrities' social network to improve the performance of recommendation system.

We further leverage the semantic analysis power [Blei *et al.*, 2003] to solve hard explainability problem exsiting in the context of matrix factorization. We collect abundant personal descriptions for each celebrity from Wikipedia, LinkedIn and their personal websites etc. A model which combines social network analysis and semantic analysis to jointly learn the user's interests is proposed. More specifically, we used a hierarchical Bayesian model to combine topic model with matrix factorization for both celebrities social network and user interests matrix.

Our contributions are as follows: We are the first to explore whether celebrity's social network has an effect on improving the performance and efficiency of recommending celebrities. We propose the Collaborative Social Topic Regression (CSTR), a novel Bayesian hierarchical model which incorporates latent factor model on both celebritiy side and user side with topic model. Our method provides a single unified framework to handle both cold and warm-start scenarios. We provide a scalable, linearly complexity model fitting procedure through coordinate ascent optimization. We test our method on two real-world social network datasets from Twitter and Tencent Weibo. The experiment results show that our method significantly outperforms the existing and commonly used methods, especially for cold-start situations. And our method captures user interests more accurately with better interpretability. We also studied user behaviours difference between Twitter and Tencent Weibo.

2 Related Work

Although our proposed CSTR is the first one which aims at jointly modeling users' interests, celebrity social network and their semantic information, our model is related to previous research in the following areas.

Latent Factor Models: Latent factor models which are used in the context of recommendation systems have been extensively studied. PMF and BPMF [Salakhutdinov and Mnih, 2008b; 2008a] incorporate the latent factor models into a Bayesian framework. Some feature based latent factor models are also proposed in previous works, such as RLFM [Agarwal and Chen, 2009] and Factorization Machine [Rendle, 2010]. However, these methods can not be directly applied to the situation in which ratings have only two states : observed or not. [Hu *et al.*, 2008] addressed such problem via considering unobserved data as negative samples with low confidence. In our situation, those unfollow actions are regarded as unobserved data and modeled as negative samples with low confidence.

Latent Factor Models meet Topic Models: Topic models are used to discover a set of "topics" from a large collection of documents, where a topic is a distribution over terms. The representative works include PLSA [Hofmann, 1999] and LDA [Blei *et al.*, 2003]. For both latent factor models and topic models, they propose to reduce original data into latent spaces. Since a latent factor can be treated as a real-value variable and topics fall into a simplex, there are some works to link them together. CTM-PPMF and LDA-MPMF [Shan and Banerjee, 2010] equal item latent vector and topic proportion vector by Logistic transformation. fLDA [Agarwal and Chen, 2010] uses topic assignments to contribute the rating directly. CTR [Wang and Blei, 2011] uses topic vector to control the prior distribution for item latent factor. Our proposed model CSTR can be seen as an extention of CTR. If not considering social network of celebrities, CSTR is equivalent to CTR. We use CTR as a baseline to verify whether celebrity's social network is useful or not.

Latent Factor Models meet social network: Recently, some works have studied the effectiveness of social networks of general users to better address general user's interest [Purushotham *et al.*, 2012; Ma *et al.*, 2008; 2011; Yang *et al.*, 2011]. [Shen and Jin, 2012] incorporated heterogeneity and diversity of user social relationships. [Cheng *et al.*, 2012] utilizes user social relationships to recommend locations. The most relevant work to ours is [Purushotham *et al.*, 2012], which fuses LDA with social matrix factorization to obtain a consistent and compact feature representation. This work is different from ours since they only considered the situation when topic model plays a part in user interest matrix factorization, however we let topic model functioned not only on user ratings but also on celebrity social network,

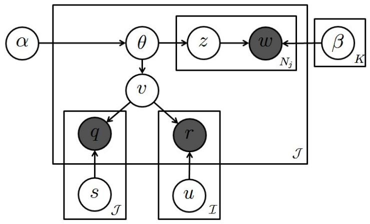


Figure 2: Graphical representation of CSTR. Celebrity latent factor v is controlled by celebrity topic proportion θ, which is generated from LDA. And v also affects the follow action records q and r simultaneously together with celebrity social latent factor s and user latent factor u.

which will gain better interpretability and higher performance of recommendation. As discussed in the previous section, social network of general users and that of celebrities are quite different. So, in this work we only consider the social networks of celebrities because we mainly focus on how social network of celebrities would have an effect on the task for celebrity recommendation.

3 Models

3.1 Notations

The latent factors for user i and celebrity j are denoted by u_i and v_j respectively. In celebrities' social network, the extra social latent factor for celebrity m is denoted by s_m. Written in matrix form, the three latent factor matrices are $U \in \mathbb{R}^{K\times\mathcal{I}}$, $V \in \mathbb{R}^{K\times\mathcal{J}}$ and $S \in \mathbb{R}^{K\times\mathcal{J}}$ where K is the dimensionality of the latent space, $\mathcal{I}$ is the number of users and $\mathcal{J}$ is the number of celebrities. Each latent factor is a column in the corresponding matrix. Throughout this section we use i to index users, j and m to index celebrities. We use matrix R to represent the user-celebrity follow table, $r_{ij} = 1$ if user i followed celebrity j, and $r_{ij} = 0$ otherwise. The social network among celebrities is a directed graph $\mathcal{Q} = (\mathcal{V}, \mathcal{E})$. The edge set $\mathcal{E}$ is represented as matrix Q, where $q_{mj} = 1$ when there is an edge from m to j and $q_{mj} = 0$ otherwise. Description of celebrity j is denoted by word sequence W_j.

3.2 Model Details

In social networks, a user following a celebrity means the user is interested in the celebrity and his/her work. On the other hand, when a user is not following a celebrity, it does not necessarily mean the user is not interested in the celebrity, because it is very likely that the user is simply not aware of that celebrity or the user might be a newly registered user. To express this asymmetry, we model the conditional distribution of r_{ij} given u_i and v_j as a Gaussian $\mathcal{N}(r_{ij}|u_i^\top v_j, c_{ij}^{-1})$, and the precision parameter c_{ij} is asymmetric for the two cases

$$c_{ij} = \begin{cases} a, \ r_{ij} = 1, \\ b, \ r_{ij} = 0. \end{cases}$$

According to our analysis, we set $a > b$ to express our confidence about a 'follow' action and uncertainty when the user is not following the celebrity. The probability of the full follow table R given U and V is assumed to be factorial, $p(R|U,V) = \prod_{i=1}^{\mathcal{I}} \prod_{j=1}^{\mathcal{J}} \mathcal{N}(r_{ij}|u_i^\top v_j, c_{ij}^{-1})$.

We place zero-mean spherical Gaussian prior on u_i following [Dueck and Frey, 2004; Salakhutdinov and Mnih, 2008b] so $p(U|\lambda_u) = \prod_{i=1}^{\mathcal{I}} \mathcal{N}(u_i|0, \lambda_u^{-1} I_K)$, where $I_K \in \mathbb{R}^{K\times K}$ is the identity matrix. As did in CTR, we place θ-mean Gaussian prior on v_j to enable the topic model to regulate the celebrity latent factors. We have $p(V|\theta,\lambda_v) = \prod_{j=1}^{\mathcal{J}} \mathcal{N}(v_j|\theta_j, \lambda_v^{-1} I_K)$. λ_u and λ_v are the precision of the corresponding Gaussian distribution. θ_j is the topic proportion vector generated by the topic model based on the text descriptions we collected for celebrity j.

The likelihood of the text descriptions under the topic model is a product of likelihood of each W_j, $p(W,\theta|\alpha,\beta) = \prod_{j=1}^{\mathcal{J}} \prod_{n=1}^{N_j} \ln \sum_{k=1}^{K} \theta_{jk}\beta_{k,w_{jn}}$, which is the same as in LDA [Blei *et al.*, 2003]. We fix the hyperparameter $\alpha = 1$ to keep the computation simple.

Now we turn to the model for celebrities' social network, represented by social matrix Q. The model is similar to the model for R, and V is again used as a latent factor matrix, so that the social network can help regularize V. We model the conditional distribution of q_{mj} as a Gaussian $\mathcal{N}(q_{mj}|s_m^\top v_j, d_{mj}^{-1})$. d_{mj} is the precision parameter, which again can represent asymmetry in our confidence for q_{mj}. We distinguish three cases here: (1) $q_{mj} = 1$ then we should be very confident that celebrity m is interested in j; (2) $q_{mj} = 0$ but $q_{jm} = 1$, then we get modest confidence; (3) both q_{mj} and q_{jm} are 0, then our confidence is low. Therefore we have the following definition for d_{mj}

$$d_{mj} = \begin{cases} e, \ q_{mj} = 1, \\ f, \ q_{mj} = 0 \text{ and } q_{jm} = 1, \\ g, \ q_{mj} = 0 \text{ and } q_{jm} = 0. \end{cases}$$

where we enforce $e > f > g$ to represent our different confidence for different cases. Then $p(Q|V,S) = \prod_{j=1}^{\mathcal{J}} \prod_{m=1}^{\mathcal{J}} \mathcal{N}(q_{mj}|s_m^\top v_j, d_{mj}^{-1})$, and we also place a zero mean Gaussian prior on celebrity social latent factor s_m, so that $p(S|\lambda_s) = \prod_{m=1}^{\mathcal{J}} \mathcal{N}(s_m|0, \lambda_s^{-1} I_K)$.

Finally, the joint likelihood of data, i.e. R, Q and W, and the latent factors U, V, S and θ under the full model is

$$\begin{aligned} &p(R,Q,W,U,V,S,\theta|\lambda_u,\lambda_v,\lambda_s,\beta) \\ =&p(U|\lambda_u)p(V|\theta,\lambda_v)p(S|\lambda_s)p(R|U,V)p(Q|V,S)p(W,\theta|\beta) \end{aligned} \tag{1}$$

The graphical model of CSTR is shown in Figure 2.

3.3 Parameter Learning and Optimization

Given a training data set, we want to find the Maximum a Posteriori (MAP) estimate of U, V, S, θ, so we can use U and V to predict the missing entries in R and use the predictions to do recommendation. In our model, finding the MAP is equivalent to maximize the log likelihood

$$\begin{aligned} \mathcal{L} = &-\sum_{i=1}^{\mathcal{I}} \sum_{j=1}^{\mathcal{J}} \frac{c_{ij}}{2}(r_{ij} - u_i^\top v_j)^2 - \frac{\lambda_u}{2}\sum_{i=1}^{\mathcal{I}} u_i^\top u_i \\ &- \frac{\lambda_v}{2}\sum_{j=1}^{\mathcal{J}}(v_j - \theta_j)^\top(v_j - \theta_j) - \sum_{m=1}^{\mathcal{M}} \sum_{j=1}^{\mathcal{J}} \frac{d_{mj}}{2}(q_{mj} - s_m^\top v_j)^2 \\ &- \frac{\lambda_s}{2}\sum_{m=1}^{M} s_m^\top s_m + \sum_{j=1}^{\mathcal{J}} \sum_{n=1}^{N_j} \ln\left(\sum_{k=1}^{K} \theta_{jk}\beta_{k,w_{jn}}\right) + \mathcal{C} \end{aligned} \tag{2}$$

Algorithm 1 Coordinate Ascent Optimization Algorithm

Require:
$\{r_{ij}\}$, $\{q_{mj}\}$, initial estimate $U, V, S, \theta_{1:\mathcal{J}}, \beta_{1:K}$,
hyper parameters $\lambda_u, \lambda_v, \lambda_s, a, b, e, f, g$.
Ensure:
$U, V, S, \theta_{1:J}$ maximize $p(U, V, S, \theta_{1:J} \mid R, Q, W)$.
Give matrix V a warm start, $V \leftarrow \theta_{1:\mathcal{J}}$.
while not convergent **do**
  $X \leftarrow VV^\top$
  for i=0 to $\mathcal{I}$ **do**
    A $\leftarrow b \cdot X + \lambda_u I + \sum_{j \in \{r_{ij}=1\}} (a-b) \cdot v_j v_j^\top$
    $u_i \leftarrow A^{-1} \sum_{j \in \{r_{ij}=1\}} a \cdot v_j$
  for m=1 to $\mathcal{M}$ **do**
    A $\leftarrow g \cdot X + \lambda_s I + \sum_{j \in \{q_{mj}=1\}} (e-g) \cdot v_j v_j^\top$
    A $\leftarrow$ A $+ \sum_{j \in \{q_{mj}=0, q_{jm}=1\}} (f-g) \cdot v_j v_j^\top$
    $s_m \leftarrow A^{-1} \sum_{j \in \{q_{mj}=1\}} e \cdot v_j$
  Y $\leftarrow b \cdot UU^\top + g \cdot SS^\top + \lambda_v I$
  for j=1 to $\mathcal{J}$ **do**
    A $\leftarrow$ Y $+ \sum_{i \in \{r_{ij}=1\}} (a-b) \cdot u_i u_i^\top$
    A $\leftarrow$ A $+ \sum_{m \in \{q_{mj}=1\}} (e-g) \cdot s_m s_m^\top$
    A $\leftarrow$ A $+ \sum_{m \in \{q_{mj}=0, q_{jm}=1\}} (f-g) \cdot s_m s_m^\top$
    $x \leftarrow \lambda_v \theta_j + \sum_{i \in \{r_{ij}=1\}} a \cdot u_i + \sum_{m \in \{q_{mj}=1\}} e \cdot s_m$
    $v_j \leftarrow A^{-1} x$
    update θ via simplex projection gradient
  update β using LDA M-step [optional]

where $\mathcal{C}$ is a constant. We optimize this function using coordinate ascent which alternatively optimizes latent factor variables u_i, v_j, s_m and the simplex variables θ_j which is similar to [Wang and Blei, 2011].

We directly set the derivative of $\mathcal{L}$ with respect to u_i, v_j and s_m to zero. Then we obtain

$$u_i = (\lambda_u I_K + V C_i V^\top)^{-1} V C_i R_i \tag{3}$$

$$s_m = (\lambda_s I_K + V D_m V^\top)^{-1} V D_m Q_m \tag{4}$$

$$\begin{aligned} v_j =& (\lambda_v I_K + U C_j U^\top + S D_j S^\top)^{-1} (\lambda_v \theta_j \\ &+ U C_j R_j + S D_j Q_j) \end{aligned} \tag{5}$$

where $C_i \in \mathbb{R}^{\mathcal{J} \times \mathcal{J}}$ is a diagonal matrix with c_{ij} as its diagonal elements and $R_i = (r_{i1}, ..., r_{i\mathcal{J}})^\top$ is the record of user i in the follow table R, $C_j \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ and $R_j \in \mathbb{R}^{\mathcal{I}}$ are similarly defined for celebrity j. $D_m \in \mathbb{R}^{\mathcal{J} \times \mathcal{J}}$ is a diagonal matrix with d_{mj} as its diagonal elements and $Q_m = (q_{m1}, ..., q_{m\mathcal{J}})^\top$, $D_j \in \mathbb{R}^{\mathcal{J} \times \mathcal{J}}$ and $Q_j \in \mathbb{R}^{\mathcal{J} \times 1}$ are similarly defined.

Iterating through the update equations (3), (4) and (5), we can see that the new U and S latent factors will depend on the celebrity latent factor matrix V. Based on current U, S and θ, we can compute the new V. Note that the topic proportion matrix θ affects both U and S through V, and the parameter λ_v controls how much we trust the topic model.

Given U, V and S, we update θ using projected gradient ascent [Bertsekas, 1999]. After a full iteration of coordinate ascent, we update β using the variational M-step developed for LDA [Blei *et al.*, 2003].

The detailed algorithm is described in Algorithm 1.

Using the learned parameters U^*, V^*, S^*, $\theta^*_{1:K}$ and β^*, we make predictions by $r^*_{ij} \approx (u^*_i)^\top v^*_j$.

3.4 Computational Complexity

The computational bottleneck here is computing $UC_jU^\top$, $SD_jS^\top$, $VC_iV^\top$ and $VD_mV^\top$. The naive calculation requires time $O(K^2\mathcal{I})$ and $O(K^2\mathcal{J})$ for each user and celebrity respectively. Inspired by [Hu *et al.*, 2008], we can rewrite $UC_jU^\top$ as

$$UC_jU^\top = U(C_j - bI)U^\top + bUU^\top, \tag{6}$$

where I is the corresponding identity matrix. Note that we can pre-compute $bUU^\top$, and $(C_j - bI)$ has only $\mathcal{I}_a$ non-zeros elements, where $\mathcal{I}_a$ represents the number of general users who followed the celebrity j, and empirically $\mathcal{I}_a \ll \mathcal{I}$. This sparsity can significantly speed up computation. For other matrix products, we can use similar tricks.

The final time complexity for one full iteration of coordinate ascent is $O(2K\mathcal{I}(\frac{K}{2} + \overline{\mathcal{E}_{uc}}))$, where $\overline{\mathcal{E}_{uc}}$ is the average number of celebrities a user followed. In practice, each user would only follow a limited number of celebrities, so $\overline{\mathcal{E}_{uc}}$ is small. Therefore this new algorithm is significantly faster than the naive algorithm.

We can see that three factors determine the complexity of our algorithm: 1) the dimensionality of the latent space; 2) the number of users in the network; 3) sparsity of the network. Both 1 and 3 are independent of 2, so our algorithm scales linearly with the number of general users in the network.

4 Experiments

4.1 Experiment Settings

We used the social network from Twitter and Tencent Weibo. User following records and celebrity social network are extracted. Text descriptions of 916 celebrities from Twitter are collected from Wikipedia, LinkedIn and their personal websites by searching with their names and short descriptions. For Tencent Weibo, descriptions are given in the dataset. To show the effectiveness of celebrity social network, we used the full celebrity network and only sampled a subset of 10,000 users from the full set of users. When sampling, users who follow less than 5 celebrities are not considered, because these data are very noisy and of high variance.

Twitter: 916 celebrities and 10,000 users. 96,929 user follow actions are observed, density of user actions matrix is about 1.06%. 45406 edges exist in celebrity social network, density of celebrity matrix is about 5.41%.

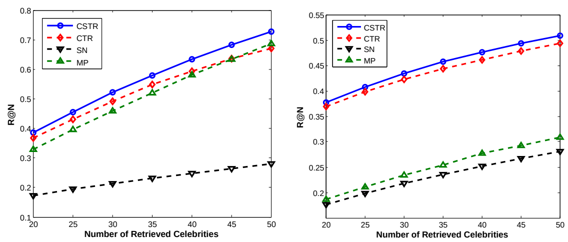
Tencent Weibo: 4394 celebrities and 10,000 users. 278,825 user follow actions are observed, density of user actions matirx is about 0.63%. 154,317 edges exist in celebrity social network, density of celebrity matrix is about 0.80%.

Evaluation Metrics: Recall and Average Precision(AP) are used, (1)Recall@N: $\frac{\text{follow number in top-N list}}{\text{total follow number}}$, where follow number means the number of celebrities a user followed; (2)AP@N: $\frac{\sum_{n=1}^{N} precision(n) * rel(n)}{\text{total follow number}}$ [Manning *et al.*, 2008], where $rel(n)$ is a 0-1 binary variable which indicates follow or not follow action. Both Recall and AP are averaged across users. They measure the inclusiveness and ranking performance of recommendation algorithms respectively.

In the experiments, CTR, social neighborhood (SN) and most popular (MP) methods are used as baselines. SN rec-

	Warm-Start		Cold-Start	
	Twitter	Tencent Weibo	Twitter	Tencent Weibo
CSTR	**11.056**	**21.135**	**7.899**	**14.095**
CTR	10.400	20.900	0.234	1.078
SN	5.117	6.191	3.347	6.371
MP	8.699	6.646	–	–

Table 1: AP@20 is reported for both warm-start and cold-start situation. Our model CSTR performed best on all 4 situations.



(a) Twitter (b) Tencent Weibo

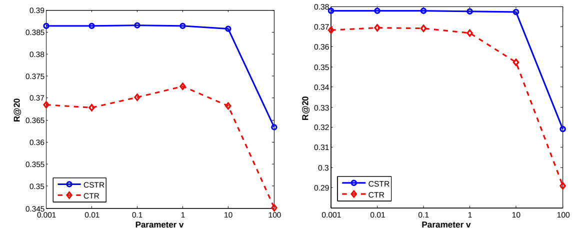
Figure 3: Recall results reported on warm-start situation. The number of retrieved celebrities varies from 20 to 50.

ommends target user the celebrities who have the most connections with the ones followed by target user. MP recommends users those celebrities who have the largest number of followers. Both Recall and AP are averaged over five random repeats in experiments. We used a parameter tuning strategy that does grid search on one parameter while fixing other parameters, which is done in turn for each parameter.

4.2 Warm-Start Situation

For each user we evenly split their records into 3 folds, and iteratively use each fold as the test set and the others as the training set. The results we report here is the average over 3 folds. Using the tuning strategy described in Section 4.1, we set $\lambda_u = \lambda_v = 0.01$, $a = 1$, $b = 0.1$ which works well for both CTR and CSTR on both datasets. For CSTR, extra parameters are set as $K = 10$, $e = 10$, $f = 2$, $g = 1$ on Twitter dataset and $K = 100$, $e = 1$, $f = 0.2$, $g = 0.1$ on Tencent Weibo dataset using the same tuning strategy. Figure 3 and Table 1 shows that CSTR model outperformed the other models consistently both in Twitter and Tencent Weibo. We believe the performance gain of CSTR over CTR comes from the model of celebrity social networks. Note that on Twitter data set the simple baseline MP can sometimes be even better than CTR. This shows that the celebrity social network do provide us important information about user's interests.

Then we studied how parameter λ_v affects the recommendation effectiveness. λ_v is the parameter which controls the impact of topic model. For CTR, we observe that when increasing λ_v, recall would increase and then fall down on both datasets. For CSTR, as impact of topic model becomes larger ($\lambda_v = 100$), performance starts to decrease. Though it seems that topic model contribute little to performance directly, it does contribute to user interests mining via affecting celebrity and user latent factors and it can bring better interpretability as shown in section 4.5.



(a) Twitter (b) Tencent Weibo

Figure 4: Comparison of Recall@20 varying λ_v.

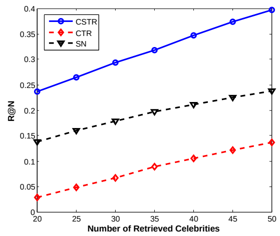
4.3 Cold-Start for Recommending New Celebrities

After a new celebrity starts to use a social network service, it usually takes a long time for those interested general users to get connected to him/her because of the slow process of information propagation in sparsely connected social networks. We call this a "cold-start" situation. It's valuable to recommend these celebrities to the public so that they can get involved in the social network faster. We tested the performance of the proposed CSTR model under this situation, where we preserve the new celebrities' connections to other celebrities, which are likely to come from the offline friend circle; but we don't use any follow records from general users, to simulate the cold-start setting. All celebrities are evenly grouped into 3 folds. We iteratively use each fold as test set and the others as training set, therefore at least one third of the celebrities are not in the training set, and the user follow records for them are not available during training. The final results are averaged over 3 folds.

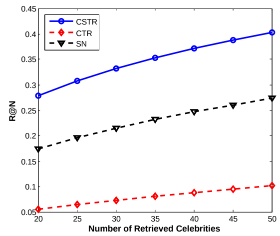
In this cold-start situation, it is impossible to use history record of user follow data. So it is necessary to increase the impact of social factors. To tune the e, f and g parameters for the social factors, we choose a base value of the three to be $e = 1$, $f = 0.2$, $g = 0.1$ respectively which works well in the warm-start situation, then scale them up by a factor of 10 every time. The relationship between the scale factor and the performance of the model is shown in Figure 6. We found $e = 100000$, $f = 20000$, $g = 10000$ to work well, which matches our intuition that we should put more weight on the social factors. Figure 5 shows that our model CSTR significantly outperforms neighborhood and CTR model. From Figure 5(a), recall@20 for CSTR on Twitter is 23.68% and for neighborhood method is 13.73%, an improvement of 72%. From Figure 5(b), recall@20 for CSTR on Tencent Weibo is 27.8% and for neighborhood method is 17.33%, an improvement of 60%. CTR performs poorly in this situation. For ranking metric AP, CSTR also has the best performance, which improving 136% and 121% over SN on Twitter and Tencent Weibo respectively, as shown in Table 1. The results show the effectiveness of celebrities network to help improve recommendation performance, and social matrix factorization performs better than the neighborhood method.

4.4 Performance for Different User Groups

We wanted to explore the effectiveness for recommending celebrities to different users who have different number of followees. We divided the users into 7 groups according to

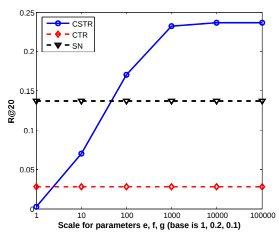


(a) Twitter

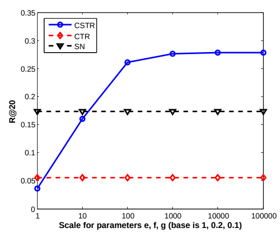


(b) Tencent Weibo

Figure 5: Recall results reported on cold-start situation

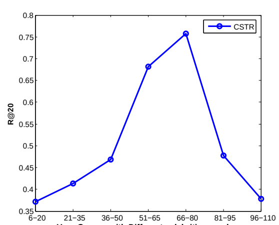


(a) Twitter

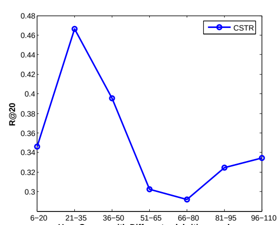


(b) Tencent Weibo

Figure 6: Comparison of Recall@20 varying the scale of the social parameters e, f and g.



(a) Twitter



(b) Tencent Weibo

Figure 7: Performance of recommendation for different user groups is studied, where x-axis represents the characteristics (number of celebrities a user follow) of different user groups.

the number of celebrities they followed. For each group, we sampled 4000 users from the whole social network, and we evaluate the performance in a warm-start setting. From Figure 7(a) and 7(b), we found that on Twitter, the user group that has followed 66-80 celebrities are more likely to accept the recommendations, while on Tencent Weibo it is the group that has followed 21-35 celebrities. The willingness to follow celebrities declines as users follow more celebrities. After users on Tencent Weibo have already followed more than 50 celebrities, their eagerness to follow new celebrities falls down quickly. However, users on Twitter tend to follow more celebrities. It shows that the behavior of American and Chinese users on social networks are quite different.

4.5 User Interests Exploration

We study how CSTR gives better interpretability and improves recommendation performance at the same time on Twitter dataset. Warm-start settings are used here. Top matched topics of one user can be found by ranking the factors of his latent feature vector u_i. Table 2 shows a representative user on Twitter. CTR believes this user likes news, technology and business. Then CTR tries to recommend the users some celebrities related to news and sociology, such as Barack Obama and CNN Breaking News. But this user do not follow any of them. On the other side, CSTR captures that this user likes film, music and reading. CSTR recommends celebrities this user really likes, Warren Ellis who is the writer of Gun Machine and David Wain who is a movie director. In a higher level view, CTR recommends this user 2 actors, a singer, a politician, 2 entrepreneurs and 2 medias, however CSTR focuses on recommending actors and singers to this user, where 2 writers, 4 actors, 1 singerwriter and 1 tv host are recommended. Obviously this user is prone to accept the recommendations from CSTR. From this case, we can conclude that user interest is regularized through celebrity social network. It shows that CSTR has advantages over CTR, that is CSTR can propagate topics through both celebrity and user interest matrix, which can capture users interest more precisely and gain better interpretability.

	CTR	CSTR
	Top 2 Preferred Topics	
1	bbc,radio,uk,tour,firefox, starbucks,mozilla,ufc,live france,harper,time,march	film,award,episode,star, movie,series,role,awards comedy,people,actor,life
2	iphone,airlines,apple, southwest,buffett, greenpeace,ebay,company	born,uk,american,trump, series,music,article,album people,tv,taylor,song,burke
	Top 8 Recommended Celebrities	
1	Wil Wheaton(✓)	Michael Ian Black(✓)
2	Barack Obama(×)	Wil Wheaton(✓)
3	Michael Ian Black(✓)	Veronica Belmont(×)
4	Kevin Rose(×)	Aziz Ansari(×)
5	MC Hammer(×)	Jonathan Coulton(×)
6	CNN Breaking News(×)	Orphan Annie(×)
7	The New York Times(×)	Warren Ellis(✓)
8	Michael Arrington(×)	David Wain(✓)

Table 2: An example user in Twitter from warm start situation. Two of the mostly preferred topics are listed. Each model recommended 8 celebrities to this user. The marker ✓ means follow action and × means no follow action.

5 Conclusions and Future Work

In this paper, we explore how to recommend celebrities to general users in the context of social network. We present a novel celebrity recommendation framework fusing matrix factorization for user following actions with matrix factorization for social network of celebrities, which are both affected by a unified topic model framework. The experiment results show that our approach outperforms the state-of-art algorithms for both warm start and cold start situations and captures users' interest more precisely. In the future, nonparametric model can be considered to jointly model the celebrity social network and user follow action records.

Acknowledgments

The work was supported by National Natural Science Foundation of China (60973103, 90924003).

References

[Agarwal and Chen, 2009] D. Agarwal and B.C. Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM, 2009.

[Agarwal and Chen, 2010] D. Agarwal and B.C. Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 91–100. ACM, 2010.

[Ahmad, 2011] A. Ahmad. Social network sites and its popularity. *International Journal of Research and Reviews in Computer Science*, 2(2), 2011.

[Bertsekas, 1999] Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[Blei *et al.*, 2003] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[Cheng *et al.*, 2012] C. Cheng, H. Yang, I. King, and M.R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. *AAAI, Toronto, Canada*, 2012.

[Dueck and Frey, 2004] D. Dueck and B. Frey. Probabilistic sparse matrix factorization. *University of Toronto technical report PSI-2004-23*, 2004.

[Friedkin, 2006] N.E. Friedkin. *A structural theory of social influence*, volume 13. Cambridge University Press Cambridge, 2006.

[Hofmann, 1999] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[Hu *et al.*, 2008] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, pages 263–272. IEEE, 2008.

[Koren *et al.*, 2009] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[Kwak *et al.*, 2010] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[Ma *et al.*, 2008] H. Ma, H. Yang, M.R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[Ma *et al.*, 2011] H. Ma, D. Zhou, C. Liu, M.R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.

[Manning *et al.*, 2008] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[Purushotham *et al.*, 2012] S. Purushotham, Y. Liu, and C.C.J. Kuo. Collaborative topic regression with social matrix factorization for recommendation systems. In *Proceedings of the 29th international conference on Machine learning*. ACM, 2012.

[Rendle, 2010] S. Rendle. Factorization machines. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, pages 995–1000. IEEE, 2010.

[Salakhutdinov and Mnih, 2008a] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[Salakhutdinov and Mnih, 2008b] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20:1257–1264, 2008.

[Shan and Banerjee, 2010] H. Shan and A. Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM)*, pages 1025–1030. IEEE, 2010.

[Shen and Jin, 2012] Y. Shen and R. Jin. Learning personal+social latent factor model for social recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1303–1311. ACM, 2012.

[Su and Khoshgoftaar, 2009] X. Su and T.M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.

[Wang and Blei, 2011] C. Wang and D.M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456. ACM, 2011.

[Yang *et al.*, 2011] S.H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.

[Zimmerman *et al.*, 2002] J. Zimmerman, L. Parameswaran, and K. Kurapati. Celebrity recommender. *Carnegie Mellon University Reaseach Showcase*, 2002.