Relational Metric Learning with Dual Graph Attention Networks for Social Recommendation

Xiaodong Wang, Zhen Liu^(✉), Nana Wang, and Wentao Fan

School of Computer and Information Technology, Beijing Jiaotong University,
Beijing 100044, China

jackwangsysu@gmail.com, {zhliu,nnwang,fwt.one}@bjtu.edu.cn

Abstract. Existing social recommenders typically incorporate all social relations into user preference modeling, while social connections are not always built on common interests. In addition, they often learn a single vector for each user involved in two domains, which is insufficient to reveal user's complex interests to both items and friends. To tackle the above issues, in this paper, we consider modeling the user-item interactions and social relations simultaneously and propose a novel metric learning-based model called RML-DGATs. Specifically, relations in two domains are modeled as two types of relation vectors, with which each user can be regarded as being translated to both multiple item-aware and social-aware representations. Then we model the relation vectors by neighborhood interactions with two carefully designed dual GATs to fully encode the neighborhood information. Finally, the two parts are jointly trained under a dual metric learning framework. Extensive experiments on two real-world datasets demonstrate that our model outperforms the best baseline by 1.91% to 4.74% on three metrics for top-N recommendation and the performance gains are more significant under the cold-start scenarios.

Keywords: Social recommendation · Metric learning · Graph attention networks · Neighborhood interactions

1 Introduction

Nowadays people are troubled by the information overload problem caused by the explosively growing online contents. Recommendation system, which aims at addressing this problem by providing personalized items to each user, has become an indispensable element of many web applications. Recently, top-N recommendation based on implicit feedback has attracted much research interest since implicit feedback is much more abundant and easier to collect in practice [10]. Among the various recommendation methods, collaborative filtering (CF) stands out to show good performance, with the assumption that similar users tend to share similar preferences [11, 17]. However, the performance of CF models are often hindered by the data sparsity and cold-start issues [3].

With the prevalence of the online social platforms, social recommendation has emerged as a promising solution to address the above two issues. It assumes that people tend to spread their interests to their social connections and focuses on combining social information with user-item interactions for better item recommendations [19]. In conventional social recommenders [8, 12], social relations are typically modeled as an ensemble term or a regularization based on the matrix factorization (MF). There also exists hybrid models which collectively factorize the user-item interaction matrix and trust matrix [4]. Recently, researchers also designed more advanced neural models for social recommendation and showed state-of-the-art performance [1, 2, 18]. E.g., [1] models both aspect-level and friend-level differences for social recommendation by attention mechanism and memory network. Although the above models are effective, we argue that they have three common limitations: Firstly, they typically incorporate all social relations into user preference modeling, ignoring the fact that users may have totally opposite tastes on specific items. Secondly, they often learn a single vectorized representation for each user, which is insufficient to reveal user’s complex interests to both items and friends. Finally, none of them have fully considered the neighborhood information in both domains.

Fig. 1. The two kinds of relation vectors.

In this paper, we focus on implicit feedback and propose a dual metric learning framework to handle the above issues. As users involve in two heterogeneous graphs, we model the user-item interactions and social relations simultaneously instead of directly incorporating social information into user embeddings. Specifically, as shown in Fig. 1, we propose to model the two kinds of relations as interaction vector (e.g., $\vec{r}_{11}$) and social vector (e.g., $\vec{s}_{12}$) respectively. Then in the metric space, each user can be regarded as being translated to both multiple item-aware points by the interaction vectors or to multiple social-aware points by the social vectors, which results in a much finer-grained model. In addition, from the perspective of CF, each user-item pair can be seen as being related by their respective neighboring items and users. Analogously, two users in the social domain are also connected by their neighboring friends. Therefore, we model the two kinds of relation vectors by neighborhood interactions, in which

neighborhood-based representations in the two domains are generated by two carefully designed dual GATs [16]. Then they are fed into two corresponding MLPs to model the complex interactions between two neighborhoods as relation vectors. Finally, the two relation modeling parts are incorporated into a joint model to mutually enhance each other.

Our main contributions are summarized as followings:

- We propose a novel social recommendation model RML-DGATs which considers utilizing social information by simultaneously modeling the user-item interactions and social relations under a dual metric learning framework.
- Relations in two domains are modeled as relation vectors to translate each user to both multiple item-aware and social-aware representations. We model the two kinds of relation vectors by two carefully designed dual GATs, which could fully encode the neighborhood information in an explicit manner.
- Under extensive experiments conducted on two real-world datasets, our model consistently outperforms state-of-the-art baselines on three metrics for top-N recommendation tasks, especially in the cold-start scenarios.

2 Related Work

Top-N recommendation with implicit feedback has attracted more and more research interest due to its good applicability in practice [10]. However, traditional MF approaches are incapable of handling such a scenario for the absence of negative feedback. To address this issue, [14] first proposed a pairwise ranking assumption tailored for implicit feedback which states that user tend to rank items she consumed over those unobserved, then it optimizes MF with BPR-OPT criterion. [5] further extends MF to a neural architecture to model the implicit feedback as complex nonlinear user-item interactions. Recently, [7] argues that inner product used in most previous methods violates the import triangle inequality property and adopts the Euclidean distance as the relation modeling metric.

To alleviate the data sparsity and cold-start issues, many efforts has been devoted to incorporate social relations for better recommendation. These studies could be broadly categorized into two groups: MF-based and DNN-based. MF-based methods typically take social information as an ensemble term or a regularization. [22] extends BPR with the assumption that users tend to rank items consumed by their friends over those unobserved. [4] collectively factorizes the rating matrix and trust matrix to share the same user latent vectors. However, the above MF-based methods could only capture linear information encoded in social network, so several recent studies proposed to learn more complex features using deep neural networks [1, 2, 18]. Authors in [2] designed a NSCR model based on NCF [5] to model the social relations as a graph regularization. To learn non-linear features of each user from social relations, [18] presented a deep model which substitutes the user latent modelling part of PMF with DNN. A newly proposed method SAMN [1] leveraged attention-based memory network to model both aspect-level and friend-level differences for social recommendation.

3 The Proposed Model

In this section, we introduce the proposed RML-DGATs model and the overall architecture is shown in Fig. 2. We start with formulating the social recommendation problem concerned in this paper and then elaborate the three main components of the model: the ID Embedding part, the Relation Modeling part and the Joint Learning part.

3.1 Problem Formulation

Let $G = (U \cup I, Y)$ be the user-item interaction graph, where U is the user set with n users and I is the item set with m items. $Y \in R^{n \times m}$ is the user-item interaction matrix with $Y_{ui} = 1$ if user u has interacted with item i , otherwise $Y_{ui} = 0$. As social connections can be unilateral (e.g., followings on Twitter) or bilateral (e.g., friendships on Facebook), we describe the social network as a directed graph $S = (U, T)$. $T \in R^{n \times n}$ represents the social relations between users which satisfies $T_{uv} = 1$ if user v is in user u 's trust list. Note that we term u as the trustor and v as the trustee in T_{uv} . Then, the social recommendation problem can be described as:

Social Recommendation. Given an user-item interaction graph G and a social network S , the social recommendation aims to learn a mapping function f to predict users' preferences to items as: $\hat{R}_{ui} = f(G, S)$, where $\hat{R}_{ui} \in R^{n \times m}$ is the predicted preference scores.

Fig. 2. The proposed RML-DGATs model.

3.2 ID Embedding Part

Let $P \in R^{n \times d}$ and $Q \in R^{m \times d}$ denote the low-dimensional embedding matrices of users and items, with d represents the dimension size of latent vectors. Given an user u , one of her interacted item i and an user v in her trust list, the ID Embedding part takes one-hot representation of triplet (u, i, v) as input and performs indexing operations from P and Q according to their one-hot IDs. Then it outputs the corresponding latent vectors p_u, q_i and p_v for u, i and v , respectively. Note that p_u could be regarded as the shared essential preference between the item and social domains.

3.3 Relation Modeling Part

The Relation Modeling part aims to model the user-item interaction relations and user-user social relations simultaneously. Next we will detail the user-item interaction modeling and social modeling respectively.

Interaction Modeling. In the user-item interaction graph, the items user has interacted form its neighborhood and they can reflect her preferences to some extent. Analogously, item's neighbors are the users who interacted with it and its attributes could be revealed by these users. We propose a dual GATs which consists of an user-level GAT and an item-level GAT to aggregate item's or user's neighborhood information.

User-Level GAT. Let $N(i)$ be the set of users who have interacted with item i . The user-level GAT aggregates latent vectors of item i and users in $N(i)$ attentively and then outputs the neighborhood-based item representation q_i^N . We define the aggregation function as:

$$q_i^N = \sigma(W \cdot \sum_{j \in N(i) \cup i} \alpha_{ij} h_{ij}) \quad (1)$$

where $h_{ij} = q_i$ if $j = i$, otherwise $h_{ij} = p_j$ for $j \in N(i)$. α_{ij} is the attention weight and we parameterize it with a two-layer neural network as:

$$\alpha_{ij} = \text{Softmax}(h^T \cdot \sigma(W' \cdot [q_i, h_{ij}] + b)) \quad (2)$$

where σ is the nonlinear activation function and $[.]$ denotes the concatenation operation. In addition, $\Theta_1 = [W, W', h, b]$ are the model parameters of the user-level GAT, in which W is the weight matrix for the GAT layer and W', h, b are the weights and bias for the attention network. Note that we also tried other forms of attention networks such as simple inner product or inner product with nonlinear activation, but we empirically found that Eq. (2) performs best.

Item-Level GAT. Latent vectors of user u and its neighboring items in $C(u)$, namely the set of items u has interacted, are aggregated to generate the neighborhood-based user representation t_u^N in item domain. Analogously, we define the aggregation function as:

$$t_u^N = \sigma(W \cdot \sum_{j \in C(u) \cup u} \beta_{uj} h_{uj}) \quad (3)$$

$$\beta_{uj} = \text{Softmax}(h^T \cdot \sigma(W' \cdot [p_u, h_{uj}] + b)) \quad (4)$$

where $h_{uj} = p_u$ if $j = u$, otherwise $h_{uj} = q_j$ for $j \in C(u)$.

Then a MLP which takes the two representations as input is designed to model the complex interactions between the two neighborhoods as the *interaction vector* r_I . Formally, we empirically design a two-layer tower structure as:

$$r_I = \sigma(W_2 \cdot \sigma(W_1 \cdot [q_i^N, t_u^N] + b_1) + b_2) \quad (5)$$

where W_1, W_2, b_1 and b_2 are the weights and biases of the MLP and we denote them as $\Theta_2 = [W_1, W_2, b_1, b_2]$.

Social Modeling. The social relation between two users can be seemed as related by their respective neighborhoods including themselves. Given user u and user v in her trust list, we propose another dual GATs which consists of two friend-level GATs to aggregate user's neighboring friends into neighborhood-based user representation.

Friend-Level GAT. For user u , the latent vectors of u and users in her trust list denoted as $v \in N(u)$ are aggregated into the neighborhood-based user representation s_u^N in social domain. We define the aggregation function as:

$$s_u^N = \sigma(W \cdot \sum_{j \in N(u) \cup u} \mu_{uj} h_{uj}) \quad (6)$$

$$\mu_{uj} = \text{Softmax}(h^T \cdot \sigma(W' \cdot [p_u, h_{uj}] + b)) \quad (7)$$

For user v , her neighborhood-based representation p_v^N is generated by the same friend-level GAT and we only need to replace p_u with p_v . Then we also model the *social vector* r_S by feeding s_u^N and p_v^N into a two-layer MLP as:

$$r_S = \sigma(W_2 \cdot \sigma(W_1 \cdot [s_u^N, p_v^N] + b_1) + b_2) \quad (8)$$

Note that we use the same parameters $\Theta_1 = [W, W', h, b]$ for the four different GATs and $\Theta_2 = [W_1, W_2, b_1, b_2]$ for the two MLPs, as we empirically found that it works the best. The rationale behind maybe that users and items share similar neighborhood structures and unifying them to train the same parameters booms the training process and reduce the risk of overfitting. In addition, as the number of users' or items' neighbors vary greatly, traversing all of them is time consuming and also may introduce too much noise. Therefore, we set a threshold K and for those nodes with more than K neighbors, we randomly sample K of them to aggregate their features.

3.4 Joint Learning Part

Scoring Functions. Since we consider the user-item interactions and social relations individually, we define the following two scoring functions to evaluate the qualities of interaction vector and social vector learned by the Relation Modeling part:

$$s_1(u, i) = \|p_u + r_I - q_i\|_2^2, \quad s_2(u, v) = \|p_u + r_S - p_v\|_2^2 \quad (9)$$

Given latent vectors of user u , item i and their relation vector r_I , the interaction scoring function $s_1(u, i)$ penalizes any deviation of $p_u + r_I$ from q_i brought by the relation vector r_I . Analogously, the social scoring function $s_2(u, v)$ calculates the distance between user's translated vector $p_u + r_S$ and p_v . The smaller the scoring values s_1 and s_2 are, the higher preference user u will show to item i and the more user u will trust user v .

Loss Function. We adopt the widely-used pairwise hinge loss which was tailored for item-ranking task [7]. For each positive tuple (u, i) which satisfies $Y_{ui} = 1$, we pair it with t sampled negative tuples (u, j) to form the pairwise training set in item domain as $D_I = \{(u, i, j) | Y_{ui} = 1 \wedge Y_{uj} = 0\}$, where t is the negative sampling ratio. Then we define the loss function in the item domain as:

$$L_I^p = \sum_u \sum_{(u, i, j) \in D_I} \max(0, s_1(u, i) + \tau_I - s_1(u, j)) \quad (10)$$

τ_I works as the margin to ensure that the distance score between a positive user-item pair after translation is smaller than that between a negative pair, and a larger setting value of τ_I will push the two pairs far away from each other. Similarly, the loss function in the social domain is defined as:

$$L_S^p = \sum_u \sum_{(u, v, w) \in D_S} \max(0, s_2(u, v) + \tau_S - s_2(u, w)) \quad (11)$$

where $D_S = \{(u, v, w) | T_{uv} = 1 \wedge T_{uw} = 0\}$ is the pairwise training set in social domain. Note that D_S is constructed in the same way as D_I and the margin τ_S functions similarly to τ_I .

Regularizations. Motivated by [13], we design two kinds of regularizers for our RML-DGATs model. As we translate each user u to her interacted item i (or user v in her trust list) by the interaction vector r_I (or social vector r_S), we define two distance regularizers to explicitly pull the translated user embedding closer to corresponding item or trustee embedding as:

$$reg_{dist}^I = \sum_u \sum_{i \in C(u)} s_1(u, i), \quad reg_{dist}^S = \sum_u \sum_{v \in N(u)} s_2(u, v) \quad (12)$$

Furthermore, in order to explicitly encode the collaborative information in the user’s (or item’s) neighborhood, we design two neighborhood regularizers to guide users and items to be closer to their neighborhood representations as:

$$reg_{nbr}^I = \sum_u \|p_u - t_u^N\|_2^2 + \sum_i \|q_i - q_i^N\|_2^2, \quad reg_{nbr}^S = \sum_u (\|p_u - s_u^N\|_2^2 + \|p_v - p_v^N\|_2^2) \quad (13)$$

Based on the pairwise loss functions and the two kinds of regularizers, the overall losses in two domains can be formulated as:

$$L_I = L_I^p + \lambda_1 reg_{dist}^I + \lambda_2 reg_{nbr}^I, \quad L_S = L_S^p + \lambda_1 reg_{dist}^S + \lambda_2 reg_{nbr}^S \quad (14)$$

where λ_1 and λ_2 are the regularization coefficients for the distance regularizers and neighborhood regularizers respectively. Finally, we integrate the two relation modeling parts into a joint learning framework and the overall objective function to be minimized is:

$$L = L_I + \gamma L_S \quad (15)$$

Model Training. To optimize the above objective function, we implement the proposed model with Tensorflow and train it with mini-batch Adam [5]. At the end of each mini-batch, all the user and item latent factors are constrained within a unit sphere as $\|v\|_2^2 \leq 1$, to prevent overfitting and ensure the robustness of the model [7].

Table 1. Statistics of the datasets.

	#Users	#Items	#Interactions	#Links	#R.density	#S.density
Ciao	7292	34840	192960	192960	0.0761%	0.4144%
Epinions	21121	92860	644316	353227	0.0329%	0.1584%

4 Experiments

In this section, we conduct experiments on two real-world datasets to evaluate the proposed RML-DGATs model. We first elaborate the experimental settings and then show the overall performance comparison, followed by detailed parameters analysis and cold-start testing. Finally, ablation studies are conducted to further evaluate different parts of the proposed model.

4.1 Experimental Settings

Datasets. In our experiments, we used two publicly accessible datasets: Ciao¹ and Epinions², to evaluate the performance of the proposed model. They both

¹ <http://www.jiliang.xyz/trust.html>.

² <https://alchemy.cs.washington.edu/data/epinions/>.

contain user-item ratings information which scale from 1 to 5 and social relations. As done in many previous works [5, 6], we transformed the observed ratings to 1 indicating the interactions between corresponding user-item pairs, and all the unobserved ratings to 0. Then we filtered out users whose rating records or number of social relations are less than 2, and also removed items rated by less than 2 users. The statistics of the preprocessed datasets are summarized in Table 1, where #Links denotes the number of social relations, #R.density and #S.density denote the density of interactions and social relations respectively.

Baselines. We compared our proposed RML-DGATs model with following eight baselines:

- BPR [14]: A widely used pairwise ranking model tailored for implicit feedback recommendation and it optimizes MF with BPR optimization criterion.
- FISM [9]: An item-based CF model which generates user’s embedding by the items she consumed and learn the item similarity matrix by a low-rank assumption.
- NeuMF [5]: A recently proposed DNN-based model which integrates MF and MLP into a unified model and achieves the state-of-the-art performance.
- CML [7]: By learning a joint metric space which satisfies the triangle inequality, this model can capture users’ finer-grained preferences and show strong performance.
- LRML [15]: It also adopts the idea of translating users to items by translation vectors, while it relies on memory network to learn the relation vectors.
- SBPR [22]: It assumes that users tend to prefer items consumed by her friends than those unobserved and extends the preferences order in BPR to get a finer-grained model.
- CUNE-BPR [21]: It extracts latent semantic friends using random walks and graph embedding and then incorporates them into BPR for ranking task.
- SAMN [1]: A newly proposed DNN-based social recommender which leverages attention-based memory network to model both aspect-level and friend-level differences.

These baselines could be categorized into three groups: the first three belong to CF methods, the subsequent two are built on metric learning and the remainings are social-aware models.

Evaluation Protocols. We adopt the widely used leave-one-out evaluation protocol to evaluate the above methods [9]. Specifically, for each user, we keep her latest two consumed items for validation and testing and the remaining items for training. In line with many previous studies [5, 15], we randomly sample 999 negative items that the user didn’t interact with along with the last interacted one for testing, as it’s too time consuming to rank all unobserved items. In addition, we use three widely used ranking-based metrics HR, MRR and NDCG to measure the recommendation quality [20]. Then we repeated each experiment 5 times and report the average performance.

Parameter Settings. We initialized all the baselines by the parameters declared in the original papers and carefully tuned them to achieve the optimal NDCG@10 on the validation set. Note that we also pretrained the NeuMF model with MF and MLP as the authors did in [5]. For our model, we empirically set the learning rate to 0.001, $d = 128$, the size of attention vectors to 32, $\lambda_1 = \lambda_2 = 0.01$, $t = 4$ on both datasets, and $\tau_I = 0.25$, $\tau_S = 0.5$. Moreover, we set K to 20 and 30, γ to 0.2 and 0.1 on Ciao and Epinions respectively. We also adopted ReLU as the nonlinear activation for all GATs and MLPs after the tuning process.

Table 2. Top-K recommendation performance of different methods on two datasets. The best performance is in boldface and the second is underlined. * denotes the significance p-value < 0.05 compared with the best baseline.

Dataset	Metric	BPR	FISM	NeuMF	CML	LRML	SBPR	CUNE-BPR	SAMN	RML-DGATs	Improv.
Ciao	HR@10	0.2286	0.2076	0.2328	0.2302	0.2471	0.2364	0.2823	<u>0.2964</u> *	0.3108	4.62%
	MRR@10	0.1101	0.0942	0.1196	0.1183	0.1232	0.1165	0.1292	<u>0.1378</u> *	0.1405	1.91%
	NDCG@10	0.1379	0.1207	0.1385	0.1365	0.1473	0.1446	0.1704	<u>0.1732</u> *	0.1803	4.74%
	HR@20	0.3074	0.2847	0.3147	0.3129	0.3184	0.3282	0.3729	<u>0.4028</u> *	0.4157	3.11%
	MRR@20	0.1154	0.0996	0.1219	0.1228	0.1289	0.1213	<u>0.1441</u> *	0.1425	0.1478	2.47%
	NDCG@20	0.1579	0.1403	0.1601	0.1563	0.1703	0.1625	0.1861	<u>0.2009</u> *	0.2066	2.75%
Epinions	HR@10	0.4104	0.3842	0.4017	0.4118	0.4327	0.3749	0.4377	<u>0.4406</u> *	0.4625	4.74%
	MRR@10	0.2123	0.1983	0.2142	0.2240	0.2209	0.1944	<u>0.2295</u> *	0.2284	0.2348	2.24%
	NDCG@10	0.2591	0.2421	0.2489	0.2562	0.2586	0.2436	0.2598	<u>0.2623</u> *	0.2728	3.83%
	HR@20	0.5087	0.4850	0.5023	0.5051	0.5319	0.4868	0.5583	<u>0.5749</u> *	0.6018	4.46%
	MRR@20	0.2191	0.2053	0.2178	0.2241	<u>0.2353</u> *	0.1903	0.2317	0.2334	0.2414	2.54%
	NDCG@20	0.2839	0.2676	0.2881	0.2972	0.3017	0.2653	0.2806	<u>0.2981</u> *	0.3080	3.20%

4.2 Overall Performances

The recommendation performances of different methods are shown in Table 2. We can make the following observations from the table:

- Our proposed model RML-DGATs consistently outperforms all the baselines in all cases, including the two state-of-the-art neural methods NeuMF and SAMN and also the competitive metric learning model LRML. The improvements could be attributed to the finer-grained modeling of each user and the two carefully designed dual GATs in neighborhood interactions.
- Methods incorporating social information generally perform better than those utilizing only user-item interactions in most cases. For example, social recommender CUNE-BPR is superior to non-social models like BPR, FISM and CML. In addition, all with neural architectures, SAMN and our model significantly outperform NeuMF and LRML.
- LRML consistently show superior performances to all other non-social methods as it also adopts the idea of translating users to multiple item-aware representations. However, we empirically found that it underperforms our model when both considering only user-item interactions. This further demonstrates

the effectiveness of our model in modeling the relation vectors by neighborhood interactions.

4.3 Parameters Analysis and Cold-Start Testing

Parameters Analysis. We also conduct experiments to investigate the impacts of the number of sampled neighbors K , the joint learning weight γ and the margin value τ on the two datasets. The results on NDCG@10 are shown in Fig. 3, from which we have the following findings:

- Performances on the two datasets both boost quickly first and then decrease when K varies from 5 to 100. This is reasonable since when neighbors are not enough, the larger K is, the more beneficial information will be incorporated. However, once K exceeds a certain value, the performance will be harmed by the noise introduced by neighbors.
- With the increase of γ , performance on the two datasets also improve first and then decrease. Furthermore, the optimal results on the two dataset are both achieved when γ is relatively small (0.1 and 0.2 respectively). It conforms to the reality since the interaction modeling part in item domain dominates the recommendation task.
- The margin value τ shows similar impact as K and γ . With a relative small τ , users and items would be squeezed tightly in the Euclidean space and results in the geometrical restriction issue as CML [16]. However, if τ exceeds certain value, the similarities among users and items may not retain as they need to meet the distance requirements first.

Fig. 3. Impacts of different parameters.

Fig. 4. Performances on user groups with different levels of sparsity.

Cold-Start Testing. In order to validate whether our proposed RML-DGATs model could mitigate the cold-start issue, we conduct experiments on different sparsity levels of user groups. Firstly, similar to [21], users are categorized into four groups according to the number of interacted items they have in the training set (i.e. <6 , $6-20$, $21-35$, >35) and we term users in the first group as cold-start users. Then we take LRML as a baseline for its competitive performance as a non-social model. Results of the two methods on the two datasets w.r.t. NDCG@10 are shown in Fig. 4 (Results on HR@10 show similar trends). Compared with LRML, RML-DGATs achieves much better performance in all cases. Generally speaking, the sparser the training instances are, the more improvement will it obtain. For example, the relative improvements of our model on the four user groups are 6.32%, 4.15%, 3.76% and 2.43% respectively on Ciao. The rationale behind is that social information could work as the complement to user-item interactions and provide evidence to users’ preferences. For users in the cold-start group which lack sufficient interactions, they will ask for more social relations to make up for the deficiency. In addition, Epinions achieves 3.71% of relative improvement in cold-start scenario, which is obviously smaller than on Ciao, this could be attributed to that interactions and social relations in Epinions are both much sparser than Ciao.

Fig. 5. Performances on different variants.

4.4 Ablation Study

The proposed RML-DGATs leverages two different relation vectors to model the complex relations between user-item in item domain and user-user in social domain. To investigate the effectiveness of the two dual GATs and the metric learning component, we compare our model with the following four variants:

- RML-Avg: This variant replaces the two dual GATs with mean operations, namely averaging neighbors’ latent factors to obtain the corresponding neighborhood-based representations.
- RML-IGATs: The dual GATs in social domain are eliminated and mean aggregations are imposed.
- RML-SGATs: This variant is similar to RML-IGATs except that we remove the dual GATs in item domain instead.
- DGATs: It regards the inner product of two neighborhood-based representations in specific domain as the relation score and incorporates it into the log loss as NeuMF [15].

We report the results of all variants and the proposed model on two datasets w.r.t. HR@10 and NDCG@10 in Fig. 5. As shown in this figure, RML-IGATs and RML-SGATs both show better performances than RML-Avg, which demonstrates the importance of discriminating the contributions of different neighbors when aggregating their features. In addition, RML-SGATs performs better than RML-IGATs on the two metrics. The rationale behind may be that social connections does not necessary guarantee similar preferences while user’s consumed items typically conform to her tastes. Therefore, considering the different contributions of social neighbors will bring more benefits. Furthermore, although all with two dual GATs, DGATs performs poorly than both RML-IGATs and RML-SGATs. This is not surprising since DGATs ignores the constructions of two relation vectors which could translate each user to both multiple item-aware and social-aware representations. Finally, our RML-DGATs consistently and significantly outperforms all the variants, which further verifies the effectiveness of the two carefully designed dual GATs in aggregating neighborhood information and also the metric learning framework for finer-grained user and item modeling.

5 Conclusions

In this paper, we propose a metric learning-based model named RML-DGATs for social recommendation. Specifically, user-item interaction and social relation in corresponding domains are modeled as interaction vector and social vector respectively. Then we model the two relation vectors by neighborhood interactions with two carefully designed dual GATs to fully encode the neighborhood information. Finally, the two domains are jointly trained to mutually enhance each other. Extensive experiments on two real-world datasets demonstrate the superiority of our model, especially in the cold-start scenarios. In the future, we are interested in exploring the dynamic diffusion of user’s interests in social network.

Acknowledgement. This work was supported by the National Key R&D Program of China under grant No.2019YFB2102501.

References

1. Chen, C., Zhang, M., Liu, Y., Ma, S.: Social attentional memory network: modeling aspect-and friend-level differences in recommendation. In: WSDM, pp. 177–185 (2019)
2. Fan, W., Li, Q., et al.: Deep modeling of social relations for recommendation. In: AAAI (2018)
3. Guo, G., Zhang, J., Thalmann, D.: A simple but effective method to incorporate trusted neighbors in recommender systems. In: Masthoff, J., Mobasher, B., Desmarais, M.C., Nkambou, R. (eds.) UMAP 2012. LNCS, vol. 7379, pp. 114–125. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31454-4_10
4. 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)

5. He, X., Liao, L., et al.: Neural collaborative filtering. In: WWW, pp. 173–182 (2017)
6. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: SIGIR, pp. 549–558 (2016)
7. Hsieh, C.K., Yang, L., et al.: Collaborative metric learning. In: WWW, pp. 193–201 (2017)
8. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys, pp. 135–142 (2010)
9. Kabbur, S., Ning, X., Karypis, G.: FISM: factored item similarity models for top-N recommender systems. In: KDD, pp. 659–667 (2013)
10. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: KDD, pp. 426–434 (2008)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(8), 30–37 (2009)
12. Ma, H., et al.: Learning to recommend with social trust ensemble. In: SIGIR, pp. 203–210 (2009)
13. Park, C., et al.: Collaborative translational metric learning. In: ICDM, pp. 367–376. IEEE (2018)
14. Rendle, S., Freudenthaler, C., et al.: BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint [arXiv:1205.2618](https://arxiv.org/abs/1205.2618) (2012)
15. Tay, Y., Anh Tuan, L., et al.: Latent relational metric learning via memory-based attention for collaborative ranking. In: WWW, pp. 729–739 (2018)
16. Veličković, P., et al.: Graph attention networks. arXiv preprint [arXiv:1710.10903](https://arxiv.org/abs/1710.10903) (2017)
17. Wang, J., et al.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: SIGIR, pp. 501–508 (2006)
18. Wang, X., He, X., et al.: Item silk road: recommending items from information domains to social users. In: SIGIR, pp. 185–194 (2017)
19. Xu, G., et al.: Social networking meets recommender systems: survey. *IJSNM* **2**(1), 64–100 (2015)
20. Yu, J., Gao, M., et al.: Adaptive implicit friends identification over heterogeneous network for social recommendation. In: CIKM, pp. 357–366 (2018)
21. Zhang, C., Lu, Y., et al.: Collaborative User Network Embedding for Social Recommender Systems (2017)
22. Zhao, T., McAuley, J., et al.: Leveraging social connections to improve personalized ranking for collaborative filtering. In: CIKM, pp. 261–270 (2014)