Social Recommendations: Communities and Graph Neural Networks for Social Recommendations

Lior Shaposhnikov, Hanan Libhaber

July 17, 2021

Abstract

With the increase in popularity of social networks and the large quantities of data gathered in social media, social recommendations have become an important topic of research for recommender systems. Utilizing social relations data to better understand user's preferences by their direct neighbors' preferences be effective for many recommender systems. Graph Neural Networks (GNNs) have recently been applied with success on recommender systems, including social recommender systems. In this paper, we extend the work of GraphRec Fan et al. (2019) on their social relations part by finding users' communities using a community detection algorithm to interpolate social relations data from distant neighbors. We show minor improvements over the current method which can suggest that relevant information for users, lays in a group of interest and not only from direct neighbors. The implementation code is available at: `https://github.com/Lior157/ComGraphRec`

1 Introduction

Social recommender systems which include social relations are getting more attraction from research in recent years Tang et al. (2016); Shokeen and Rana (2019); Fan et al. (2019). The popularity of deep learning techniques is also effecting recommender systems Wang et al. (2015). Recently, GNNs, which are a specific type of graph neural network, have also started being used for recommender systems Ying et al. (2018), and more recently, on social recommender systems Fan et al. (2019); Guo and Wang (2020); Huang et al. (2021). Graphs structures of user-to-item and user-to-user can be are useful for data passing through neighbors

GraphRec Fan et al. (2019) have shown good results on social relations, though, it has a limitation of only addresses direct neighbors. We argue that aggregation of information from distant neighbors that fit the user's community can be beneficial and extend the learning of the user representation. It is possible to extend the GNN model to reach distant neighbors, but due to the structure of the graph, this can introduce more noise and increase the complexity of the network.

In this work, we will create communities based on the social relations graph, using a community detection algorithm Traag et al. (2019) and extend the work of GraphRec to enrich user information with information from distant neighbors or communities with similar interest.

Our contribution of this paper:

- We propose an extension to the GraphRec Fan et al. (2019) model to extract additional information from distant neighbors and related communities
- We jointly aggregate information of social relations from both close neighbors using the original GNN, and from a community using a community detection algorithm.
- We show minor improvements on two real world datasets compared to the the original GraphRec model.

In the next sections we will present the follows: We will cover some background details in 2. Next, we will review related work to our model in 3. Then, We will present our method and changes made to the original model in 4. We will present the experiments details in 5 and present the results in 6. Lastly, we will discuss and conclude our work.

2 Background

2.1 Social recommender system

The rapid growth and popularity of social media websites such as Facebook and Twitter introduced a large amount of data containing social social relations of friends following each other. Social recommendation derived from social relations between users as an addition to the regular recommender system. Social relations can be describe as users follows other users like friends in Facebook, or a trust based relations like in Epinions, a online shopping social review websites, where user can review items and also create a trust relationship with other users by adding them as friends.

2.2 Communitiy detection algorithms

Community-based techniques work on network structure and can find groups in large networks. A popular algorithm of community detection is Louvain Blondel et al. (2008) which optimizes a quality function and consists of two phases: (1) local moving of nodes; and (2) aggregation of the network. Recently, the Leiden Traag et al. (2019) showed to perform better than the Louvain algorithm in terms of well connected comunities which they claimed that Louvain returns communities that some communities are not well connected or even disconnected. The Leiden algorithm consists of three phases: (1) local moving of nodes, (2) refinement of the partition, and (3) aggregation of the network based on the refined partition. We will use the Leiden algorithm to discover communities from social relations of users.

2.3 GraphRec

GraphRec Fan et al. (2019) is our base model which we extend our work upon. This paper presented a graph neural network framework for social recommendations. Their contribution was consist of three parts. First they faced a challenge of connecting user-to-item and user-to-user graphs inherently. Second, they showed they were able to captured interactions and opinions between users and items jointly. Their third part of their contribution was a way to distinguish between heterogeneous strength of social relations, which they used an attention mechanism to solve. We will use parts of their solution and extend the social aggregation part by adding another part that utilize social relations and concatenate it with their own, using the same techniques they applied.

3 Related Work

With the increasing popularity of social media, social relations have become an essential part of recommender systems and attracted attention from this field of research (Ma et al. (2008); Tang et al. (2016); Li et al. (2015)). In SoRecMa et al. (2008) they used a co-factorization, both on user-terms rating matrix and social network at the same time. This method aimed to solve data sparsity problem. SoDimRec Tang et al. (2016) have shown that users related groups is useful with weakly connected users or even disconnected. At our work we will consider only strongly connected users. In Pan et al. (2012) they used a nearest neighbors approach on social relations to enhance matrix factorization. They showed how trust information improves top-k recommendations. Li et al. (2015) used several overlapping community detection algorithms on social relations graph, to find overlapping communities and incorporate them into matrix factorization framework. Their motivation was based on the assumption that weights should be treated differently for each community based on user preferences. They showed how multiple communities help improves rating prediction.

Recently, GNNs, which are a type of neural net-

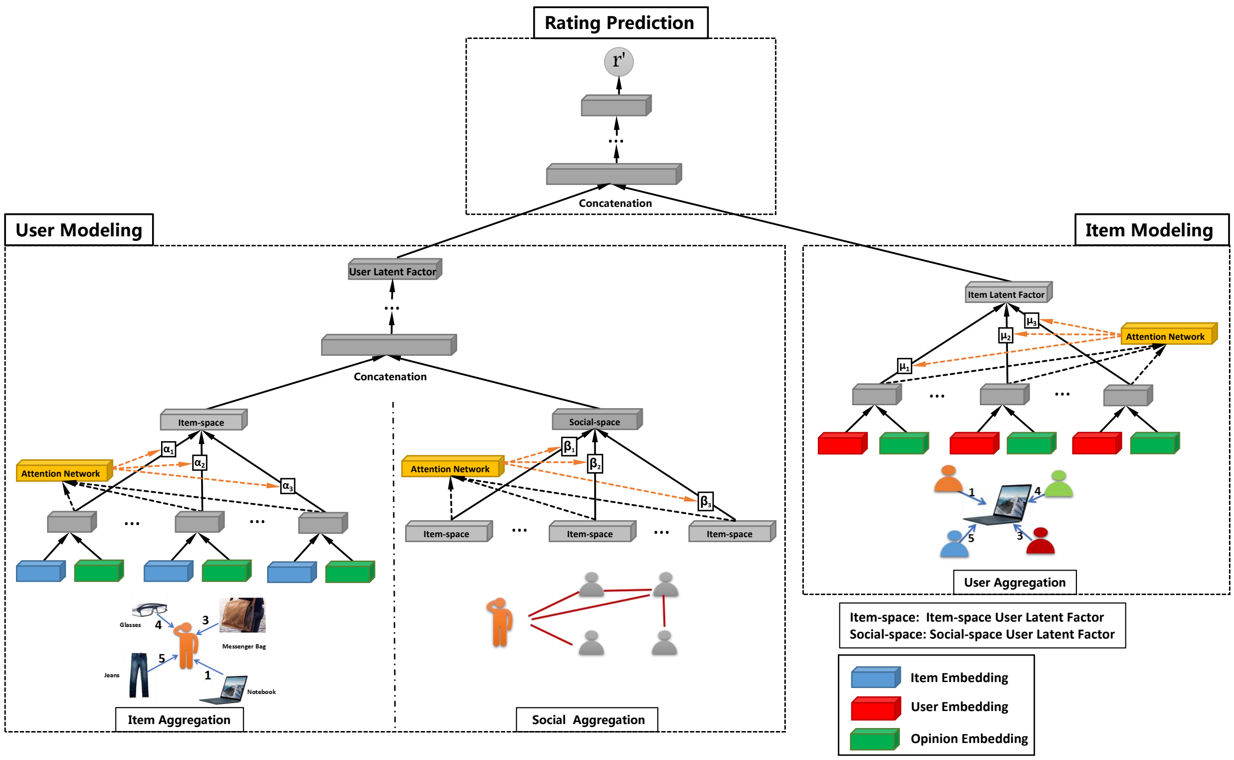

Figure 1: The GraphRec architecture as it presented in Fan et al. (2019). Our method will focus on extending the social aggregation part from the User Modeling component.

works that works on graph data, are also being used with social relations (Wu et al. (2020); Fan et al. (2019); Huang et al. (2021)) In Fan et al. (2019) they used GNN on the social relations graph to propagate information from neighbors to the current user. They used attention mechanism to differentiate between weak and strong relations. While aggregation of social information was applied in Fan et al. (2019), it would be more difficult to propagate information from long distant neighbors. information from distant neighbors can be reached using graph neural networksSun et al. (2020), but this can introduce more noise and still will be limited as the distance from the neighbors becomes greater. Communities detection algorithm have also been used to create a community with shares interests based on social relations. in Zhou et al. (2010) they proposed a user recommendation framework using greedy community detection algorithm Clauset et al. (2004) to find user interest through tag-graph based community.

4 Proposed Method

Propagate information from distant neighbors in a graph could be a challenging task. Noise might be introduced to the model while trying to retrieve relevant information. Our method aims to aggregate information from distant neighbors as well as directed neighbors. We will use a community detection algorithm named Leiden Traag et al. (2019) on our social relations graph to retrieve communities of nodes. The social space of each node will be calculated based on neighbors and other community members. Like in GraphRec Fan et al. (2019), we applied an attention mechanism to separate strong and weak strengths of user connections.

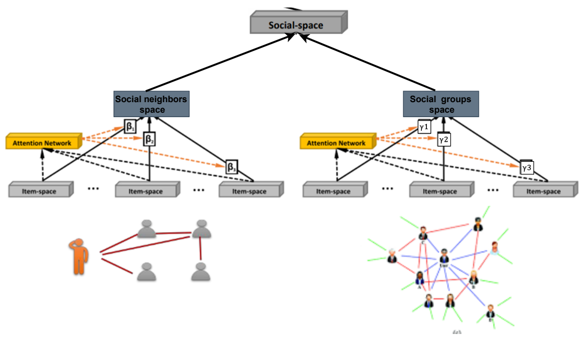

Figure 2: Modified social aggregation part from the User Modeling component of GraphRec. On the right side, there is our addition to the model. Each user aggregate information from other users in his community, results in an embedding social space from it community.

We will use the same notations as they used in GraphRec for the social aggregation part of neighbors, and apply it on our community part. We proposed User Modeling for user $\mathbf{u}_i$ consist of Item Aggregation $\mathbf{h}_i^I$ and Social Aggregation mentioned as $\mathbf{h}_i^S$. We proposed to extend Social Aggregation to use Community Aggregation $\mathbf{h}_i^G$ in addition to existing direct neighbors Aggregation in original GraphRec model.

4.0.1 Community Aggregation

To represent user from community perspective, we propose to add to the GraphRec model, community-space user latent factors, which is aggregates the community-space user latent factors from community's graph. Specially, the community-space user latent factor of $\mathbf{u}_i, h_i^G$ is to aggregate the community-space user latent factors of users in $\mathbf{u}_i$'s community associates $\mathbf{G(i)}$, as the follows:

$$\mathbf{h}_i^G = \sigma(\mathbf{W} \cdot Aggre_{\text{community}(\{\mathbf{h}_o^I, \forall o \in G(i)\}) + \mathbf{b})} \tag{1}$$

where $Aggre_{community}$ denotes the aggregation function on user's community associates. One natural aggregation function for $Aggre_{community}$ is the mean operator which take the element-wise mean of the vectors in $\{\mathbf{h}_o^I, \forall o \in G(i)\}$, as the following function:

$$\mathbf{h}_i^G = \sigma(\mathbf{W} \cdot \left\{ \sum_{o \in G(i)} \gamma_i \mathbf{h}_o^I \right\} + \mathbf{b}) \tag{2}$$

Where γ_i is fixed to $\frac{1}{|G(i)|}$ for all neighbors for the mean-based aggregator. It assumes that all neighbors contribute equally to the representation of user $\mathbf{u}_i$. However, strong and weak ties are mixed together, and users are likely to share more similar tastes with strong ties than weak ties. We can use an attention mechanism with a two-layer neural network to extract these users that are important to influence $\mathbf{u}_i$, and model their tie strengths, by relating social community attention γ_{io} with $\mathbf{h}_o^I$ and the target user embedding $\mathbf{p}_i$, as below,

$$\gamma_{io}^* = \mathbf{w}_2^T \cdot \sigma(\mathbf{W}_1 \cdot [\mathbf{h}_o^I \oplus \mathbf{p}_i] + \mathbf{b}_1) + b_2 \tag{3}$$

$$\gamma_{io} = \frac{exp(\gamma_{io}^*)}{\sum_{o \in G(i)} exp(\gamma_{io}^*)} \tag{4}$$

where the γ_{io} can be seen as the strengths between users.

4.0.2 Learning Social Latent Factors

To learn the novel proposed new social space consist of direct neighbors' aggregation $\mathbf{h}_i^S$ and community aggregation $\mathbf{h}_i^G$. We concatenated the neighbors-space and community-space before feeding them into MLP. Formally, the new Social Latent Factor $\mathbf{h}_i^{S'}$ is defined as,

$$\mathbf{c}_1 = \left[\mathbf{h}_i^G \oplus \mathbf{h}_i^S\right] \tag{5}$$

$$\mathbf{c}_2 = \sigma(\mathbf{W}_2 \cdot \mathbf{c}_1 + \mathbf{b}_2) \tag{6}$$

$$\ldots$$

$$\mathbf{h}_i^{S'} = \sigma(\mathbf{W}_l \cdot \mathbf{c}_{l-1} + \mathbf{b}_l) \tag{7}$$

where l is the index of a hidden layer.

The rest of the model stays exactly the same as it presented in GraphRec. Further details can found in the paper Fan et al. (2019).

5 Evaluation

5.0.1 Datasets

We tested two datasets in our experiment Ciao and Epinions[1]. Epinions and Ciao are popular social media for online sopping websites. Both websites allows users to publish reviews and ratings on items with a score between 1 to 5. In addition, users can follow other users which indicate a trust systems. In our dataset we have two graphs: The first graph contains user-items and the rating user gave to that item. The second graphs contains users that are following other users. The number 1 is attached to indicate there is a relation between the users. We used the same datasets as reported in GraphRec Fan et al. (2019)

5.0.2 Pre-processing

To replicate the base model and add our implementation, we used an unofficial implementation of GraphRec [2] In the pre-processing stage, they removed users that rated less than 5 items For each dataset we perform two types of splits. One is 80%-10%-10% for train, validation and test respectively, and the other one is 60%-20%-20% for the train, validation and test respectively.

5.0.3 Baseline

The algorithm we used as baseline is the base algorithm for our model - GraphRec. We replicate the model and since our results were different than presented in the GraphRec Fan et al. (2019), we considered only our replicated results, and did not use the baselines they have tested.

5.0.4 Metrics

We will evaluate our model with two well-known metrics named MAE and RMSE:

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N} \tag{8}$$

[1] https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm

[2] https://github.com/lcwy220/Social-Recommendation

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2}{N}}, \tag{9}$$

Smaller values of MAE and RMSE indicates better predictive accuracy.

5.0.5 Parameters

For the parameters of the base model we used the parameters presents in the unofficial implementation of GraphRec[2]. For the baseline model the embedding size dimension was 64, the batch size was 128 and the learning rate was 0.001. The number of hidden layers was three and the size of the hidden layer was the same as the embedding size, as it was in the original implementation. The dropout parameter in the hidden layers was set to 0.5. Each experiment was trained for 30 epochs.

Table 1: Statistics of the datasets

Dataset	Ciao	Epinions
# of Users	7,375	40,163
# of Items	106,797	139,738
# of Ratings	283,119	664,824
# of Density (Ratings)	0.0365%	0.0118%
# of Social Connections	111,781	487,183
# of Density (Social Relations)	0.2055%	0.0302%

6 Results

We compared the results of our replication of GraphRec with our extended model. Due to the lack of pre-processing part in the original implementation of the GraphRec model, we used an unofficial implementation that reproduced the original results, but it is uncertain that the data was the same after the preprocess part. We can compare the results from Table 2. The result from our model with the additional communities part and show some improvements in all of the tests on the Ciao dataset and on some of the Epinions dataset. We can see that the differences in results are higher for the MAE and very small for RMSE. We deduce from these results, that even though the difference in results is small, the added complexity of our model with the additional layers, did not hurt the model, and even improve it a little. We believe that retrieving information from distant neighbors or communities with similar interest can be helpful.

Table 2: Performance comparison of GraphRec and our model

Training	Metrics	Algorithms	
		GraphRec	Ours
Ciao (60%)	MAE	0.7540	**0.7224**
	RMSE	1.0093	**1.0009**
Ciao (80%)	MAE	0.7387	**0.7104**
	RMSE	0.9794	**0.9759**
Epinions (60%)	MAE	0.8103	**0.8068**
	RMSE	**1.0791**	1.0796
Epinions (80%)	MAE	0.8027	**0.7998**
	RMSE	**1.0669**	1.0679

7 Discussion

The results in previous sections showed improvement from the addition communities on the social relations compared to the original model that contained direct neighbors. Due to the changes in data and lack of pre-processing stage in the code or the chosen parameters of the original model, we could not exactly replicate the results GraphRec declared in their paper. There have been several attempts to replicate the results. one replication [3] was not able to replicate the results in the paper. The second implementation [2] was built on top of the first one and added an additional pre-processing stage of removing users with less than 5 items. Although the results are very similar and even slightly better in the replication than in the original model, we can not guarantee that the data they trained their model is the same and could not use their published results, including other baselines. We have attempted to reach the main author of the GraphRec paper and ask for the missing details, but we did not get a reply.

[3] unofficial implementation of GraphRec: `https://github.com/Wang-Shuo/GraphRec_PyTorch`

8 Conclusion and Future Work

In this paper, we presented an extension to the paper GraphRec Fan et al. (2019) which leverage social relations and adding communities in addition to data from direct neighbors. We showed that additional information from social relations with communities can improve the rating prediction of the model compared to aggregation of social relations from direct neighbors only. We believe that distant neighbors of a user, might contain useful information for the representation of a user. Furthermore, we were able to join the social space information from our community, with the direct neighbors social space. Because we could not be sure the replication of the test and process was exactly the same as tested in the original paper Fan et al. (2019), we did not use the baselines presented in the paper. For future work, a possible direction is exploring new ways to model communities and analyze their effects on the user.

References

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, pages 417–426, 2019.

Z. Guo and H. Wang. A deep graph neural network-based mechanism for social recommendations. *IEEE Transactions on Industrial Informatics*, 17(4):2776–2783, 2020.

C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, L. Bo, H. Xing, X. Lai, and Y. Ye. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

H. Li, D. Wu, W. Tang, and N. Mamoulis. Overlapping community regularization for rating prediction in social recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 27–34, 2015.

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, 2008.

R. Pan, P. Dolog, and G. Xu. Knn-based clustering for improving social recommender systems. In *International workshop on agents and data mining interaction*, pages 115–125. Springer, 2012.

J. Shokeen and C. Rana. Social recommender systems: techniques, domains, metrics, datasets and future scope. *Journal of Intelligent Information Systems*, pages 1–35, 2019.

Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, and Y. Qu. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 222–229, 2020.

J. Tang, S. Wang, X. Hu, D. Yin, Y. Bi, Y. Chang, and H. Liu. Recommendation with social dimensions. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

V. A. Traag, L. Waltman, and N. J. Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244, 2015.

S. Wu, F. Sun, W. Zhang, and B. Cui. Graph neural networks in recommender systems: a survey. *arXiv preprint arXiv:2011.02260*, 2020.

R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.

T. C. Zhou, H. Ma, M. R. Lyu, and I. King. Userrec: A user recommendation framework in social tagging systems. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.