2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

Collaborative Topic Regression based on the Social Network and Sequential Behaviors

Bin Pan
Department of Computer Science
University of Inner Mongolia
Hohhot, Inner Mongolia, China
panbin1010a@qq.com

Junyi Wang
Department of Computer Science
University of Inner Mongolia
Hohhot, Inner Mongolia, China
wjyi@imu.edu.cn

***Abstract*—Social Network has become a very useful platform for users to share information and make friends with each other. In our daily life, we would like to take friends' advice when we choose products through the Internet. The sequential behaviors also play an important role in making recommendation, we can make use of the sequential factors to mine the relation between users. In order to take advantage of different factors when predicting ratings and enhancing the recommendation accuracy, we propose a novel hierarchical Bayesian model called N-CTR which combines topic model with probabilistic matrix factorization. Our model incorporates not only topic model to mine the latent topic between items and their tags, but also matrix factorization which handles ratings，social network and sequential behaviors. We have conducted experiments on data set hetcrec-2011-Lastfm.Compare with other recommendation algorithms, our method can effectively enhance the recommendation accuracy.**

Keywords—topic model; probabilistic matrix factorization; social network; sequential behaviors; collaborative topic regression

I. INTRODUCTION

Nowadays, with the dramatic development of Internet, there is lots of information that we can get through the Internet. But it becomes difficult for us to find what we really concern because of the explosion of information. Although search engines such as Google and BIDU can improve the problem of information overload effectively, it just uses the keywords given by users to offer the required query results. In order to provide personalized service to different people, the emergence of the recommendation system (RSs) [1] can give a platform for users to find what they really need in a short time. The social network platforms such as Facebook, YouTube, Weibo and electronic commerce like Amazon, Netflix and Taobao prefer to use the recommendation system to improve the experience of users.

Collaborative filtering [2] is a one of the most popular recommendation algorithms [3]. Although collaborative filtering has been extensively studied, it has inherent problems such as cold start and data sparsity. In order to solve the problems of the CF, lots of models have been proposed to improve the recommendation performance. Apart from CF, the content-based recommendation [4] is the other algorithm which has always been used in recommendation. It can analyze the valuation information of users. And it is verified that the hybrid model which combines with CF and the content-based algorithm outperforms the CF [5]. Matrix Factorization [6] is the most popular CF-based model, but it suffers from the sparsity and imbalance of rating data, especially for the new and infrequence users. In order to improve the MF model, Salakhutdinov has proposed Probabilistic Matrix Factorization and Bayesian Probabilistic Matrix Factorization [7,8].

To alleviate the shortcomings of the CF-based model, some additional information has been incorporated into RSs, such as content information, social trust relationship. Latent Dirichlet Allocation [9] can handle the content information and gets the implicit topic distribution between articles and words. Liu [10] and Zhao [11] have used the LDA model to extract the interest of users. Besides the content information of items, the social network [12] is also a very important factor to influence the recommendation performance. This approach makes recommendation for a user based on the ratings of the users that have direct or indirect social relations with the given user. The method based on the social network can improve the cold-start problem. Recently, the model based approaches for recommendation with social networks have been investigated [13,14]. These algorithms exploit the MF model to learn the user and item latent feature vectors. The sequential behavior is another vital factor which is easily to be ignored in recommendation. Korean has proposed SVD++ [15] model which combines many factors such as time and social network. And many methods have used time decay function to calculate the ratings. The Ebbinghaus Forgetting Curve [16] has been mixed into the recommendation algorithm properly.

In order to take full advantage of different information to make a better recommendation, we propose a novel model called N-CTR based on the LDA and PMF, where LDA can mine the topic distribution of items content which is consisted of various tags, and PMF can learn the user latent feature vector by user-item ratings, social network and sequential behaviors.

The rest of this paper is organized as follows: we discuss related work in Section Ⅱ. We describe our proposed N-CTR model in Section Ⅲ. Our experiments are reported in Section Ⅳ. Finally, we conclude the contribution and present some directions for future work.

II. RELATED WORK

Collaborative Topic Regression is proposed by Chong Wang in 2011 [17]. It is a novel model which combines the topic model and matrix factorization together. CTR assumes that the topic distribution which is generated by topic model represents the item latent feature vector generated by PMF. The advantage of the model is to integrate both feedback information and item content information successfully, and it has achieved promising performance in recommendation. CTR considers the item latent feature matrix is generated by topic model and includes a variable ε_j which offsets the topic distribution θ_j . ε_j can capture the item preference of a particular user based on their ratings. The generative process of CTR model is as follows:

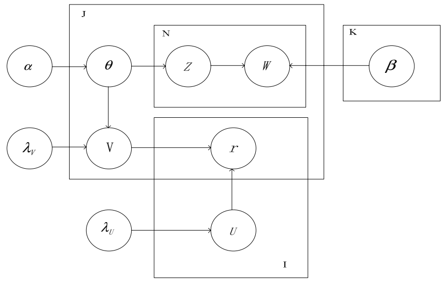

Figure 1. Collaborative Topic Regression Model

1.For each user i , draw user latent vector $u_i \sim N(0, \lambda_u^{-1} I_K)$

2.For each item j

(a) Draw topic proportions $\theta_j = Dirichlet(\alpha)$

(b) Draw item latent offset $\varepsilon_j = N(0, \lambda_v^{-1} I_K)$

And set the item latent vector as $v_j = \varepsilon_j + \theta_j$

(c)For each word w_{jn} ,

i. Draw topic assignment $z_{jn} \sim Mult(\theta_j)$

ii. Draw word $w_{jn} \sim Mult(\beta_{z_{jn}})$

3.For each user-item pair(i , j), draw the rating

$$r_{ij} \sim N(u_i^T v_j, c_{ij}^{-1})$$

CTR model performs well when it handles the data set which has both user-item ratings and item contents. But for a new or inactive user, this model does not learn the user latent vector well. Sanjay [18] proposed CTR-SMF model which utilizes Matrix Factorization based on social network to deal with the problem of cold start. Chen [19] added context information into CTR model, proposed an improved model called CTR-SMF2 in 2014. In our work, we would like to modify the CTR model by taking into account social network, sequential behaviors and item-tag information.

III. THE PROCESS OF BUILDING N-CTR MODEL

In this section, we will use the social network and sequential behaviors to modify PMF model and use the LDA model to mine the potential relation between items and tags. Firstly, we introduce the SAT-PMF model. Then we propose the T-LDA model. Finally, we propose the N-CTR model which combines the T-LDA model with SAT-PMF model. It takes good use of user-item ratings, social network, sequential behaviors and item-tag information to make recommendations.

A. The trust relationship between users

For the sake of saving time and making right decision, we like to take advice from others. Compare with the suggestion from strangers, we prefer to listen to the friends who we trust a lot. So in this paper, when considering the similarity in the social network, we often take the friendship into consideration. The decision of friends often influences our choice [20].

Previous work often uses 0 to represent there is no relation between users and uses 1 to represent that users trust each other completely. But in reality, the trust degree should be different between different users. We formalize (1) to calculate the trust.

$$T_{u,v} = \frac{friend(u) \cap friend(v)}{friend(u) \cup friend(v)} \tag{1}$$

For the relationship of friends, it can be described as a relational graph in Figure 2.

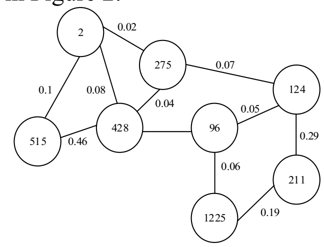

Figure 2. Trust Graph of friends

B. The influence of sequential behaviors

Traditional recommendation algorithms always ignore the time of users who evaluate items. We can use the sequential behaviors to mine the relationship between different users. If user A always evaluates same items before user B, we assume that there may be some connections between these two users [21].The user consume network is presented in Figure 3.

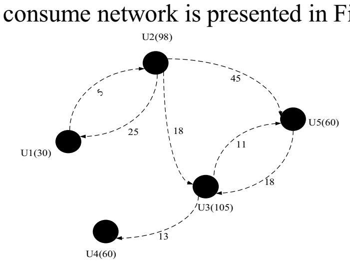

Figure 3. The graph of user evaluation based on sequential behaviors

In the user consume network $G = \{U, E\}$, U is the set of users and E is the set of lines. $W_{u \to v}$ represents the weight on each line. Number in the brackets represents the number of items which are evaluated by user. If we suppose that v and u evaluate the same item one after another in the same period of time (like one month), the $W_{v \to u}$ will increase 1.We find all items like this in the consume network, then we can calculate the influence:

$$S_{v \to u} = \frac{W_{v \to u}}{f(v,u)} \tag{2}$$

Where $f(v,u)$ is the set of items which are evaluated by v and u in the same time period. $S_{v \to u}$ is the effect of v on u based on the sequential behaviors.

C. The framework of SAT-PMF model

Considering social network and sequential behaviors, we introduce γ to mix these two important factors together like:

$$F_{v \to u} = \gamma S_{v \to u} + (1-\gamma) T_{v,u} \tag{3}$$

Where $F_{v\to u}$ is the influence of v on u . γ represents the weight between $S_{v\to u}$ and $T_{u,v}$. γ is always between 0 and 1.

We put the influence $F_{v\to u}$ into the PMF and take advantage of the trust propagation introduced by Mohsen [14]. Assuming that the user latent feature vector of user u depends on the latent feature vectors of all his neighbors $v\in N_u$. N_u is a set of users who appear not only in the trust graph of friends but also in the graph of user evaluation.

$$\hat{U}_u = \sum_{v\in N_u} F_{v\to u} U_v \tag{4}$$

For the user latent feature vector, we have two factors: the zero-mean Gaussian prior to avoid over-fitting, and the conditional distribution of the user latent feature vectors given the latent feature vecotrs of neighbors. Therefore,

$$\begin{aligned} p(U\,|\,F,\sigma_U^2,\sigma_F^2,) &\propto p(U\,|\,\sigma_u^2)\times p(U\,|\,F,\sigma_F^2) \\ &= \prod_{u=1}^{N} N(U_u\,|\,0,\sigma_U^2 I)\times\prod_{u=1}^{N} N(U_u\,|\,\sum_{v\in N_u} F_{v\to u}U_v,\sigma_F^2 I) \end{aligned} \tag{5}$$

The posterior distribution over user and item latent feature vectors is given by (6).

$$\begin{aligned} &p(U,V\,|\,R,F,\sigma_R^2,\sigma_U^2,\sigma_V^2,\sigma_F^2)\propto p(R\,|\,U,V,\sigma_R^2)p(U\,|\,F,\sigma_U^2,\sigma_F^2)p(V\,|\,\sigma_V^2) \\ &=\prod_{u=1}^{N}\prod_{i=1}^{M}[N(R\,|\,g(U_u^T V_j),\sigma_R^2]^{I_{u,i}^R}\times\prod_{u=1}^{N} N(U_u\,|\,\sum_{v\in N_u} F_{v\to u}U_v,\sigma_F^2 \mathrm{I}) \\ &\times\prod_{u=1}^{N} N(U_u\,|\,0,\sigma_U^2 \mathrm{I})\times\prod_{i=1}^{M} N(V_i\,|\,0,\sigma_V^2 \mathrm{I}) \end{aligned} \tag{6}$$

Our proposed model SAT-PMF which uses the social network and sequential behaviors is presented in Figure 4. It will affect the user latent feature vector in modified CTR model.

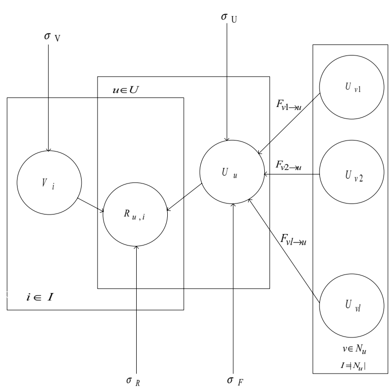

Figure 4. The SAT-PMF model based on User trust and Sequential behaviors

D. The topic model based on tag Information

LDA in CTR is often used to handle the items content information and generate the item latent feature vector. In this section, we propose T-LDA which can mine the potential relation between items and tags. The items, topics and tags in T-LDA correspond to the documents, topics and words in traditional LDA model respectively.

We take advantage of the characteristic of T-LDA to generate the topic distribution which affects the item latent feature vector in N-CTR model. The LDA combines with tag information of items is presented in Figure 5.

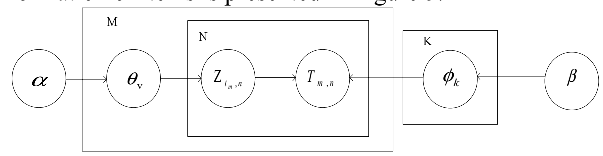

Figure 5. LDA Model combines with tag information of items

E. The framework of N-CTR

N-CTR is a hierarchical Bayesian model that jointly learns the user and item latent vector. It combines T-LDA with SAT-PMF together to modify the CTR model, showing in Figure 6. We use T-LDA to mine item-tag information and use the SAT-PMF model to handle the trust information, sequential factors and ratings.

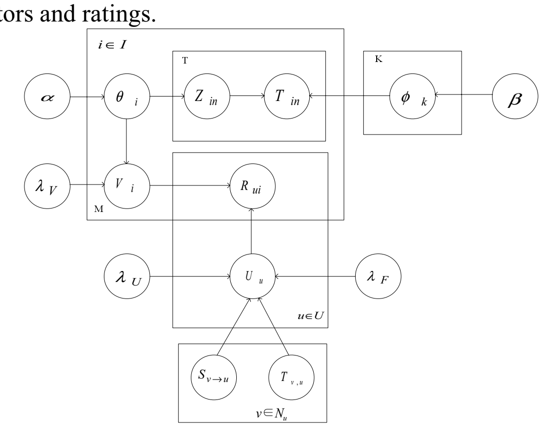

Figure 6. The graphical model for N-CTR model

The generative process of N-CTR model is as follows:

1.For each item i ,

(a)Draw topic proportions $\theta_i \sim Dirichlet(\alpha)$

(b)Draw item latent offset $\varepsilon_i = N(0,\lambda_v^{-1} I_K)$

(c)For each tag T_{in} ,

i. Draw topic assignment $Z_{in} \sim Mult(\theta_i)$

ii. Draw tag $T_{in} \sim Mult(\beta_{Z_{in}})$

(d)Set the item latent vector as $v_i = \varepsilon_i + \theta_i$

2.For each user u ,

(a)Draw user latent vector $u_u \sim N(0,\lambda_u^{-1} I_K)$

i. Add the influence between users $F_{v\to u}$

ii. The final user latent vector $u_u = \sum_{v\in N_u} F_{v\to u} u_v$

3.For each user-item pair (i , j), draw the rating

$$R_{ui} \sim N(u_u^T v_i, c_{ui}^{-1})$$

Just like the PMF model, the conditional distribution of observed ratings can be defined as:

$$p\left(R\middle|U,V,\sigma_R^2\right)=\prod_{i=1}^{M}\prod_{j=1}^{N}\left[N\left(R_{ij}\middle|g\left(U_i^T V_j\right),\sigma_R^2\right)\right]^{I_{ij}^R} \tag{7}$$

The user latent feature vector is as same as the vector in equation (5) because the N-CTR model is consisted of SAT-

PMF model. The item latent vector V_i is formed by a key property due to CTR, which can be defined as:

$$p(V \mid \sigma_V^2) \sim N(\theta_i, \lambda_V^{-1} I_K) \tag{8}$$

The following equation is the posterior of the latent feature vector given the user-item ratings, trust relation, sequential behaviors and tag information.

$$p(U,V \mid R,F,\sigma_R^2,\sigma_U^2,\sigma_V^2,\sigma_F^2) \propto p(R \mid U,V,\sigma_R^2) p(U \mid F,\sigma_U^2,\sigma_F^2) p(V \mid \sigma_V^2)$$
$$= \prod_{u=1}^{N}\prod_{i=1}^{M}[N(R_{ui} \mid g(U_u^T V_i),\sigma_R^2)]^{I_{ui}^R} \times \prod_{u=1}^{N} N(U_u \mid \prod_{v \in N_u} F_{v\to u} U_v, \sigma_F^2 I) \tag{9}$$
$$\times \prod_{u=1}^{N} N(U_u \mid 0,\sigma_U^2 I) \times \prod_{i=1}^{M} N(V_i \mid \theta_i,\sigma_V^2 I)$$

TABLE I. LIST OF NOTATIONS IN THE N-CTR MODEL

U	user set
I	item set
M	number of items
T	number of tags
K	number of topics or the dimension of latent vector
α	hyper-parameter of θ_i
β	hyper-parameter of ϕ_k
θ_i	K -dimensional topic distribution for item v_i
ϕ_k	tag distribution for topic k
z_{in}	topic for the n^{th} tag of item v_i
T_{in}	n^{th} tag of item v_i
U_u	latent feature for user u
V_i	latent feature for item i
R_{ui}	the final rating of item i by user u
$S_{u\to v}$	influence v on u based on the sequential behaviors
$T_{u,v}$	trust degree between u and v
λ_U	Gaussian offset parameter of user vector
λ_V	Gaussian offset parameter of item vector
λ_F	Gaussian offset parameter of influence network
N_u	the nearest neighbor of user u

F. Learning the parameter

It is hard for us to compute the posterior of U_u, V_i and θ_i directly. We use the EM algorithm to learn the maximum a posterior estimates. And maximization of the posterior is equivalent to maximizing the following log-likelihood of U, V, $\theta_{1\ldots I}$, F, and R given λ_U, λ_V λ_F, and β.

$$L = -\frac{\lambda_u}{2}\sum_u U_u^T U_u - \frac{\lambda_v}{2}\sum_i (V_i-\theta_i)^T (V_i-\theta_i)$$
$$+\sum_i\sum_n \log(\sum_K \theta_{ik}\beta_{k,T_{in}}) - \sum_{ui}\frac{c_{ui}}{2}(R_{ui}-U_u^T V_i)^2 \tag{10}$$
$$-\frac{\lambda_F}{2}\sum_u (U_u - \sum_{v\in N_u} F_{v\to u}U_v)^T (U_u - \sum_{v\in N_u} F_{v\to u}U_v)$$

Where $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, $\lambda_F = \sigma_R^2/\sigma_F^2$ and Dirichlet prior (α) is set to 1. We optimize this function by gradient ascent approach by optimizing the MF variables U_u, V_i and θ_i. For U_u and V_i, maximization follows in the similar fashion as MF. Given the current θ_i which is computed by T-LDA model at first, taking the gradient of L with respect to U_u, V_i and setting them to zero.

$$U_u \leftarrow (VC_uV^T + \lambda_u I_K + \lambda_F F_u I_K)^{-1}(VC_uR_u + \lambda_F U F_u^T) \tag{11}$$

$$V_i \leftarrow (UC_iU^T + \lambda_v I_K)^{-1}(UC_iR_i + \lambda_v\theta_i) \tag{12}$$

Where C_u and C_i are diagonal matrices. $F_u = F_{ui_{i=1}^I}$ is the influence matrix between users, and $R_u = R_{ui_{i=1}^I}$ for user u. R_i is similarly defined.

Given U and V, we can learn the topic proportions θ_i. We define $q(Z_{jn}=k)=\phi_{jnk}$, and we separate the items that contain θ_i and apply Jensen's inequality.

$$L(\theta_i) \geq \frac{-\lambda_v}{2}(v_i-\theta_i)^T(v_i-\theta_i) + \sum_n\sum_k \phi_{jnk}(\log\theta_{ik}\beta_{k,w_{in}} - \log\phi_{ink}) = L(\theta_i,\phi_i) \tag{13}$$

Where $\phi_j = (\phi_{jnk})_{n=1,k=1}^{N\times K}$, N is the tag number in the content of item i, and the optimal ϕ_{ink} satisfies $\phi_{ink} \propto \theta_{ik}\beta_{k,w_{in}}$. Coordinate decent can be applied to optimize U, V, $\theta_{1\ldots I}$ and $\phi_{1\ldots I}$. Then we can use U, V and ϕ to optimize β.

After the optimal parameters U^*, V^*, $\theta_{1\ldots I^*}$ and β^* are learned, N-CTR can make predictions.

$$R_{ui}^* \sim N((\sum_{v\in N_u}(F_{v\to u}U_v)^T(\theta_i+\varepsilon_i),\ c_{ui}^{-1}) = (U_u^*)^T V_i^* \tag{14}$$

IV. EXPERIMENTS AND ANALYSIS

In this section, we conduct several experiments to compare the performance of the N-CTR model with other state-of-the-art recommendation algorithms. We would like to know how our model performs compared with others and how different factors contribute to recommendation performance.

A. Dataset

We use hetrec2011-lastfm-2k (Lastfm) to conduct the experiments. The description of the dataset is showed in Table Ⅱ. We can make use of the sequential factors in the dataset of user-tags-items. The sparsity of the Lastfm is 99.7%

TABLE II. LIST OF NOTATIONS IN THE N-CTR MODEL

Dataset	Lastfm
users	1892
items	17632
tags	11946
user-user relations	25434
user-tags-items	186479
user-items-relations	92834

B. Evaluation and Comparisons

For evaluation, we choose MAE (Mean Absolute Error) and RMSE (Root Mean Square Error) metric to measure the performance of recommendation models. They are defined as below:

$$MAE = \frac{1}{N}\sum_{j=1}^{N}\left|r_{ij}-\hat{r}_{ij}\right| \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum\left(r_{ij}-\hat{r}_{ij}\right)^2} \tag{16}$$

Where N is the number of prediction, r_{ij} is the real rating of an item and $\hat{r}_{ij}$ is the predicted rating.

Apart from the evaluation, we should compare our model with the appropriate models, we list them in Table Ⅲ.

TABLE III. RECOMMENDATION MODEL COMPARED WITH N-CTR MODEL

The compered model	Information in the model
PMF	rating
SocialMF	rating+trust
CTR	rating+tag
CTR-SMF	rating+tag+trust
N-CTR	rating+tag+trust+sequential factor

C. Experimental settings

We split the dataset into two parts, a training dataset (90%) which is used to train the model and a testing dataset (10%) that can verify the performance of the experiments. After a great number of experiments, we find that $K = 50$, $\lambda_U = 0.1$, $\lambda_V = 100$, $a = 1$, $b = 0.01$ [12] gives good performance for CTR model. For N-CTR model and other models we introduce above, we choose the same parameters like CTR.

D. Impact of λ_V

The parameter λ_V balances the item latent feature vector based on the item-tag information. If $\lambda_V = 0$, N-CTR model only considers the ratings and relation between users, so the model is similar to the SAT-PMF. When $\lambda_V = \infty$, N-CTR is as same as CTR. Figure 7 shows the effect of λ_v in our model.

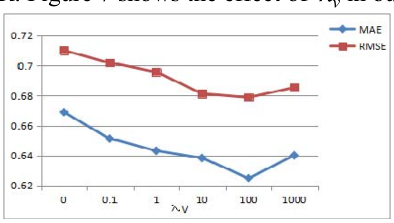

Figure 7. Comparison of predictive performance for N-CTR by varying MAE for N-CTR by varying λ_v

As we can see from the Figure 7, the accuracy changes when we set different λ_V. It shows that the item-tag information contributes to the process of recommendation. The N-CTR model achieves the best predictive in MAE and RMSE when $\lambda_V = 100$. But when the λ_V grows to a certain extent, the recommendation accuracy of N-CTR decreases, indicating that if the model considers too much impact of the item content information, it will lead to the bias. In the process of parameter optimization, the predictive performance will not be better when the value of λ_V is bigger.

Figure 8 clearly shows our model outperforms CTR model by 3.5%. It can be explained that since our model makes use of both social network and sequential behaviors, it can handle the user latent feature matrix better. It is very useful to take into account different factors when we modify the model.

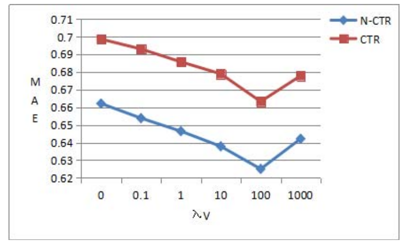

Figure 8. Comparison of MAE for CTR and our proposed model N-CTR by varying λ_v

E. Impact of λ_F and γ

The parameter λ_F balances the user latent vector based on the social network and sequential behaviors. We analyze the performances when we set λ_F which equals to 0, 0.1, 1, 10, 100, 1000 separately. As we can see from Table Ⅳ, when $\lambda_V = 100$, the MAE in N-CTR is changed by the different value of λ_F. When $\lambda_F = 100$, the model achieves the best predictive.

TABLE IV. MAE FOR N-CTR BY VARYING λ_F

λ_v	λ_F					
	0	0.1	1	10	100	1000
100	0.6632	0.6579	0.6484	0.6351	0.625	0.6452

Figure 9 shows the performance of N-CTR model when we change the value of λ_F and λ_v. Although we set different value of λ_v, the MAE performs best when $\lambda_F = 100$. And when $\lambda_F = 0$, the curve reaches the highest point. It indicates the trust relation and sequential behaviors make contributions to the model.

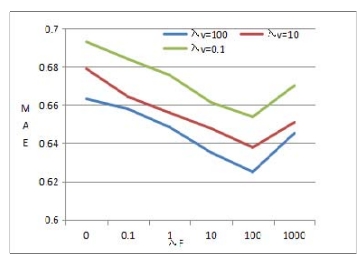

Figure 9. Comparison of MAE for N-CTR by varying λ_v and λ_F

γ represents the weight between social trust and sequential behaviors. When $\gamma = 0$, the N-CTR model just takes trust relation into consideration and is similar to the SocialMF [14] model with improved calculation of trust relation. When $\gamma = 1$, the model ignores the influence of friends and only uses the sequential behaviors to compute the relation between users. We can conclude from the Table Ⅴ. When $\gamma = 0.4$, the N-CTR model gets the best performance. And trust relation plays a more important role in N-CTR model because the sequential matrix in dataset Lastfm is sparser than the matrix of trust between friends. There are only several users who always evaluate the same items in the same time period. But mixing the two factors into the model can get a better predictive result.

TABLE V. MAE FOR N-CTR BY VARYING γ

λ_F	γ					
	0	0.2	0.4	0.6	0.8	1
0.1	0.6617	0.6584	0.6538	0.6557	0.6602	0.6678
1	0.6599	0.6496	0.6454	0.6478	0.6578	0.6643
10	0.6584	0.6402	0.6378	0.6397	0.6530	0.6615
100	0.6469	0.6354	0.6250	0.6329	0.6425	0.6530
1000	0.6593	0.6453	0.6421	0.6442	0.6543	0.6635

F. *Performance analysis*

We compare the performance of various recommendation models on Lastfm dataset. As we can see from Table , our modified model N-CTR improves the accuracy rate of PMF, SocialMF, CTR, and CTR-SMF by as high as 7.36%/7.94%, 4.68%/4.12%, 3.82%/3.28%, and 2.19%/1.1% in terms of MAE/RMSE respectively when the training data is 90%.

TABLE VI. RECOMMENDATION ACCURACY PERFORMANCE COMPARISON

Training Data	**Metrics**	**K=50**				
		PMF	**Social-MF**	**CTR**	**CTR-SMF**	**N-CTR**
90%	MAE	0.6986	0.6718	0.6632	0.6469	0.6250
	RMSE	0.7555	0.7173	0.7089	0.6871	0.6791
80%	MAE	0.7113	0.6847	0.6745	0.6508	0.6383
	RMSE	0.7684	0.7249	0.7156	0.7082	0.6896

N-CTR model performs better than other models because it contains many different factors to modify the user and item latent vectors and takes advantages of CTR model. N-CTR can handle item-tag information, trust relation between friends, influence between users obtained from sequential behaviors and user-item ratings.

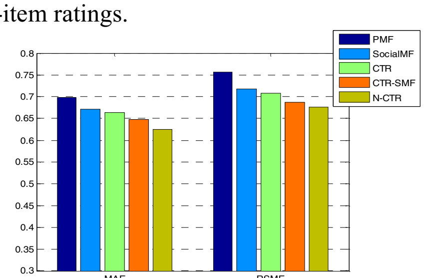

Figure 10. Prediction performance of N-CTR compared with other models

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel model called N-CTR which combines ratings, item-tag information, social network, sequential factors to improve the recommendation performance. And we apply trust relation and sequential behaviors to the PMF and use them to modify the user latent vector. We also mine the implicit relation between items and tags, using T-LDA model to get the topic distributions which can modify the item latent vector.

In the future work, we will find a new way which considers not only direct friends in the social network, but also the users who have a little connection with the specific user to extent the social network. Another direction is to conduct more experiments on different datasets such as hetrec2011-delicious - 2k and Epinions.

ACKNOWLEDGMENT

This research is financially supported by Inner Mongolia University.

REFERENCES

[1] Adomavicius G,Tuzhilin A. "Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transaction on Knowledge and Data Engineering, vol.1, no.17 ,pp.734-749, 2005.

[2] Resnick P, Varian HR. "Recommender systems," Communications of the ACM, vol.40, no.3, pp.56-58, 1997.

[3] Ji H, Li J F, Ren C. "Hybrid collaborative filtering model for improved recommendation," IEEE International Conference on Service Operations and Logistics, and Informatics, pp.142-145, 2013.

[4] Pazzani M J,Billsus D. "Content-based recommendation system," The adaptive web. Springer Berlin Heidelberg, pp.325:341, 2007.

[5] Tuzhilin A. "Towards the Next Generation of Recommender Systems," International Conference on E-Business Intelligence, pp.9-19, 2010.

[6] Sarwar B, Karypis G, Konstan J. "Application of Dimensionality Reduction in Recommender Systems," In Acm Webkdd Workshop, 2000.

[7] Mnih,Andriy,Ruslan Salakhutdinov. "Probabilistic matrix factorization," Advances in neural information processing systems, 2007.

[8] Salakhutdinov, Ruslan,A. Mnih. "Bayesian probabilistic matrix factorization using markov chain monte carlo," International Conference on Machine Learning, pp.880-887, 2008.

[9] Blei D M, Ng A Y, Jordan M I. "Latent dirichlet allocation," Journal of Machine Learning Research, vol.3 , pp.993-1022, 2003.

[10] Qi Liu. A study of Designing and Applying Recommendations Based on User Interests Modeling, University of Science and Technology of China, 2013.

[11] MeiXue Zhao. Research on Personalized Recommendation Algorithm Based on User Model, Nanjing University of Science and Technology, 2015.

[12] Golbeck J A, Hendler J. "Computing and applying trust in web-based social networks," 2005.

[13] Ma H, Yang H, Lyu M R. "SoRec : social recommendation using probabilistic matrix factorization," ACM Conference on Information and Knowledge Management, CIKM ,vol.9, pp.931-940, 2008.

[14] Jamali M, Ester M. "A matrix factorization technique with trust propagation for recommendation in social networks," ACM Conference on Recommender Systems, Recsys, pp.135-142, 2010.

[15] Koren Y. "Collaborative filtering with temporal dynamics," Communications of the ACM, vol.53, no.4, pp.89-97, 2010.

[16] Li Z, Lin J. "An Analysis of the Subscription in User-Generated Content Video Systems," International Conference on Computer Communications and Networks, pp.1-7, 2012.

[17] Wang C, Blei D M. "Collaborative topic modeling for recommending scientific articles," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.448-456, 2011.

[18] Purushotham S, Liu Y, Kuo C C J. "Collaborative Topic Regression with Social Matrix Factorization for Recommendation Systems," Computer Science, 2012.

[19] Chen C, Zheng X, Wang Y. "Context-ware Collaborative Topic Regression with Social Matrix Factorization for Recommender Systems," Twenty-Eighth AAAI Conference on Artificial Intelligence, pp.9-15, 2014.

[20] Hong C. "Capturing Semantic Correlation for Item Recommendation in Tagging Systems," Thirtieth AAAI Conference on Artificial Intelligence, pp.160-166, 2016.

[21] Sun Guangfu, Wu Yue, Liu Qi, "Collaborative filtering recommendation algorithm based on time sequence behavior, Journal of software, vol.11, pp. 2721-2733, 2013.