A Neural Network Model for Social-Aware Recommendation

Lin Xiao[1(✉)], Zhang Min[2], Liu Yiqun[2], and Ma Shaoping[2]

[1] Institute of Interdisciplinary Information Sciences,
Tsinghua University, Beijing, China
jackielinxiao@gmail.com
[2] Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University,
Beijing 100084, China
{z-m,yiqunliu,msp}@tsinghua.edu.cn

Abstract. Social-aware recommender systems have been popular with the rapid growth of social media applications. Existing approaches have attempted to accommodate social information into typical Collaborative Filtering methods and achieved significant improvements. Neural networks are gaining increasing interests in information retrieval tasks. However few studies have considered applying neural network in social-aware recommendation tasks. In this paper, we aim to fill this gap and propose a social-aware neural recommender system. Extensive experiments on real-world datasets demonstrate that our model outperforms state-of-art approaches significantly.

1 Introduction

Recommender systems aim to provide information or items of interest to users. The various user profiles have been of great importance in enhancing the performance of recommendation. The rapid growth of social media applications reflect the social connections between users, which contributes to a better understanding of user preferences. Existing studies have attempted to utilize social information in recommender systems and achieved significant improvement.

The classical social network theory builds upon two important assumptions: first, users who are socially connected are believed to be more similar than those who are not. This is also referred to as "Homophily Effect". Second, users are connected with different social ties. These social theories have been exploited in recommender systems in previous studies.

Meanwhile, the neural networks are widely applied in various fields, including computer vision, natural language processing and information retrieval. The strong expressive power of neural networks allows for extracting useful features from the input and further adapts them into the corresponding tasks. The studies on applying neural networks in recommender systems can be found in [15]. The typical approaches in previous studies are built upon Restricted Bolzman Machine and its variants. Recent studies attempt to mimic collaborative filtering

W.-K. Sung et al. (Eds.): AIRS 2017, LNCS 10648, pp. 125–137, 2017.
https://doi.org/10.1007/978-3-319-70145-5_10

and matrix factorization with neural networks. Each user and item are embedded as corresponding vectors and factorization is conducted on these vectors following classical approaches.

There exist some studies on network embedding [19,28], including social networks and information networks. Similar to word embedding [9] in NLP, the network embedding aims to represent each node as a vector in the Euclidean space and the nodes connected with edges are closer to each other. The network embedding problem is close to recommendation when each user and item are seen as nodes and the interactions between them are seen as edges. However, it is not trivial to apply network embedding techniques on recommendation, especially in social-aware recommendation tasks. The interactions between users and items are different from those between users, therefore the network embedding procedure can not be directly adapted into two heterogeneous networks directly. Meanwhile, social information is referred to as the contents of the node while network embedding is to discover the structures of the network. Therefore social-aware recommendation tasks require a modeling of the network structure with node contents.

Despite existing successes in applying neural networks on recommender systems and network embedding, no previous studies considered the specific problem of incorporating social information into the recommender systems with neural networks. There are two important challenges in designing social-aware neural recommender systems: first, it is unknown how to encode social information into the neural network framework; second, the impact of social connections needs to be modeled properly in the recommendation process. Although existing studies attempt to utilize social network theories in recommendation, the theories have not been well accommodated into neural network frameworks.

In this work, we aim to extend the classical SVD++ model with both social information and the powerful neural network framework. The classic SVD++ model is proposed in [6] and gets widely used in recommendation for its superior performances in Netflix Challenge and real-life recommender systems. The basic idea of SVD++ is to model both explicit and implicit feedbacks with matrix factorization. We treat social information as the implicit feedbacks in SVD++ and further model its interaction with users and items with factorization and neural network scheme. Extensive experiments have been conducted on two real-world datasets, the results validate the soundness of coordinating social information with neural networks.

The remainder of this paper is organized as follows: Sect. 2 introduces some important aspects of related studies; the models of SVD++ and Neural SVD++ are introduced in Sect. 3; the experimental results are presented in Sects. 4 and 5 presents the conclusion of this paper and provides a discussion about the future work.

2 Related Work

In this section, we make a brief review of related studies, including the typical Collaborative Filtering approach, the social-aware recommender systems and existing studies on recommendation with neural networks.

2.1 Collaborative Filtering for Recommendation

Collaborative Filtering (CF) is a typical approach for recommendation [16]. The motivation comes from the assumption that people often get the best recommendations from someone with tastes similar to themselves. There are two generic Collaborative Filtering approaches, i.e. user based CF and item based CF. The user-based CF adopts the motivation stated before while the item-based CF assumes that items tend to receive similar ratings with other similar items.

Existing CF approaches include two categories: memory-based and model-based approaches. User-KNN and Item-KNN are two representative algorithms in memory-based algorithms. Locality-sensitive hashing [17] is a typical algorithm adopted to find similar users, which implements the KNN algorithm in linear time.

Model-based approaches use machine learning techniques to model the generation process of ratings. Among various model-based CF methods, Matrix Factorization (MF) is the most popular and effective one, which assumes that users and items are represented as vectors in a latent factor space. Some MF based approaches, including SVD++ [6], NMF (Non-Negative Matrix Factorization) [26], MMMF (Max-Marginal Matrix Factorization) [13], BMF (Biased Matrix Factorization) [6] and PMF (Probabilistic Matrix Factorization) [14] have achieved superior accuracy and scalability in recommendation due to the dimension reduction nature.

2.2 Social-Aware Recommendation

Social information is known to be helpful in recommendation systems [3,4,10, 11,20,25]. Most studies assume that some social homophily effect exists, causing users to behave consistently with others in social connections [3,10].

[5] extends the approach in [4] by combining random walks with collaborative filtering for item recommendation. The Multi-Relational Bayesian Personalized Ranking (MR-BPR) model [7], which combines multi-relational matrix factorization with the BPR framework, predicts both user feedback on items and on social relationships.

There are two state-of-the-art algorithms that utilize social information for item recommendation tasks [27]. In [27], the authors assume that users are more likely to have seen items consumed by their friends, and use this effect to sample negative feedback in BPR. In contrast, [1] utilizes Poisson factorization to incorporate social information into a matrix factorization scheme. None of the aforementioned studies considers the modeling of missing feedback in social relationships. In the present study, we use some of the most popular social-aware recommendation algorithms as benchmarks.

A recent study [21] considers the strength of social ties and its application in social recommendations. The neighborhood overlap is used to approximate tie strength and extend the popular BPR model to incorporate the distinction between strong and weak ties. As this is an extension of BPR, the missing feedback is randomly selected as negative feedback. In [12], the Collaborative

Ranking (CofiRank) model [22] is extended to include social connections. As CofiRank considers observations of both positive and negative feedback, the issue of missing feedback is not investigated. Two different CofiRank strategies are proposed based on the notions of Social Reverse Height and Social Height, which quantify how well the relevant and irrelevant items of users and their social friends have been ranked, respectively.

2.3 Neural Network for Recommendation

Another trend on related research is to utilize deep learning method in recommender system. With the growing research in artificial neural networks and deep learning techniques, some of the famous Deep Learning models are applied in recommender systems. The first model [15] applied in recommender system is the Restricted Botzman Machine which assumes each user is depicted with a RBM where the ratings are modeled as binary input vectors and the hidden units and correlation weights are used to generate the full ratings. Another dual-reversible RBM which takes linear inputs is proposed in [2] and the model uses a single RBM instead of user-size RBMs to generate the full ratings. Despite that the RBM technique can achieve performances that are comparable to popular Matrix Factorization techniques, the training process of RBM is quite intractable. The mean-field technique is used to relax the model into a two-layer feed-forward neural network and thus the model is quite easy to train. The RBM based model is referred as Neural Auroregressive Distribution Estimator (NADE) [29]. The model is applied in modeling the distribution of high-dimensional vectors in [8] and achieves better performance than the original RBM model. Recently, a new model is proposed in [18] which is a novel autoencoder framework for collaborative filtering (CF) since auto-encoder has a good performance on dimension reduction. However existing ANN (Artificial Neural Network) based recommender systems have not started to combine side information with recommender systems and a proper model is yet to be designed in the future.

3 Model

In this section, we specifically introduce the Neural Social SVD++ (Neural SSVD++) model. Before we present the details of the model, some preliminary knowledge about SVD++ is first introduced. Then we introduce the Neural SVD++ model and how we extend the idea of SVD++ with the Neural Network framework.

3.1 SVD++

SVD++ is first proposed in [6]. In this model, each user and item and implicit feedback are represented as latent factor vectors and the rating is assumed to be a combination of user and item latent factor vectors with implicit feedbacks. The implicit feedbacks refer to the interaction records of users, i.e. the clicks of advertisements by users, the browsing history of users on the websites. In SVD++, the influence of these implicit feedbacks are considered and modeled as latent factor vectors in the matrix factorization framework. The parameters of this model is listed in Table 1.

$$\hat{r}_{ui} = b_{ui} + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j) \tag{1}$$

where $b_{ui} = b_u + b_i + \mu$. The parameters of this model can be determined by minimizing the empirical squared loss functions of observed ratings, which can be achieved with Stochastic Gradient Descent algorithm.

$$\min . \sum_{u,i \in \mathcal{O}} |\hat{r}_{ui} - r_{ui}|^2 + \lambda(\sum_i \|q_i\|_2 + \sum_u \|p_u\|_2 + \sum_j \|y_j\|_2) \tag{2}$$

Table 1. Variables of SVD++

Variables	Meaning
r_{ui}	The rating given by user u to item i
$\hat{r}_{ui}$	The prediction of rating r_{ui}
$\mathcal{O}$	The set of observed ratings
b_u	The bias of user u
b_i	The bias of item i
μ	The global bias
q_i	The latent factor vector of item i
y_j	The latent factor vector of implicit feedback j
p_u	The latent factor vector of user u
$N(u)$	The set of implicit feedbacks of user u

3.2 TrustSVD

TrustSVD model [3] is also built on top of the SVD++ model, which also takes into consideration user/item biases and the influence of rated items other than user/item-specific vectors on rating prediction. Formally, the rating for user u on item i is predicted by:

$$\hat{r}_{ui} = b_u + b_i + \mu + q_i^T (p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} w_j + |T_u|^{-\frac{1}{2}} \sum_{v \in T_u} y_u) \tag{3}$$

where $\{y_v, \forall v \in T_u\}$ refer to user-specific latent factor vector of users $v \in T_u$ who are socially connected to user u, and w_j denotes the influence of items $j \in I_u$ rated by user u in the past.

This model is a direct extension of SVD++ with trust relationships. The parameters can be learnt via SGD in a similar manner.

3.3 Neural SSVD++

We extend the idea of SVD++ with Neural Network and propose the model Neural Social SVD++ (Neural SSVD++). The model is shown in Fig. 1 (Those arrows in dash lines refer to the element-wise product of vectors). Following a similar spirit of SVD++, Neural SSVD++ embeds users, items and socially connected users (seen as implicit feedbacks in SVD++) as latent factor vectors. The impact of social connections is incorporated into the model by considering both its representation in the latent factor space and its interaction with users. Moreover, we utilize the nonlinearity of neural networks to model the relationship between ratings and the latent factor vectors and their interactions.

In this model, the input is a row vector containing three components: the first component is the target user ID u and the second component is the target item ID i, the third component is a sequence of user ID $N(u)$ where each user is socially connected to the target user. The input then goes through embedding layer and each target user and item are mapped into corresponding vectors, the sequence of socially connected users are mapped into a sequence of vectors $y_j, \forall j \in N(u)$. Then the sequence of social vectors is reduced to a single vector $|N(u)|^{-1} \sum_{j \in N(u)} y_j$. The interactions between these latent factor vectors are modeled in a pairwise manner: $p_u \otimes q_i$ and $|N(u)|^{-1} \sum_{j \in N(u)} y_j \otimes q_i$ (where $\otimes$ is an element-wise product operator). Then we concatenate these factor vectors together: $H_{ui} = [p_u; q_i; |N(u)|^{-1} \sum_{j \in N(u)} y_j; p_u \otimes q_i; |N(u)|^{-1} \sum_{j \in N(u)} y_j \otimes q_i]$. After the concatenation, we use a two-layer fully connected layer to model the

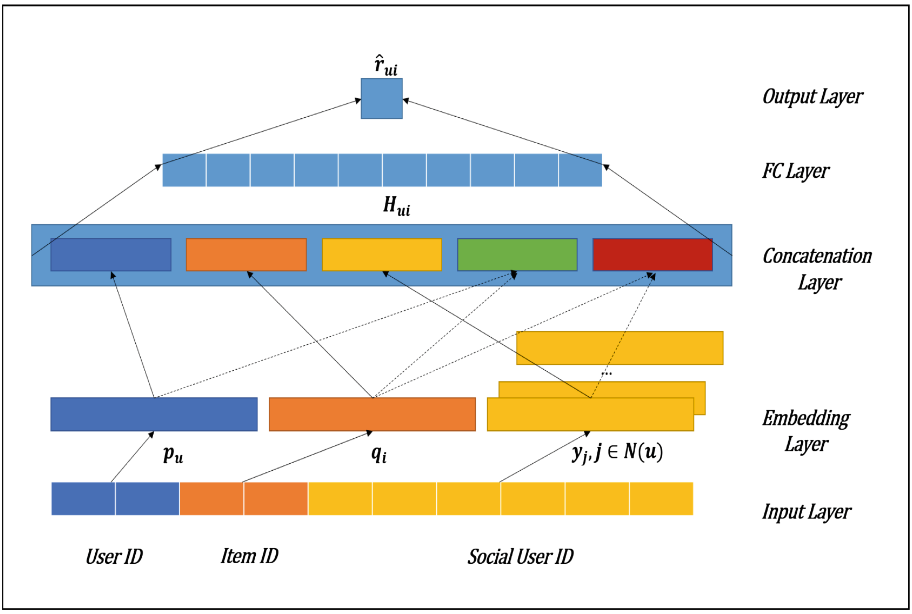

Fig. 1. Neural Social SVD++, all the activation functions are Relu.

relationship between the concatenated vector and the output rating with non-linearity:

$$y_{ui} = Relu(W_2 \cdot Relu(W_1 \cdot H_{ui} + b_1) + b_2) \tag{4}$$

Since H_{ui} is a concatenation of multiple latent factor vectors, the function can be re-written into following form:

$$\begin{aligned} y_{ui} = & Relu(W_2 \cdot Relu(W_{11}p_u + W_{12}q_i + W_{13}|N(u)|^{-1} \sum_{j \in N(u)} y_j + \\ & W_{14}p_u \otimes q_i + W_{15}|N(u)|^{-1} \sum_{j \in N(u)} y_j \otimes q_i + b_1) + b_2) \end{aligned} \tag{5}$$

For the two Fully Connected layers, we first reduce the dimension of hidden units to 16 and then reduce it to the single valued rating $\hat{r}_{ui}$.

3.4 Learning

We adopt the typical back-propagation method for learning the parameters of the model. For the rating prediction tasks, we use the empirical squared loss function as the objective and use Adagrad algorithm for optimization:

$$\mathbb{L} = \sum_{u,i \in \mathcal{O}} |\hat{r}_{ui} - r_{ui}|^2 + \lambda \Omega(W, p, q, y) \tag{6}$$

where the first component is the squared loss of rating prediction and the second component is the regularization term.

4 Experiment

In this section, we present the experimental results on two real-world datasets and the performances in different circumstances. Moreover, we compare the performances of Neural SSVD++ with and without social information as input. The results show that our model outperforms other baselines and social information contributes to the prediction accuracy significantly.

4.1 Experiment Setting

Two representative datasets are selected for experiments: Ciao and Epinions. These two datasets contain both rating records and in-site social relationships The statistics of these two datasets are presented in Table 2.

We adopt five-fold cross-validation, the datasets are split into five folds and four of them are selected as training set while the remaining fold is used for testing. The experiment is conducted in an environment with a 1.8 GHZ CPU. We use the mini-batch training scheme when training the model and the batch size is set to 128.

We select some state-of-the-art approaches as baseline algorithms, including SoRec, Soreg, TrustSVD, TrustMF and SVD++:

Table 2. Statistical details of the datasets

Datasets	#Users	#Items	#Ratings	#Links	Link type
Ciao	7,267	11,211	147,995	111,781	Unilateral
Epinions	38,089	23,585	488,917	433,416	Unilateral

- **SoRec**[10]**:** Sorec co-factorizes the rating matrix and social matrix simultaneously and both matrices share the same user factor vectors.
- **SoReg**[11]**:** The model adds social regularization into the matrix factorization framework based on the social homophily effect.
- **SVD++**[6]**:** SVD++ is a model that merges latent factor model and neighbourhood effect together. Furthermore, it can be extended to incorporate both implicit and explicit feedbacks from users.
- **TrustMF**[24]**:** TrustMF assigns each user a trustor-specific vector and a trustee-specific vector. The model can choose to incorporate either vector or both vectors in the matrix factorization framework.
- **TrustSVD**[3]**:** TrustSVD extend SVD++ with social trust information and takes into account both the explicit and implicit influence of ratings and trust information when predicting ratings of unknown item.

In order to evaluate the performances on rating prediction tasks, we use typical MAE and RMSE metrics:

$$
\begin{aligned}
MAE &= \frac{1}{|T|} \sum_{(u,i)\in T} |\hat{r}_{ui} - r_{ui}| \\
RMSE &= \sqrt{\frac{1}{|T|} \sum_{(u,i)\in T} |\hat{r}_{ui} - r_{ui}|^2}
\end{aligned} \tag{7}
$$

where T denotes the testing set, $\hat{R}_{ij}$ denotes the prediction of the ground truth R_{ij}.

4.2 Experimental Results

We list the comparison between Neural SSVD++ and other baselines under two circumstances in Tables 3 and 4. First, we compare the performances of Neural SSVD++ with other baselines on two datasets. The results indicate that Neural SSVD++ outperforms other baselines in terms of rating prediction accuracy. Generally, the social-aware recommendation algorithms perform better than those social-unaware algorithms (comparing SVD++ with others), which indicates that social connections are informative for recommendation. More specifically, Neural SSVD++ outperforms other social aware algorithms, which means that accommodating social information with neural networks significantly improves the expressive and predictive power of the model.

Table 3. Performance of Rating Prediction on Ciao and Epinions, *: $p < 0.01$

Ciao	SoRec	SoReg	TrustSVD	TrustMF	SVD++	Neural SSVD++
MAE	0.761	0.815	0.723	0.742	0.748	**0.690**$*$
MAE Improvement	9.33%	15.34%	4.56%	7.01%	7.75%	-
RMSE	1.010	1.076	0.956	0.983	1.001	**0.942**$*$
RMSE Improvement	6.73%	12.45%	1.46%	4.17%	5.89%	-
Epinions	SoRec	SoReg	TrustSVD	TrustMF	SVD++	Neural SSVD++
MAE	0.882	0.932	0.805	0.818	0.818	**0.790**$*$
MAE Improvement	10.43%	15.23%	1.86%	3.42%	3.42%	-
RMSE	1.114	1.232	1.044	1.095	1.057	**1.025**$*$
RMSE Improvement	7.99%	16.80%	1.82%	6.39%	3.02%	-

Table 4. Performance on cold-start users in Ciao and Epinions, *: $p < 0.01$

Ciao	SoRec	SoReg	TrustSVD	TrustMF	SVD++	Neural SSVD++
MAE	0.730	0.949	0.721	0.752	0.749	**0.704**$*$
MAE Improvement	3.56%	25.81%	2.36%	6.38%	6.00%	-
RMSE	0.998	1.214	0.962	1.096	1.020	**0.950**$*$
RMSE Improvement	5.05%	21.75%	1.25%	13.32%	6.86%	-
Epinions	SoRec	SoReg	TrustSVD	TrustMF	SVD++	Neural SSVD++
MAE	0.846	1.139	0.868	0.853	0.889	**0.836**$*$
MAE Improvement	1.18%	26.60%	3.69%	1.99%	5.96%	-
RMSE	1.138	1.437	1.105	1.125	1.162	**1.076**$*$
RMSE Improvement	5.45%	25.12%	2.62%	4.36%	7.40%	-

Moreover, we evaluate the performances of Neural SSVD++ and other baselines w.r.t cold-start users. We adopt the setting where users rate fewer than five items are referred as cold-start users, which is typical in related studies like [3,23]. The results show that Neural SSVD++ still performs better than others. Since the utilization of social information compensates for the shortage of ratings from cold users, neural networks have an advantage of expressive power and work more effectively for cold-start users.

The Impact of Network Depth. Typically, the depth of neural network is related to the performance. In tasks like image classification and text classification, neural networks are designed to be deep for a better performance. For this task, we conduct experiments to explore the relationship between network depth and the rating prediction accuracy. Surprisingly, we discover that deeper network does not lead to superior performances necessarily. We alter the depth of Fully Connected layers from one to five and observe the resulting RMSE in different cases.

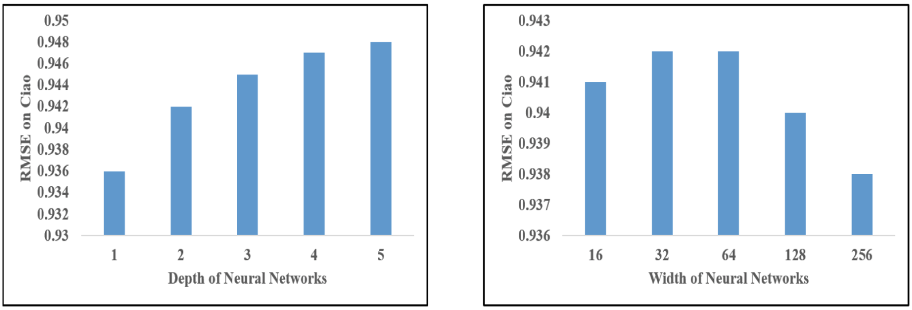

Fig. 2. RMSE of Neural Social SVD++ with different Depths and Widths

The RMSE of Neural SSVD++ under different network depths on Ciao are presented in Fig. 2. We discover that the performances of Neural SSVD++ does not necessarily get improved with deeper network. Since the depth of the network represents the model complexity, higher depth may cause overfitting in some cases.

The Impact of Embedding Dimensions. The dimension of embedded vectors decides the width of the neural network. We also evaluate the impact of embedding dimensions in Neural SSVD++. We alter the embedding dimension from 16 to 256 with a power of two for each trial. The MAE in different cases are shown in Fig. 3. Judging from the results, a proper setting for rating prediction tasks is a dimension of 64 or 128.

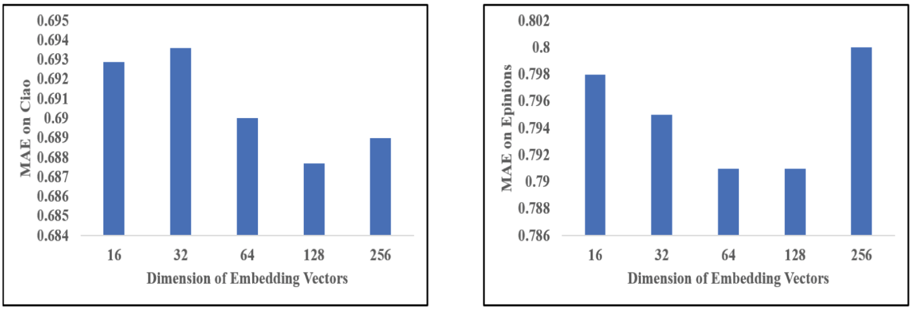

Fig. 3. MAE of Neural Social SVD++ with different Embedding Dimensions

Similarly, the performance does not keep going better with the increase of embedding dimensions. A greater dimension of embedding vectors allows for stronger expressive power, however also leads to higher risk of overfitting.

The Impact of Social Coordination. In order to evaluate the enhancement brought by social information, we also conduct experiments on Ciao dataset with Neural SSVD but removing the social input from the model. Therefore the predicted rating becomes:

$$y_{ui} = Relu(W_2 \cdot Relu(W_{11}p_u + W_{12}q_i + W_{14}p_u \otimes q_i + b_1) + b_2) \tag{8}$$

We present the comparison between the performances of Neural SSVD++ with and without social input in Table 5. The results show that incorporating social information into the model leads to significant improvement. This illustrates the importance of introducing social context into recommendation. Despite that neural network framework preserves strong nonlinearity and expressive power, social information still makes a significant contribution to the improvement of recommendation accuracy.

Table 5. Performance of Neural SSVD++ on Ciao with/without Social Input, *: $p < 0.01$

Ciao	TrustSVD	SVD++	Neural SSVD++ without social	Neural SSVD++
MAE	0.723	0.748	0.729	**0.690***
RMSE	0.956	1.001	0.995	**0.942***

5 Conclusion

In this paper, we aim to incorporate social information into the recommender systems with the framework of Neural Networks. We extend the classical SVD++ model by introducing socially connected users as implicit feedbacks and extend the matrix factorization operations with neural network schemes. The strong expressive power of neural networks contributed to the modeling of the relationship between ratings and the latent factors. In the proposed Neural SSVD++ model, the advantages of SVD++ and neural networks are kept and combined in a same model simultaneously. Extensive experiments are conducted on two real-world datasets, Neural SSVD++ has achieved significant improvements over state-of-the-art social-aware approaches.

In the future work, we aim to design a social-specific neural network structure for social-aware recommendation. Since the social network forms a user-user graph and implicit structural knowledge may not be fully captured in current model. Another interesting direction is to exploit more information in social media applications and accommodate them into the model. This also allows for a broader use of neural network approaches in recommender systems when texts and images are available.

Acknowledgement. This work was supported by Natural Science Foundation (61532011, 61672311) of China and National Key Basic Research Program (2015CB358700).

References

1. Chaney, A.J., Blei, D.M., Eliassi-Rad, T.: A probabilistic model for using social networks in personalized item recommendation. In: Recsys (2015)
2. Georgiev, K., Nakov, P.: A non-iid framework for collaborative filtering with restricted Boltzmann machines. In: Proceedings of the 30th International Conference on Machine Learning (ICML-2013), pp. 1148–1156 (2013)
3. Guo, G., Zhang, J., Yorke-Smith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI (2015)
4. Jamali, M., Ester, M.: Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: KDD (2009)
5. Jamali, M., Ester, M.: Using a trust network to improve top-n recommendation. In: Recsys (2009)
6. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2008, pp. 426–434. ACM, New York (2008). http://doi.acm.org/10.1145/1401890.1401944
7. Krohn-Grimberghe, A., Drumond, L., Freudenthaler, C., Schmidt-Thieme, L.: Multi-relational matrix factorization using Bayesian personalized ranking for social network data. In: WSDM (2012)
8. Larochelle, H., Murray, I.: The neural autoregressive distribution estimator. In: AISTATS, vol. 6, p. 622 (2011)
9. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, pp. 2177–2185 (2014)
10. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: CIKM (2008)
11. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM (2011)
12. Rafailidis, D., Crestani, F.: Joint collaborative ranking with social relationships in top-n recommendation. In: CIKM (2016)
13. Rennie, J.D., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 713–719. ACM (2005)
14. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 3–6 December 2007, pp. 1257–1264 (2007). http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization
15. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning, ICML 2007, pp. 791–798. ACM, New York (2007). http://doi.acm.org/10.1145/1273496.1273596
16. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 285–295. ACM, New York (2001). http://doi.acm.org/10.1145/371920.372071
17. Satuluri, V., Parthasarathy, S.: Bayesian locality sensitive hashing for fast similarity search. Proc. VLDB Endowment **5**(5), 430–441 (2012)

18. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference on World Wide Web, WWW 2015 Companion, pp. 111–112. ACM, New York (2015). http://doi.acm.org/10.1145/2740908.2742726
19. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
20. Tang, J., Wang, S., Hu, X., Yin, D., Bi, Y., Chang, Y., Liu, H.: Recommendation with social dimensions. In: AAAI (2016)
21. Wang, X., Lu, W., Ester, M., Wang, C., Chen, C.: Social recommendation with strong and weak ties. In: CIKM (2016)
22. Weimer, M., Karatzoglou, A., Le, Q.V., Smola, A.J.: COFI RANK - maximum margin matrix factorization for collaborative ranking. In: NIPS (2008)
23. Yang, B., Lei, Y., Liu, D., Liu, J.: Social collaborative filtering by trust. In: IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013 (2013). http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6750
24. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1633–1647 (2016)
25. Ye, M., Liu, X., Lee, W.C.: Exploring social influence for recommendation: a generative model approach. In: SIGIR (2012)
26. Zhang, S., Wang, W., Ford, J., Makedon, F.: Learning from incomplete ratings using non-negative matrix factorization. In: SDM, vol. 6, pp. 548–552. SIAM (2006)
27. Zhao, T., McAuley, J., King, I.: Leveraging social connections to improve personalized ranking for collaborative filtering. In: CIKM (2014)
28. Zhao, W.X., Huang, J., Wen, J.-R.: Learning distributed representations for recommender systems with a network embedding approach. In: Ma, S., Wen, J.-R., Liu, Y., Dou, Z., Zhang, M., Chang, Y., Zhao, X. (eds.) AIRS 2016. LNCS, vol. 9994, pp. 224–236. Springer, Cham (2016). doi:10.1007/978-3-319-48051-0_17
29. Zheng, Y., Tang, B., Ding, W., Zhou, H.: A neural autoregressive approach to collaborative filtering. arXiv preprint arXiv:1605.09477 (2016)