PA-GAN: Graph Attention Network for Preference-Aware Social Recommendation

Liyang Hou¹, Wenping Kong¹, Yali Gao¹, Yang Chen¹ and Xiaoyong Li*¹

¹ Key Laboratory of Trustworthy Distributed Computing and Service, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China

*Corresponding author’s e-mail: lixiaoyong@bupt.edu.cn

Abstract. Social recommendation has been recently proposed by incorporating trust relationship to alleviate data-sparsity and cold-start problems. However, most of existing works only focus on friends different contribution to model user representation. They ignore users have different preference on items, and share different preference with friends. To address these problems, in this paper, we propose a novel Preference-Aware Graph Attention Network (PA-GAN) for trust recommendation. And we design three modules: item aggregation for user, friend-preference aggregation for user and user aggregation for item to model users’ local and global preference. Experiments on two publicly available datasets shows the proposed model PA-GAN outperforms the state-of-the-art recommendation models, and improves performance greatly.

1. Introduction

Recommender systems (RS) are playing an important role in helping online users to find relevant information and the increasing popularity of social media helps RS to combine more information.

Recently, the exploitation of social relations for recommender system has attracted increasing interest[1][2]. By considering users’ social connections, the proposed models can utilize a much larger amount of data to alleviate data-sparsity and cold-start problems, and further improve the performance of recommender systems. [3] comes up with matrix factorization to capture explicit and implicit factors, which has inspired a considerable number of variations[4][5]. [6] replaces missing trust with similarity between users to alleviate data-sparsity and uses traditional matrix factorization to model prediction. [7] introduces extra distrust relationship to provide more accurate and personalized recommendations. [8] uses rank model to make users embedding become more similar with their friends and far away with their foes. [9] introduces local and global information in item network and social network to capture more effective information. [10] models implicit information by deep learning and does regularization with trust information, which shows great potential in RS. [11] changes matrix factorization format by adding local and global social context weight.

However, using social networks to build recommender system also faces challenges. The first is how to combine item network and social network. Generally, the state-of-the-art methods introduces graph neural network to capture these two graph structure information[12][13], so we introduce graph neural network to jointly embed two graph into the same space. The second challenge is how to distinguish local and global preference between users and friends[14]. In real life, users tend to give higher scores to their favorite items, so the ratings they give to different items reflect the user’s local preference. Meanwhile, users and their friends share different interests on items,. That is, user asks one friend for an opinion on one item, but may ask another friend for another item, because users share different
preference on items. The third challenge is how to allocate different contribution to users’ friends, because their relationship have strength. Previous studies thought the strength is equal[15], which is not suitable for capturing users’ global preference on different friends. To address challenge, the model details we proposed are as follows.

2. Materials and Methods

2.1. Definitions and Notations
Let $U = \{u_1, u_2, \ldots, u_n\}$ and $I = \{i_1, i_2, \ldots, i_m\}$ be the sets of users and items in recommender systems, where $n$ is the number of users and $m$ is the number of items. So the user-item rating matrix can be represented as $R = [R_{ui}]_{m \times n}$, which can also be called user-item graph(Item Network). In this matrix, $R_{ui}$ denotes the rating of user $u$ on item $j$. Typically, $R_{ui}$ can be any real number, but in most existing works ratings are integers in range $[1, 5]$. And in this paper, we employ 0 to represent the unknown rating from user $u$ to item $j$. In a social rating network, each user has a set $N_u$ of direct neighbors(friends), and we employ $T = [T_{uv}]_{n \times n}$ to denote the value of trust for user $u$ to user $v$. In real world, we usually do not know the exact value of trust, so we denote trust as a real number in $[0, 1]$. And we call matrix $T = [T_{uv}]_{n \times n}$ user-user graph(Social Network). Given Item Network and Social Network, we aim to predict the missing value in rating matrix $R$, which is an essential task in recommender systems. And we will introduce more details in the following parts.

2.2. The Proposed Framework PA-GAN

The overall architecture of the proposed model PA-GAN is shown in figure 1. The model consists of three parts: Item-Aggregation for User, Friend-Preference Aggregation for User, User-Aggregation for Item. The first part is to capture user’s local preference, which is user’s different ratings to items. The second part uses user-user graph to capture users and their friends’ different preference in different items and trust relationship, which we call global preference. Through these two user modeling parts, we concatenate them to generate final user latent representation. The last part is item modeling, which is to learn item latent representation. Then we send the user and item representation to multi-layer neural network to predict the missing ratings.

2.3. Item-Aggregation for User
The detailed structure model graph is shown in figure 2. This part aims to learn user representation by user-item graph, denoted as $h_i$. In real world, the value of rating usually represents how much a user likes an item. For example, Xiao Ming rates 5 for basketball and 1 for literary books, indicating that he prefers sports items, which is an important explicit factor for capturing user’s local preference on different items. According to [18], we introduce an effective approach Item-Aggregation for User to jointly combine user-item interactions and user-item different local preference latent factors. The key of this aggregation is to consider items user $u_i$ has interacted and the rating between them. To mathematically represent this aggregation, we use the following function as
\[ h_i = \sigma(W \cdot \text{Aggre}_{\text{items}}(x_{ia}, \forall a \in N(i)) + b) \]  

where \( h_i \) denotes the embedding of item-aggregation for user, \( N(i) \) denotes all items user \( u_i \) has rated. \( x_{ia} \) is a representation vector to denote local preference interaction between \( u_i \) and an item \( i_a \), and \( \text{Aggre}_{\text{items}} \) denotes item aggregation function. In addition, \( \sigma \) denotes non-linear activation function, and \( W, b \) denotes the weight and bias of the neural network.

![Figure 2 Item-Aggregation for User](image)

And to model local preference for user on different items, we introduce an rating embedding vector \( r \) to represent the latent factors. So for an interaction between user \( u_i \) and item \( i_a \), we model \( x_{ia} \) as a combination of item embedding \( q_a \) and rating embedding \( r \) via a Multi-Layer Perceptron as the following function show

\[ x_{ia} = \text{MLP}(q_a \oplus r) \]  

where \( \oplus \) denotes the concatenation operation.

And distinguish different contribution of items, we introduce attention mechanism to address the problem. And allocate \( x_{ia} \) with different weight \( a_{ia} \). The rating attention architecture are as follows:

\[ A_{ia} = W_2 \cdot \sigma(W_1 \cdot [x_{ia} \oplus r] + b_1) + b_2 \]  

\[ A_{ia} = \frac{\exp(a_{ia})}{\sum_{a \in N(i)} \exp(a_{ia})} \]  

And (4) denotes the normalization about attention scores using Softmax operation, which is necessary for indicating the contribution of different items’ local preference.

2.4. Friend-Preference Aggregation for User

![Figure 3 Friend-Preference Aggregation for User Module](image)

The detailed structure model graph is shown in figure 3. Friend-Preference Aggregation for User is to model user’s friends different preference on user’s items, which we call global preference. And due to the social relationship, a user also has different trust on it’s friends, but the trust value is not explicit. So the challenges are to capture global preference and different trust value between users and friends. For
each user $u_i$, we collect it’s friends, denoted as $f(j \in F(i))$, where $F(i)$ denotes $u_i$ direct friends. And firstly we employ joint embedding to capture user’s and friends’ relationship latent factor $t_{ij}$ as follows:

$$t_{ij} = \frac{u_i \cdot f_j}{\|u_i\| \cdot \|f_j\|}$$  \hfill (5)

Secondly, according to [16], we introduce Key Matrix($K_{nxd}$) to generate attention score for following preference embedding. And the attention scores are defined as:

$$\alpha_{jk} = t_{ij}^T K_k$$  \hfill (6)

where $K_k \in \mathbb{R}^{d}$ is each row of Key Matrix. Then the final attention scores are generated by softmax operation:

$$\alpha_{jk} = \frac{t_{ij}^T K_k}{\sum_{k=1}^{n} t_{ij}^T K_k}$$  \hfill (7)

Thirdly, the most important part in this module is to capture global preference between users and their friends. And we introduce a memory module to store each friend’s unique preference on user’s item, which denotes as $M_{nxd}$, and $d$ is the same embedding size as user embedding, $n$ can be seen as the number of preference kinds between user and friends. And how we use the preference-memory module to generate friend representation is as follows:

$$F_k = f_j \odot M_k$$  \hfill (8)

where $\odot$ denotes element-wise product of vectors. And finally the friend-preference vector are generated with attention score:

$$f_{p_j} = \sum \alpha_{jk} F_k$$  \hfill (9)

So the vector $f_{p_j}$ can be seen as the friend vector with it’s global preference on user’s item. Now that we have friend-preference representation, we need to consider different friends’ contribution to user. Similar with Item-Aggregation For User, we use user embedding and friend-preference embedding to generate attention score for another user Modeling.

2.5. User-Aggregation for Item

The detailed structure model graph is shown in figure 4, which is similar as Item-Aggregation for User. This part aims to learn item representation by user-item graph, denoted as $i_a$. Before train item embedding, we need to consider that there are great difference about users rating on same item. To address this challenge, we introduce attention mechanism to capture this pattern again. To mathematically represent this aggregation, we use the following function as

$$f_{ai} = MLP(i_a \oplus u_i)$$  \hfill (10)

$$\alpha_{ai} = W_2 \cdot (\sigma(W_1 \cdot [f_{ai} \oplus i]) + b_1) + b_2$$  \hfill (11)

$$\alpha_{ai} = \frac{\exp(\alpha_{ai})}{\sum_{i \in N(a)} \exp(\alpha_{ai})}$$  \hfill (12)
where \( i_a \) denotes the embedding of item \( a \), \( u_i \) denotes user \( i \) who rates item \( a \), and \( N(a) \) denotes the set of all users who rates item \( a \), and \( z_a \) denotes the final representation of item \( a \) with unique preference from different users.

### 2.6. Model Learning

In this section, we will introduce the final task: rating prediction through all modules above. With the item-aggregation embedding \( h_i \) for user, friend-preference aggregation embedding \( f_p_j \) for user and user-aggregation embedding for item \( z_a \), we will feed them into MLP for rating prediction as:

\[
\hat{r}_{ia} = MLP(MLP(h_i \oplus f_p_j \oplus z_a))
\]

And we employ the commonly used loss function for optimization, which considers the difference between true rating \( R_{ij} \) and predicted rating \( \hat{R}_{ij} \).

\[
\text{Loss} = \frac{1}{N} \sum_{i,j} (R_{ij} - \hat{R}_{ij})^2,
\]

where \( N \) is the number of samples in dataset.

### 3. Experiments

In our experiments, we use two public accessible datasets Epinions and Ciao to evaluate the performance. The statistics of the datasets are summarized in Table 1. And we compare our model PA-GAN with SoRec[17], DeepSoR[18], GraphRec[19]. During the whole experiments, we introduce metrics Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the prediction accuracy.

#### Table 1. The description of datasets.

| Dataset | Users | Items | Ratings | Trust Relation | Density(Rating) | Density(Trust) |
|---------|-------|-------|---------|----------------|-----------------|----------------|
| Epinions | 49289 | 139738 | 664824 | 487183 | 0.0097% | 0.0201% |
| Ciao | 7375 | 106797 | 284086 | 111781 | 0.0361% | 0.2056% |

The rating prediction results are shown in Table 2. Our model PA-GAN are generally better than other models. DeepSoR shows better than SoRec, which indicates deep neural network has great power. GraphRec performs better than SoRec and DeepSoR, which shows graph neural network and attention mechanism is very suitable for capturing graph data information. And our model is the best, that is, capturing local and global preference for users can improve performance of recommender system greatly.

#### Table 2. The results of compared models.

| Dataset | Metrics | SoRec | DeepSoR | GraphRec | PA-GAN |
|---------|---------|-------|---------|----------|--------|
| Epinions (60%) | MAE | 0.8976 | 0.8318 | 0.8293 | 0.8184 |
| Epinions (80%) | RMSE | 1.1563 | 1.1135 | 1.0670 | 1.0648 |
| Ciao (60%) | MAE | 0.8751 | 0.8393 | 0.8219 | 0.8112 |
| Ciao (80%) | RMSE | 1.1437 | 1.0971 | 1.0621 | 1.0612 |
| Ciao (60%) | MAE | 0.8489 | 0.7813 | 0.7930 | 0.7864 |
| Ciao (80%) | RMSE | 1.0738 | 1.0437 | 1.0160 | 1.0082 |

### 4. Conclusions

We proposed a novel Preference-Aware Graph Attention Network (PA-GAN) for rating prediction. We also design three modules to capture local and global preference between users and their friends and
introduce attention network to weight friends’ contribution. And the experiment results on two real-world datasets shows that our model outperforms most recommendation models. In the future, we are interested in exploring users dynamic preference, and combine distrust relationship between users.

Acknowledgments
This work was supported by the National Key Research and Development Program of China (No. 2018YFC0831501), the NSFC-General Technology Fundamental Research Joint Fund U1836215, and Capital Science and Technology Leading Talent Training Project, China (Z191100006119030).

References
[1] Tang, J., Hu, X., & Liu, H. (2013). Social recommendation: a review. Social Network Analysis and Mining, 3(4), 1113-1133.
[2] Ren, Z., Liang, S., Li, P., Wang, S., & de Rijke, M. (2017, February). Social collaborative viewpoint regression with explainable recommendations. In Proceedings of the tenth ACM international conference on web search and data mining (pp. 485-494).
[3] Guo, G., Zhang, J., & Yorke-Smith, N. (2015, February). Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 29, No. 1).
[4] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web (pp. 173-182).
[5] He, X., Zhang, H., Kan, M. Y., & Chua, T. S. (2016, July). Fast matrix factorization for online recommendation with implicit feedback. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 549-558).
[6] Saleem, F., Iltaf, N., Afzal, H., & Shahzad, M. (2019, June). Using Trust in Collaborative Filtering for Recommendations. In 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 214-222). IEEE.
[7] Ma, H., Lyu, M. R., & King, I. (2009, October). Learning to recommend with trust and distrust relationships. In Proceedings of the third ACM conference on Recommender systems (pp. 189-196).
[8] Forsati, R., Barjasteh, I., Masrour, F., Esfahanian, A. H., & Radha, H. (2015, September). Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations. In Proceedings of the 9th ACM Conference on Recommender Systems (pp. 51-58).
[9] Tang, J., Aggarwal, C., & Liu, H. (2016, April). Recommendations in signed social networks. In Proceedings of the 25th International Conference on World Wide Web (pp. 31-40).
[10] Deng, S., Huang, L., Xu, G., Wu, X., & Wu, Z. (2016). On deep learning for trust-aware recommendations in social networks. IEEE transactions on neural networks and learning systems, 28(5), 1164-1177.
[11] Tang, J., Hu, X., Gao, H., & Liu, H. (2013, August). Exploiting local and global social context for recommendation. In IJCAI (Vol. 13, pp. 2712-2718).
[12] Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint arXiv:1606.09375.
[13] Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864).
[14] Ye, M., Liu, X., & Lee, W. C. (2012, August). Exploring social influence for recommendation: a generative model approach. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (pp. 671-680).
[15] Jamali, M., & Ester, M. (2010, September). A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the fourth ACM conference on Recommender systems (pp. 135-142).
[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I.
(2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

[17] Ma, H., Yang, H., Lyu, M. R., & King, I. (2008, October). Sorec: social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM conference on Information and knowledge management (pp. 931-940).

[18] Fan, W., Li, Q., & Cheng, M. (2018, April). Deep modeling of social relations for recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).

[19] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., & Yin, D. (2019, May). Graph neural networks for social recommendation. In The World Wide Web Conference (pp. 417-426).