Social Recommendation With Characterized Regularization

Chen Gao, Nian Li, Tzu-Heng Lin, Dongsheng Lin, Jun Zhang, Yong Li, *Senior Member, IEEE*, and Depeng Jin, *Member, IEEE*

Abstract—Social recommendation, which utilizes social relations to enhance recommender systems, has been gaining increasing attention recently with the rapid development of online social networks. Existing social recommendation methods are based on the assumption, so-called *social-trust*, that users' preference or decision is influenced by their social-connected friends' purchase behaviors. However, they assume that the influences of social relationships are always the same, which violates the fact that users are likely to share preference on different products with different friends. More precisely, friends' behaviors do not necessarily affect a user's preferences, and the influence is diverse among different items. In this paper, we contribute a new solution, CSR (short for **C**haracterized **S**ocial **R**egularization) model by designing a universal regularization term for modeling variable social influence. This regularization term captures the finely grained similarity of social-connected friends. We further introduce two variants of our model with different optimization manners. Our proposed model can be applied to both explicit and implicit interaction due to its high generality. Extensive experiments on three real-world datasets demonstrate that our CSR can outperform state-of-the-art social recommendation methods. Further experiments show that CSR can improve recommendation performance for those users with sparse social relations or behavioral interactions.

Index Terms—Social recommendation, matrix factorization, adversarial training

1 Introduction

Traditional recommender systems that learn user's preference from only user-item interaction data often suffer from data sparsity and cold start problem, which worsens the recommendation performance [1], [2]. To overcome them, social recommendation [3] utilizes the user's social relations as auxiliary information to help estimate users' preferences better.

Over the past recent years, extensive researches have been working on how to utilize social information to improve recommendation performance [3], [4], [5], [6], [7], [8], [9], [10], [11]. There is a commonly accepted so-called *social-trust* assumption among those researches: users' preference is similar or influenced by their social-connected friends. With this assumption, existing approaches [4], [5], [6], [7] set similarity constraints to users and their friends when estimating interests, and achieve success in improving recommendation performance. Specifically, from the perspective of representation learning, users' preferences and interests are embedded as dense vectors in the latent space, and items' features are the same. When the interaction function, dot product in most models, is fixed, users with close user embeddings in latent space means similar preferences towards items. Based on this, these approaches set a constraint on the distance of each users' embeddings with their friends' embeddings via a regularization term with L1-norm or L2-norm distance. Then this term is combined with the objective function of learning from interactions, introducing social-relational data to the recommendation task.

Recently, some works [8], [12], [13], [14] consider re-building social relations as another task other than estimating user-item interaction. To be specific, these works adopt multi-task learning (MTL) to perform both the task of recommendation and social-link prediction at the same time. However, these models can hardly work in real applications where users' social-relational data is very sparse since abundant data is needed in every sub-task in multi-task learning. Therefore, in this paper, we focus on the old school yet effective methods for the social recommendation, social regularization based models, which are demonstrated to extract sparse social relation to aid recommendation effectively.

Despite the effectiveness of existing social regularization based models, we argue that all the existing social recommendation models suffer from the following three limitations:

- *Strict preference constraint*. In real-world scenarios, sometimes users can make decisions without taking suggestions from friends. In other words, the impact of friends occurs indeed but does not in all scenarios. Therefore, set a universal constraint for all friends is rather strict and not so reasonable.

- *Chen Gao, Nian Li, Dongsheng Lin, Jun Zhang, Yong Li, and Depeng Jin are with Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. E-mail: {gc16, lin17, zhangjun17}@mails.tsinghua.edu.cn, lds354449275@gmail.com, {liyong07, jindp}@tsinghua.edu.cn.*
- *Tzu-Heng Lin is with the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China, and also with AI Innovation Center, Peking University, Beijing 100084, China. E-mail: linzh14@mails.tsinghua.edu.cn.*

Manuscript received 27 June 2019; revised 24 Feb. 2020; accepted 31 July 2020. Date of publication 18 Aug. 2020; date of current version 29 Apr. 2022.
(Corresponding author: Yong Li.)
Recommended for acceptance by K. Shim.
Digital Object Identifier no. 10.1109/TKDE.2020.3017489

- *Fixed similarity for social-connected users.* The social relation between users influences their preference on all items, which violates the fact that a user shares preference on an item with only a part of friends in real life. For example, a user may have similar preferences on books with her schoolmates or workmates, while having similar tastes on food with her family members. This makes the social-trust approaches may bring noises into preference learning, which results in inaccurate recommendation, such as recommending to the user some food that her schoolmates like.
- *Low robustness caused by static regularization.* Existing models rely on regularization to constrain the distance between a user's embedding and her friends directly, which makes the model not robust. Specifically, a slight obfuscation on user embedding will make the distance constraint of the regularization technique does not work.

To address the above mentioned limitations in social recommendation, we propose a new solution named **C**haracterized **S**ocial **R**egularization (CSR). Briefly, we model the item-specific preference similarity of friends to efficiently and accurately learn the social influence on preferences. Specifically, we first design a general item-specific regularization term that can be easily adapted to most existing recommendation models. We then combine this regularization term with the objective function of learning from interaction data. The item-specific regularization term makes the constraint of embedding distance varies in different dimensions. In this way, we can distinguish different similarity with different social-connected friends, which address the limitation of existing methods and make models more effective. Use the previous example, the CSR term helps make sure the user has similar food-taste with her family members.

Moreover, we propose two manners, static and dynamic, to apply the characterized regularization term. To be specific, the static manner directly considers adding the regularization term into the objective function with a weight. The dynamic manner adopts the technique of adversarial training to perform a mini-max game: the adversarial side tries to add noise to worsen the recommendation performance; in contrast, the recommendation side tries to update the latent vector for better recommendation performance.

To summarize, the main contributions of this work are as follows.

- To the best of our knowledge, we are the first to model the characteristics of social relations in the field of social recommendation. We bridge the gap between social connection and preference similarity from the item-specific aspects.
- We introduce a novel CSR method, with two variants, CSR-Joint and CSR-Adv, that effectively models the characteristics of social relations' influence. Existing social recommendation models can be interpreted as the simplest special case of our CSR model.
- Extensive experiments on three real-world datasets demonstrate that our CSR models can outperform the state-of-the-art approaches. Further experiments show that CSR can help to improve the recommendation performance of those users with sparse social relations.

The remainder of the paper is as follows. We rst formalize the problem and introduce some preliminaries in Section 2. We then present our proposed method in Section 3. We conduct experiments in Section 4, before reviewing related work in Section 5 and concluding the paper in Section 6.

2 PRELIMINARIES

We first formulate the task of social recommendation and then introduce the existing social regularization models by highlighting the limitations to motivate our work.

2.1 Problem Formation

In social recommendation, different with traditional recommendation tasks of which only user-item interaction data is available, social relations between users are considered as auxiliary information to help learn users' preferences. The target of social recommendation is the same with traditional recommendation tasks, i.e., estimating the unobserved values of the interaction matrix $\mathbf{Y_{M\times N}} = \{y_{ui}\}$ with M and N denoting the number of users and items. Specifically, for explicit interactions, y_{ui} of user u and item i is a continuous score, such as rating score in movie rating data. For implicit interactions, it is a binary value such as *purchase or not purchase* in E-Commerce behavioral logs. We further use **U** and **V** to denote set of user and item, respectively, and denote the social relations between users as $\mathbf{S} = \{\mathbf{s}_{uu'} | u, u' \in \mathbf{U}\}$.

It is a vector representing the various and diverse relationship between two users u and u', such as friends, following, followed, etc. In existing works, this vector is 1-dimension, which is a constant value 1 standing for the social link or a continuous value between 0 and 1 standing for social strength.

Then, the task of social recommendation is to estimate the unobserved values in the interaction matrix $\mathbf{Y}$ with the help of social relation $\mathbf{S}$. It can be formulated as follows.

Input: The user-item interaction data $\mathbf{Y_{M\times N}}$, and the social relation data $\mathbf{S}$.

Output: A model that estimates the likelihood that a user u will interact with an item i that her has never interacted with before.

After obtaining the predictive model, we can use it to score all items for a user u, and select the top-ranked items as the recommendation results for u.

2.2 Social Regularization

Regularization, a widely used technique in the matrix factorization approach, sets some constraints to learned latent vectors via adding regularization term in the objective function. In most scenarios, this added term introduces some auxiliary information to the task. In social recommendation, the primary auxiliary data is the social relations of users. Therefore, various regularization techniques are developed to set constraints to the learned embeddings of those social-connected users, which is named social regularization.

In brief, social regularization is a regularization term considering and exploiting the input of the task of social recommendation, social relation $\mathbf{S}$. We denote the term of social regularization as $Social(\mathbf{\Omega}, \mathbf{S})$, where $\mathbf{\Omega}$ stands for parameters of the latent model. Now, we discuss two mainstream manners of designing $Social(\mathbf{\Omega}, \mathbf{S})$. In other words, there

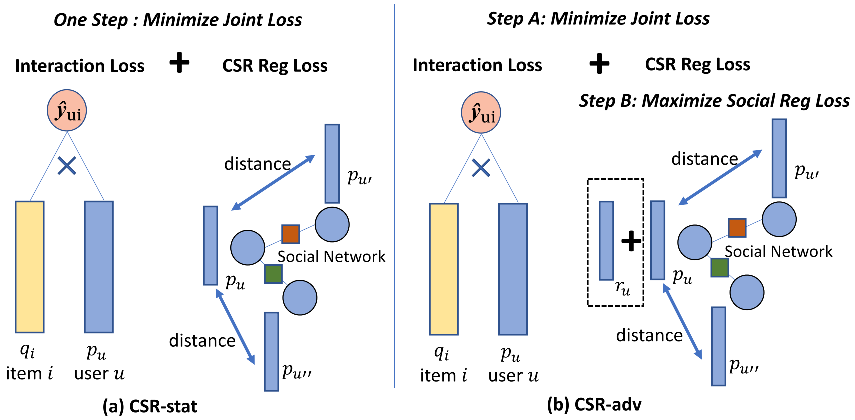


Fig. 1. Two variants of our proposed CSR model.

are two kinds of methods to set constraints for *distance of embeddings* of social-connected users in latent space.

Distance to Weighted Sum of Friends [4], [6]. This method aims to minimize the distance of a user's latent vector with the weighted sum of the connected users' vector as small as possible. To be specific, it first calculates a weighted sum of all friends and then uses the L2 distance between user and the sum as the regularization term. From another perspective, this averaged sum can be considered as the embedding of a *dummy* user. The added social regularization term is expressed as follows:

$$Social(\mathbf{\Omega}, \mathbf{S}) = \sum_{u \in \mathbf{U}} ||\mathbf{p}_u - \sum_{s_{uu'} \in \mathbf{S}} \mathbf{s}_{uu'} \mathbf{p}_{u'}||_2^2, \tag{1}$$

where $s_{uu'}$ is a continuous value between 0 and 1 standing for social strength and $\mathbf{p}$ denotes users' latent vector ($\mathbf{p_u}$ for user u and $\mathbf{p_{u'}}$ for her friend u'). A larger $s_{uu'}$ results in a closer embedding between u and u', which is reasonable and easy to understand. This method is used in [4] for explicit interactions and in [6] for implicit interactions, respectively. The only difference between them is that [6] adopts the pairwise loss [15] to handle implicit data.

Sum of Weighted Distance to Friends [5], [7]. This method aims at minimizing the sum of the weighted distance between the latent vector of a user and her connected users' vector, in which the weight depends on the strength of social relation. To be specific, it first calculates the L2 distance between a user and each of her friend and then sums up distances with weight as the regularization term. The added social regularization term is denoted as follows,

$$Social(\mathbf{\Omega}, \mathbf{S}) = \sum_{u \in \mathbf{U}} \sum_{s_{uu'} \in \mathbf{S}} s_{uu'} ||\mathbf{p}_u - \mathbf{p}_{u'}||_2^2, \tag{2}$$

where symbols are consistent with 1. This method is used in [5] for explicit interactions with pointwise loss and in [7] for implicit interactions with pairwise loss, respectively.

As discussed earlier in the Introduction, all these existing methods assume $\mathbf{s}$ is constant for a given pair (u, u'). In other words, it also exists, and it is item-irrelevant. This ignores the fact that the connection $\mathbf{s}$ may 1) not affect u's purchase decision; 2) vary for purchase decision on different products. Specifically, the preference of u and u' may be very similar for some products, while may be totally different for other products. For example, a user may have similar preferences on books with her schoolmates or workmates, while having similar tastes on food with her family members. As a result, social relations' influences on users' preferences should be variable on different products. Besides, directly add such static regularization term to the objective function makes the model not robust. For static regularization, recommendation performance is relatively sensitive to obfuscation on user embedding. It is due to that a slight obfuscation on user embedding will violate the regularization term and worsen recommendation performance.

To address these shortcomings of existing methods, we take various influences into account by modeling different impacts on different products. We further extend the static solution of social regularization to the dynamic one, improving the robustness of the recommendation model.

3 Methods

Fig. 1 illustrates our proposed CSR model. Following a common setting of MF approaches, our model can both learn from user-item interaction and social relations.

Our CSR method is featured with three parts of designs:

- Extracting CF signal with matrix factorization. Based on matrix factorization, users' interests and items' features are encoded with latent representations.
- Distill social signal with social regularization. We propose a novel regularization method for introducing social influence to estimate user interests.
- Two training manner for combining social regularization into matrix factorization. We propose two manners of fusing matrix factorization and social regularization, which can make sure the model can be easily adapted to real-world datasets.

In what follows, we present our method by elaborating on the above three designs.

3.1 Learning From Interactions

For the task of social recommendation, social relation data serves as the side information to aid recommendation. Therefore, the main target of social recommendation is to learn from interaction data and infer users' interests, which is also reasonable in real-world applications as interaction data is always far denser than social data.

Here we rely on Matrix Factorization (MF), a widely used latent factor model in recommender systems, to extract the predictive signal from interactions for recommendation. Let $\mathbf{P} \in \mathbb{R}^{E \times M}$ and $\mathbf{Q} \in \mathbb{R}^{E \times N}$ be latent user and item features matrices, with column vectors $\mathbf{p}_u$ and $\mathbf{q}_i$ representing E-dimensional user-specific and item-specific latent feature vectors of users u and item i, respectively. Here M and N denote the number of users and items. Since the target of social recommendation is also to estimate the unobserved values in interaction matrix $\mathbf{Y}$, MF model tries to decompose $\mathbf{Y}$ as $\mathbf{Y} = \mathbf{P}^{\mathrm{T}}\mathbf{Q}$. Then the objective function of MF model can be formulated as

$$\begin{aligned}\min_{\mathbf{P},\mathbf{Q}} \mathcal{L}(\mathbf{Y},\mathbf{P},\mathbf{Q}) = &\sum_{u=1}^{M}\sum_{i=1}^{N}(\mathbf{Y}_{ui} - \mathbf{p}_u^T\mathbf{q}_i)^2 \\ &+ \lambda_P||\mathbf{P}|| \\ &+ \lambda_Q||\mathbf{Q}||,\end{aligned} \tag{3}$$

where λ_P and λ_Q are L_2 regularization coefficients, two important hyper-parameters in training procedure, of latent matrices $\mathbf{P}$ and $\mathbf{Q}$, a common technique to solve over-fitting.

Note that in this work to utilize MF to learn from interaction, and our proposed method can be adapted to more complicated interaction functions, such as neural matrix factorization [16].

3.2 Characterized Social Regularization

In social recommendation, users' social connections are leveraged as side information to enhance a recommender system. We exploit such side information with a novel regularization term. The motivation of our CSR model is to use social relations to set the constraint to the parameters during learning from interactions, which is similar with existing methods [4], [5], [6], [7]. Then, a general objective function of MF models considering social regularization can be denoted as

$$\begin{aligned}\min_{\mathbf{P},\mathbf{Q}} \mathcal{L}(\mathbf{Y},\mathbf{P},\mathbf{Q},\mathbf{S}) = &\sum_{u=1}^{M}\sum_{i=1}^{N}(\mathbf{Y}_{ui} - \mathbf{p}_u^T\mathbf{q}_i)^2 \\ &+ \lambda_s Social(\mathbf{P},\mathbf{Q},\mathbf{S}) \\ &+ \lambda_P||\mathbf{P}|| \\ &+ \lambda_Q||\mathbf{Q}||,\end{aligned} \tag{4}$$

where λ_s is a positive hyper-parameter (less than 1 in most researches) controlling the weight of social information compared with interaction data. To be specific, a larger λ_s means social influence plays a more important role in final prediction.

Dimension-Weighted Distance. In latent factor models such as MF, embedding vector for a user encodes his/her latent interest, and embedding vector for an item encodes its features. Due to the social influence, a user's interest is similar to his/her friends, which means the embedding vector of the user is similar to friends' embedding vectors. This motivates the two mainstream social regularization methods introduced in Section 2.2, which utilize euclidean distance to build $Social(\mathbf{P},\mathbf{Q},\mathbf{S})$ to set a constraint on the similarity between friends.

However, from the perspective of representation learning, $\mathbf{q}_{ik}$, the kth dimension in the E-dimensional item latent vector, may represent kth feature of item i; similarly, $\mathbf{p}_{uk}$ may stand for users' preference on the kth item feature. As mentioned in the Introduction, we argue the drawback of social-trust assumption in traditional social regularization is two-fold. First, the social connection does not always cause influence in purchase decisions. Especially for today's online social network, friendship is *online* and can hardly affect users' real-world behaviors. For today's online social networks, some online friendship may be weak and cannot affect users' offline real-world behaviors. Second, a user may share different preferences on various products with different friends. This is apparent as a user's friends are diverse. For example, a user may have similar preferences on books with her schoolmates or workmates, while having similar tastes on food with her family members. Then we can assume two users have a similar preference on an item i is due to some dimensions of item features. In other words, the embedding vector of this item has a larger value in those dimensions. Therefore, we design a dimension-weighted distance rather than a simple euclidean distance to guarantee the similarity of social-connected users. For kth dimension of relationship between user u and one of his/her friends u', we use $s_{uu'}^k$ to denote the kth weight, then we obtain a weight vector: $\mathbf{s}_{uu'}$. Based on this vector, the characterized regularization term can be denoted as

$$Social(\mathbf{P},\mathbf{Q},\mathbf{S}) = \sum_{(u,u')\in\mathbf{S}} ||(\mathbf{p}_u - \mathbf{p}_u') \circ \mathbf{s}_{uu'}||_2^2, \tag{5}$$

where $\circ$ means a Hadamard (element-wise) product.

We can observe that a relatively large weight $s_{uu'}^k$ means user u have a strong similarity in kth feature. For example, if kth dimension stands for appearance of item, and then it means that u and u' both prefer to more nice-looking products.[1]

Product-Sharing Based Social Relations. Theoretically speaking, social relation vector $\mathbf{s_{uu'}}$ can help mode social influence at a fine-grained level. To be more precise, the social influence between two users is not a fixed value representing so-called social strength, and it is replaced with a feature-level similarity between two users. However, it is hard to directly optimize them, since the above objective function will result in a trivial all zero solution for $\mathbf{s_{uu'}}$.

Thus, it is essential to guide the training procedure with more input. In practice, users exchange opinions on various kinds of products, such as movies, news, etc., through sharing with friends. Therefore, to some extent, a sharing behavior or discussion on item i between user u and a friend u' is a social relation accompanied by item i. Thus, we can infer that u and u' share tastes on item i and those items close to i. Since the embedding vector of shared item $\mathbf{q}_i$ represents item features as mentioned before, it can be straightly used as weight vector $\mathbf{s_{uu'}}$. The weight vector can also be built from other data or knowledge of the social domain, indeed.

1. In matrix factorization, the relation between latent vector and user preferences/item attributes is implicit since each dimension of vectors has no exact meaning, and this example explains how our proposed regularization term works in capturing the preference similarity of friends

In this work, we only focus on the general framework and leave these more specific details as future work.

Thus, we formulate our proposed social regularization term as

$$Social(\mathbf{P}, \mathbf{Q}, \mathbf{S}) = \sum_{(u,u',i)\in\mathcal{D}} ||(\mathbf{p}_u - \mathbf{p}'_u) \circ \mathbf{q}_i||_2^2, \tag{6}$$

where $\mathcal{D}$ is the triplets of product-sharing logs, which is utilized to build social relations $\mathbf{S}$, and $(u, u', i) \in \mathcal{D}$ means user u shows similar preference on item i with his/her friend u'.

Note that the characterized social relation is not limited to product-sharing behaviors. For example, in product recommendation of e-commerce websites, co-click behaviors of social-connected users can also serve as the extra input taking the co-clicked items as $\mathbf{S}_{uu'}$. If we have data describing some shared interests of users, then the dimension-weighted distance can be leveraged.

3.3 Fusing Interactions and Social Relations

After we get the models for learning from interactions and social relations, respectively, we need to combine it as social relations serve as side information to help learn from interactions. We propose two manners to fuse learning from interactions and learning from social relations.

Static Characteristic Social Regularization

It is intuitive to utilize a weighted sum to fuse two parts in the objective function. Then the objective function of our CSR model is

$$\begin{aligned}\min_{\mathbf{P},\mathbf{Q}} \mathcal{L}(\mathbf{Y}, \mathbf{P}, \mathbf{Q}, \mathbf{S}) = &\sum_{u=1}^{M}\sum_{i=1}^{N}(\mathbf{Y}_{ui} - \mathbf{p}_u^T\mathbf{q}_i)^2\\ &+ \lambda_s \sum_{(u,u',i)\in\mathcal{D}} ||(\mathbf{p}_u - \mathbf{p}'_u) \circ \mathbf{q}_i||_2^2\\ &+ \lambda_P||\mathbf{P}||\\ &+ \lambda_Q||\mathbf{Q}||,\end{aligned} \tag{7}$$

where two parts of objective function is learned jointly. Such fusing manner makes the social relation affects the learning procedure of interaction statically. We named this variant as CSR-stat (short for static) as the characterized social regularization term keeps static once relevant hyper-parameters are fixed.

Dynamic Characteristic Social Regularization. As we have discussed in the Introduction, traditional static regularization suffers from the drawback of low robustness. To be specific, a small obfuscation on the embedding vectors may affect recommendation performance. It is found by [17] that the traditional MF is vulnerable to perturbations on its parameters, user and item embedding matrices. Note that for latent factor models, generalization ability which determines the performance, is close related to robustness. With adversarial training, not only the robustness but also the generalization is improved, which further brings better recommendation performance. Thus, we propose a novel method for learning with dynamic regularization. To be more precise, we combine two parts as mentioned above with a mini-max game based on adversarial learning, and name this variant as CSR-adv (short for adversarial). The mini-max game is as follows.

- *Maximization*: CSR-adv adds noise to learned embeddings of each user and tries to maximize the CSR term. In other words, the objective of this stage is to maximize the distance between the user and his/her friends. The noise is controlled by a hyper-parameter ϵ, which stands for the noise level. This step can be formulated as follows,

$$\max : \mathbf{r}_u^g = \arg\max_{\mathbf{r}_u, \|\mathbf{r}_u\|\le\epsilon} \sum_{(u,u',i)\in\mathcal{D}} \left\|\left(\mathbf{p}_u + \mathbf{r}_u - \mathbf{p}'_u\right) \circ \mathbf{q}_i\right\|_2^2, \tag{8}$$

  where $\mathbf{r}_u^g$ is the perturbation towards user u's embedding vector controlled by noise level ϵ. It is not easy to obtain the closed form solution of $\mathbf{r}_u^g$. Here we adopt an approximation technique proposed by [18], which is widely used in adversarial learning. To be precise, the objective function around $\mathbf{r}_u$ is approximated to be linear. Then the perturbation on user embeddings can be formulated as follows,

$$\mathbf{r}_u^g \approx \epsilon\frac{\mathbf{g}}{\|\mathbf{g}\|}, \mathbf{g} = \nabla_{\mathbf{r}_u} \sum_{(u,u',i)\in\mathcal{D}} \left\|\left(\mathbf{p}_u + \mathbf{r}_u - \mathbf{p}'_u\right) \circ \mathbf{q}_i\right\|_2^2. \tag{9}$$

  Here $\mathbf{g}$ denotes the gradient with respect to $\mathbf{r}_u$ and other model parameters are fixed during the updating procedure of $\mathbf{r}_u^g$.
- *Minimization*: CSR-adv minimizes the objective function of matrix factorization with a CSR term. This stage is very similar to CSR-stat, and the only difference is the user embedding to optimize has been obfuscated before. It makes learning procedures more robust compared with the traditional manner, CSR-stat, with the help of the perturbed vector. This step can be formulated as follows,

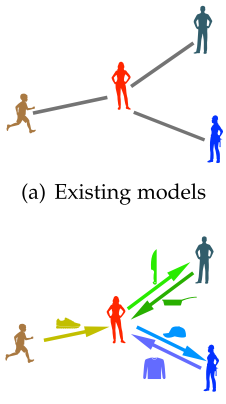
$$\begin{aligned}\min : \Gamma_{CSR-adv} = \Gamma_{MF}&\\ &+ \lambda_s \sum_{(u,u',i)\in\mathcal{D}} \left\|\left(\mathbf{p}_u + \mathbf{r}_u^g - \mathbf{p}'_u\right) \circ \mathbf{q}_i\right\|_2^2,\end{aligned} \tag{10}$$

  where Γ_{MF} represent the loss function in (3).

No matter we choose static or dynamic manner for characterized social regularization, the way obtaining final prediction is the same with basic matrix factorization: adopting inner product as the interaction function. This makes our propose characterized social regularization can be adapted to various recommendation models. Compared with existing methods, our CSR model considers the characteristics of each social relation, which is shown in Fig. 2.

To make it more clear, we introduce the optimization details in Algorithm 1.

In short, the proposed CSR term help solves modeling social-trust effect in a novel manner. The proposed regularization term provides a new perspective for approaching social recommendation. Although the CSR term is concise, it is insightful. Furthermore, the proposed CSR is easy to apply to other tasks and models. For example, the CSR term can also be used in more complicated models, such as



(a) Existing models

(b) Our CSR model

Fig. 2. Comparison of existing models and our CSR model.

NeuMF [16] which use neural networks as the interaction function between user and items.

Algorithm 1. CSR-Stat and CSR-Adv

input: Training data $\mathcal{D}$, social regularizer coeffcient λ_s, learning rate η;
output: Model parameters $\mathbf{p}_u$, $\mathbf{q}_i$.
Initialize the model parameters randomly;
do
  Randomly select training paris from $\mathcal{D}$;
  switch *CSR Variants* **do**
    case *Static*
      $\mathbf{p} \leftarrow \eta \frac{\partial \mathcal{L}}{\partial \mathbf{p}}$ with (7);
      $\mathbf{q} \leftarrow \eta \frac{\partial \mathcal{L}}{\partial \mathbf{q}}$ with (7);
    end
    case *Dynamic*
      $\mathbf{p} \leftarrow \eta \frac{\partial \mathcal{L}}{\partial \mathbf{p}}$ with (10);
      $\mathbf{q} \leftarrow \eta \frac{\partial \mathcal{L}}{\partial \mathbf{q}}$ with (10);
      $\mathbf{r}^g \leftarrow$ (9);
    end
  end
while *not meet stop criteria*;
return learned recommendation model.

3.4 Training

For our proposed CSR models, we adopt mini-batch training and optimize parameters with stochastic gradient descent (SGD), and implement it on Tensorflow,[2] which provides the function of automatic differentiation. Thus, here we omit the derivation of the objective function. We introduce our model in a point-wise manner for explicit interaction data, and for implicit interaction data, our CSR model can also be trained in a pair-wise manner [15]. Therefore, our proposed model can be adapted to these two kinds of interaction data just by switching the training manner.

3.5 Discussions

Here, we summarize some desirable properties of CSR. First, it is obvious that if we set the value of every dimension of $\mathbf{s}$ to be 1 in (5), then our proposed CSR-stat model degenerates to the methods discussed in Section 2.2. Thus, existing social regularization methods can be taken as a particular case of our CSR-stat model. The proposed regularization term provides a new perspective for explaining the latent model in recommendation. Although the CSR term is concise, it is insightful. Second, as our proposed CSR term is general, it can be adapted to different interaction besides inner product in matrix factorization, such as multi-layer perceptron in neural matrix factorization [16]. This means the CSR term can also be used in more complicated models. As a result, our CSR model able to handle more complicated tasks and various interaction data. Last, we provide two manners of unifying the CSR term with learning from interactions, which further implies that our proposed CSR is general.

In this work, we use the product-sharing data for modeling the characteristic social similarity among different friends. This does not mean the CSR can only be applied to such data. In fact, as long as there is data that can describe some shared interests of users, then the design of dimension-weighted distance in our CSR can be leveraged.

4 Experiments

In this section, we conduct extensive experiments on three real-world datasets to answer the following four research questions:

- *RQ1:* How does our proposed CSR model perform as compared with state-of-the-art social recommendation methods?
- *RQ2:* How do the number of social relations affect the recommendation performance of our proposed models? How does the number of users' interaction records affect the recommendation performance of our CSR model?
- *RQ3:* How do the key hyper-parameters affect the performance of our CSR model?

4.1 Experimental Settings

4.1.1 Datasets and Evaluation Protocol

We experiments with three real-world E-commerce datasets that contains both users' purchase logs and users' item-sharing logs.

- *WeChat Dataset* This dataset is collected from the largest E-commerce platform for maternal and infant products in China. Users can both purchase products and share some of them with friends on WeChat,[3] one of the largest social networks in China. This dataset includes user logs from 2017/06/01 to 2017/06/30. It is made up of two parts: purchase record containing users' purchase log with format $\langle user, item, timestamp \rangle$, and sharing record containing users' item-sharing logs with format $\langle user, item, friend, timestamp \rangle$.
- *Beidian Dataset*. This dataset is collected from Beidian,[4] which is a rising platform for social e-commerce [19], [20]. This dataset includes user logs from 2018/09/20 to 2018/10/22, which also contains

2. https://www.tensorflow.org/

3. http://weixin.qq.com
4. http://www.beidian.com

TABLE 1
Statistics of Three Utilized Datasets

Dataset	# Users	# Items	# Purchases	# Social Relations
WeChat	337	553	2,572	476
Beidian	3,773	4,544	39,466	4,679
Beibei	63,576	3,698	413,966	282,468

purchase logs and item-sharing logs that have the same format as WeChat dataset.

- *Beibei Dataset.* This dataset is collected from Beibei,[5] which is another social e-commerce website. Similarly, user can both purchase products or share products with friends.

For our utilized datasets, each item refers to an exactly unique one. In other words, different brand, type, or kind will make two items assigned with different IDs. This is also a widely used manner in e-commerce recommendation [21], [22]. Note that there is seldom an overlapped user or item between these three datasets since they are separate e-commerce websites and platforms, and the target population is completely different. The social connection in our dataset is defined as sharing records in online social network. Once a user shared the item's URL link to his/her friend and the friend clicked that link, the server will add a record containing user IDs and item IDs. Such collected social connections do not reflect the so-called strength explicitly. Therefore, we need to design models that can distill and distinguish social relations' strengths for enhancing recommendation. The available datasets are raw log files collected from the server of the e-commerce website. For evaluation of recommendation models, we conduct a clean and processing, including extracting user-item interaction records and product-sharing records, conducting anonymization on original user IDs, and splitting data to training and testing set.

The statistics of three utilized datasets are summarized in Table 1.

Note that we learn the CSR model based on pairwise loss [15] to adapt to the implicit datasets.

Evaluation Metrics. To evaluate the performance, we adopted the *leave-one-out* evaluation method, used in a lot of works [16], [23], with the following metrics. Specifically, for both datasets, we merged the duplicated user-item interaction and only kept the latest one. Then we use the last interaction of each user as the test item and other interactions as the training set. As our problem falls into the scope of ranking, we adopt two widely used top-K metrics [16], [24] to evaluate the recommendation performance.

- *HR@K: Hit Ratio* (HR) measures whether the test item is contained by the top-K item ranking list (1 for yes and 0 for no).
- *NDCG@K: Normalized Discounted Cumulative Gain* (NDCG) complements HR by assigning higher scores to the hits at higher positions of the ranking list.

Baselines. We compared our CSR models with seven baseline methods suitable for our task, which can be divided into two categories.

5. http://www.beibei.com

The first group contains three methods that only utilize interaction data, i.e., pure CF recommendation scenario.

- *Random (Rand)* This method randomly orders the item candidates, of which the performance is only relevant to the size of candidate sets.
- *ItemPop* This method ranks items base on their popularity, as judged by the number of historical interactions. This is a non-personalized method to benchmark the recommendation performance.
- *BPR [15]*This is the state-of-the-art matrix factorization model for pure CF recommendation. It adopts the pairwise loss to handle the implicit feedback of user-item interaction data. This is a competitive method for recommendation on implicit datasets.

The second group contains three competitive social recommendation methods for implicit interaction data.

- *TBPR [25]* This is a social recommendation method which also consider diversifying social relations. It first defines the weak and strong tie via a heuristic manner and then building several kinds of pairs, according to the defined ties, for Bayesian personalized ranking. This method can be considered as an improved version of [6].
- *SCP [26]* This model first infer some potential friends from data and then combine this predicted implicit social network with the existing social network to enhance recommendation. The modeling of diverse social relations is involved in the generation of implicit social network.
- *UGPMF[7]* This is also a social regularization model. Different from SocialBPR, the term in UGPMF is defined as the sum of the weighted distance to friends. It also adopts a pairwise loss function.
- *SBPR[27]* This method based on Bayesian personalized ranking, and it assumes that users tend to assign higher ranks to items that their friends prefer. Based on this, it builds two kinds of social-aware pairs (positive, negative, but friends prefer) and (negative but friends prefer, negative, and friends do not prefer) for BPR sampling.

Parameter Settings. To make sure our model and all the baselines are optimized well, we conduct a very careful grid search for all hyper-parameters. We search λ_P in [0.001, 0.005, 0.05, 0.01, 0.1], λ_Q in [0.001, 0.005, 0.05, 0.01, 0.1], embedding size E in [10, 16, 32, 64] and λ_S in [0.01, 0.05, 0.1, 0.5]. We search the batch size in [64, 128, 256] for the WeChat dataset and [256, 512, 1024, 2048] for Beidian and Beibei datasets. For CSR-adv, we search the noise level ϵ in [0.01, 0.05, 0.1, 0.5]. We search the learning rate with grid search and find that a range from 0.0001 to 0.001 can achieve good performance for our utilized datasets. And we choose 0.001 as the final choice. We fix the ratio of negative sampling to 1:5, which has been demonstrated to be a reasonable value by a lot of researches [16], [28], [29]. In the following sections, we report the best parameter settings for each model.

4.2 Performance Comparison (RQ1)

We first compare the top-K recommendation performance for our proposed two variants of CSR and all the baselines.

TABLE 2
Top-K Recommendation Performance Comparison on the Utilized Datasets)

Method	Beidian				Wechat				Beibei			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Rand	0.0525	0.0303	0.1023	0.0461	0.0653	0.0382	0.1068	0.0515	0.0587	0.0350	0.1038	0.0494
ItemPop	0.2566	0.1688	0.3798	0.2087	0.2285	0.1518	0.3116	0.1786	0.4613	0.3139	0.6242	0.3665
BPR	0.4487	0.3273	0.5714	0.3658	0.3086	0.2263	0.4065	0.2572	0.7062	0.5493	0.7975	0.5802
SBPR	0.2703	0.1846	0.3907	0.2235	0.2878	0.2060	0.3650	0.2283	0.6521	0.4799	0.7888	0.5248
SCP	0.0482	0.0300	0.0803	0.0403	0.1602	0.1052	0.2255	0.1255	0.0026	0.0017	0.0052	0.0025
UGPMF	-	-	-	-	0.3412	0.2416	0.4214	0.2655	-	-	-	-
TBPR-W	0.4198	0.3059	0.5441	0.3462	0.2878	0.1981	0.4006	0.2350	0.6531	0.4924	0.7656	0.5291
TBPR-S	0.4140	0.3016	0.5441	0.3435	0.2938	0.2060	0.4125	0.2445	0.6705	0.5067	0.7844	0.5438
CSR-stat	**0.4522**	**0.3294**	0.5714	0.3673	0.3472	0.2501	0.4154	0.2625	0.7244	0.5532	**0.8290**	0.5871
CSR-adv	0.4498	0.3283	**0.5735**	**0.3680**	**0.3591**	**0.2564**	**0.4599**	**0.2847**	**0.7281**	**0.5576**	0.8286	**0.5897**

To make the selected K consistent across different datasets, we follow the evaluation manner in [16], [30], which samples 99 negative items for each test item. For the 100-size list, we rank the items according to the predicted values and then use ranking metrics to evaluate it. Table 2 shows the performance of HR@K and NDCG@K for our two CSR methods, three CF methods, and three social recommendation methods. K is set to 5 and 10, following [16]. From the results, we have the following observations:

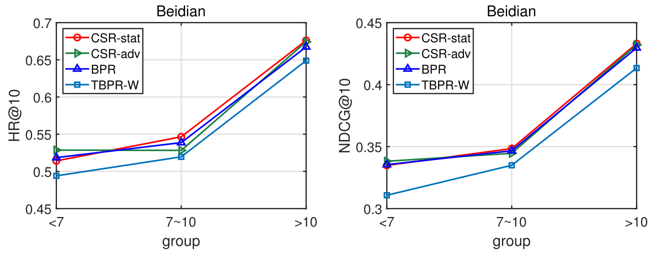
- *Our CSR models achieve the best performance.* Our proposed CSR methods obtain the best performance in terms of HR@K and NDCG@K as compared to all baselines. The one-sample paired t-test indicates that all improvements are statistically significant for $p < 0.05$. Within two variants of our proposed CSR, CSR-adv achieves better on the whole. For the Beidian dataset, our proposed CSR can outperform the strongest baseline from 0.31 to 0.78 percent; for the WeChat dataset, our proposed CSR can outperform the strongest baseline by 10.69 to 16.36 percent ; for the Beibei dataset, our proposed CSR can outperform the strongest baseline by 1.51 to 3.95 percent. Note that the improvement on the Beidian dataset is not so large, and further experiments in Section 4.3 demonstrate the improvement mainly comes from sparse users, which verifies that our CSR still has high application value for the Beidian dataset. To better present the improvement, we observe some actual examples by studying the rank of test items for different models. For the Beidian dataset, for a 100-size list, if the strongest baseline can succeed to rank the test item to the top-10 positions, then our CSR-adv can further improve the rankings by 2.30 positions on average. For the WeChat dataset, the value is about 4.69. This example shows the improvement in actual rankings is significant.
- *Dynamic regularization is more stable than static regularization for our proposed CSR.* On most of the evaluation metrics for our utilized datasets, CSR-adv achieves better or very similar recommendation performance compared with CSR-stat. Especially for the WeChat dataset, which is far more sparse than Beidian, the improvement compared with CSR-stat is significant. This demonstrates that the improved robustness via adversarial learning can also improve the generalization ability, especially for sparse datasets.
- *Rand and ItemPop achieve the worst performance.* Two non-personalized recommendation methods, Rand and ItemPop, achieve bad performance on all metrics for utilized three datasets. It indicates the power of the MF model in modeling personalized behaviors and demonstrates that it is essential to design personalized recommendation models. For the SCP model, we have tried our best to tune the hyperparameters, but it still performs badly. Its poor performance can be explained that it is not suitable for the implicit data form in our implicit dataset since it is originally proposed for rating datasets.
- *Social-aware baselines do not always achieve better performance than BPR.* For UGPMF, due to its low optimization efficiency caused by the sum of L2 distance, we failed to optimize it into convergence in two relatively larger datasets, Beibei and Beidian. We can observe that BPR, which cannot leverage social-relation data, is a very strong baseline. Another competitive social recommendation model, TBPR, achieve similar or even worse performance compared with BPR. This can be explained that there are some biases in our social-relation data. Such biases may make those social recommendation models fail to leverage social information.

  To summarize, the extensive comparison of three real-world datasets verifies that our proposed CSR solution can effectively leverage social signals in item-sharing behaviors to improve the recommendation performance.

This demonstrates that our proposed CSR can help to alleviate the data sparsity issue of interaction data.

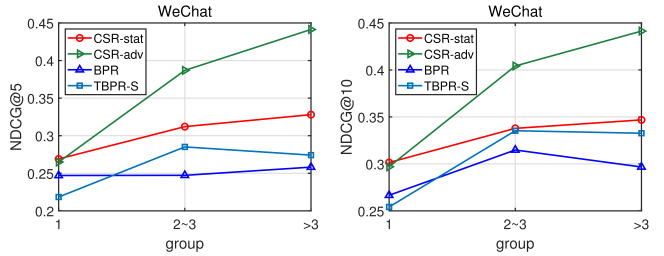
4.3 Impact of Number of Interaction Records and Social Relations (RQ2)

One objective of social recommendation is leveraging social information to alleviate data sparsity. In other words, for users with sparse interactions, traditional methods which can only exploit interaction data cannot estimate users' interests well. Thus, it is meaningful to study whether our proposed CSR methods can help alleviate the data sparsity issue. We divide users into several groups according to the number of



(a) HR@10 (b) NDCG@10

Fig. 3. Top-K recommendation performance of users with different number of interactions.



(a) NDCG@5 (b) NDCG@10

Fig. 4. Top-K recommendation performance of users with different number of social relations.

interactions and we make sure each group has enough users to eliminate randomness. For these three groups, the numbers of users are 1591,1151 and 1,031. Then we report the average top-K recommendation performance of each group. We present the recommendation performance on Beidian dataset in Fig. 3. We can observe that the CSR-adv's performance improvement mainly comes from sparser users. For users with enough records, only learning from user-item interaction records can achieve good performance.

It is also meaningful to study how our model outperforms in making use of social relations. We divide users into three groups according to the number of sharing records: [1, 2-3, >3]. This makes sure that the number of users in each group is close. The recommendation performances for each group of our CSR models and existing methods of social recommendation are shown in Fig. 4. Here we also present the performance of BPR and TBPR-S, two of the best baselines. First, we can observe that the recommendation performance of users with more than three social relations is much better than users with sparse social relations, which demonstrates the importance of social information. With abundant social relations, all social-aware methods, including our CSR and TBPR-S, can perform better than BPR. Second, although all models achieve good performance for users with abundant social relations, our CSR-adv model always performs good for users with sparse social relations.

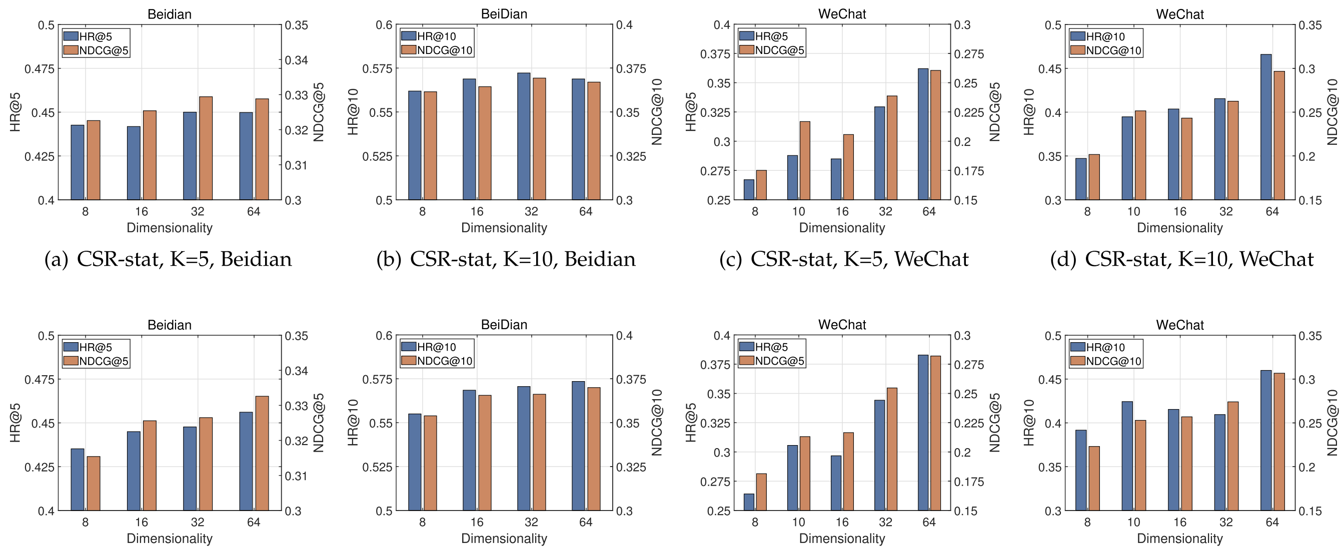
4.4 Parameter Study of CSR (RQ3)

For our proposed two variants of CSR, there are some key hyper-parameters. Here we try to study how these hyper-parameters affect the top-K recommendation performance.

Impact of Embedding Size E. For both two methods, the embedding size E is a significant hyper-parameter. To study the effect of E, we conduct a careful grid search to report the best performance for different E on two utilized datasets. We present the impact of E in Fig. 5. We find that a relatively larger embedding size can bring better performance. This can be explained that model capacity increases with more parameters. For each dataset, the searched ranges of embedding sizes are the same for our models and all baseline models, which is fair for comparison.

Impact of the Noise Level in Adversarial Training. For CSR-adv that adopts adversarial training for introducing social regularization, there are two key hyper-parameters relevant to the training procedure, noise level, and adversarial learning rate. To study how these two affect the recommendation performance, we compared the HR@K and NDCG@K and presented the results in Table 3. We can observe that a suitable noise level ϵ can achieve the best performance on all metrics, which implies that our proposed CSR-adv is easy to apply in real-world applications.

Impact of Batch Size. Batch size plays an important role in SGD-based optimization. As mentioned before, we conduct



(a) CSR-stat, K=5, Beidian (b) CSR-stat, K=10, Beidian (c) CSR-stat, K=5, WeChat (d) CSR-stat, K=10, WeChat

(e) CSR-adv, K=5, Beidian (f) CSR-adv, K=10, Beidian (g) CSR-adv, K=5, WeChat (h) CSR-adv, K=10, WeChat

Fig. 5. Top-K recommendation performance of our proposed CSR with different embedding size E.

TABLE 3
Top-K Recommendation Performance of CSR-Adv With Different Noise Level ϵ on Beidian Dataset

Dataset	Noise Level ϵ	HR@5	NDCG@5	HR@10	NDCG@10
WeChat	0.05	0.3472	0.2458	0.4243	0.2686
	0.01	0.3412	0.2448	0.4243	0.2727
	0.005	0.3442	0.2547	0.4095	0.2740
	0.001	0.3591	0.2564	0.4599	0.2847
Beidian	0.05	0.4495	0.3270	0.5709	0.3664
	0.01	0.4498	0.3283	0.5735	0.3680
	0.005	0.4503	0.3270	0.5722	0.3659
	0.001	0.4479	0.3277	0.5661	0.3652

TABLE 4
Top-K Recommendation Performance With Different Batch Size on Utilized Datasets

Model		CSR-Adv		CSR-Stat	
Dataset	Batch Size	HR@5	NDCG@5	HR@5	NDCG@5
WeChat	32	0.3442	0.2547	0.3294	0.2387
	128	0.3145	0.2357	0.3264	0.2380
	256	0.3056	0.2248	0.3294	0.2260
Beidian	256	0.4495	0.3270	0.4426	0.3226
	512	0.4498	0.3283	0.4418	0.3254
	1024	0.4503	0.3270	0.4500	0.3294
	2048	0.4479	0.3277	0.4498	0.3288

a sufficient grid search for the choices of batch size. Here we present the best performances under different batch size settings in Table 4. We can observe that, in general, the performance of CSR variants is not so sensitive to the choice of batch size.

Impact of Optimizer and its Learning Rate. In our experiments, with a careful grid search, we have found that for all three utilized datasets, a learning rate in the range from 0.0001 to 0.001 can optimize the model well. Thus we choose 0.001 as it can bring faster convergence.

5 Related Work

5.1 Social Recommendation

Social recommendation aims to leverage social relation data to improve recommender systems. A most widely accepted paradigm for social recommendation is the assumption of social-trust. That is, users are assumed to have similar interests with all social-connected users. Based on this, there are a lot of researches [4], [5], [6], [7] proposed social-trust models to make sure learned user interests are similar with friends as much as possible. Ma *et al.* [5] proposed a regularization term to minimize the distance of a user's latent vector with the weighted sum of the connected users' vector. This method is adapted to implicit feedback data by [7]. Jamali *et al.* [4] proposed a similar regularization term that tries to minimize the sum of the weighted distance between the latent vector of a user and her connected users' vector, in which the weight depends on the social strength. Guo *et al.* [6] further extended this regularization term to the recommender system for implicit feedback via pairwise learning.

There are other works [31], [32], [33] further study high-order social relation for social recommendation. To be specific, these works consider some implicit relations based on explicit social connection and user-item interaction. Then these implicit relation is utilized for enhancing recommendation. Gulati *et al.* [31] proposed to define some explicit relationships to represent high-order social relations between users based on traditional social-relation data. Then the similarity metric on the relationship serves as weights at the regularization term. With the development of graph convolutional neural networks (GCN) [34], high-order connectivity can be extracted implicitly with neural graph convolutional layer. Fan *et al.* [32] and Wu *et al.* [33] propose GCN-based models respectively to distill high-order social-connection signal to aid collaborative recommendation.

There are some works [8], [12], [13], [14] exploit social-trust effect from another perspective. Specifically, these works regard the social-relation matrix similarly with the user-item matrix and apply collective matrix factorization to decompose two matrices jointly. Ma *et al.* [8] assumed the user latent vector can be shared across the rating matrix and social-relation matrix. Rafailidis *et al.* [13] extended it with listwise loss function. Recently, Liu *et al.* [14] proposed an essential preference space to capture the differences between user preferences in recommender systems and social networks. To be specific, these two kinds of user embeddings, encoding user preferences in two domains, are mapped from essential preference space with different mapping functions.

Social effect also plays a significant role in other recommendation tasks, such as social-aware point-of-interest recommendation [35], [36], [37], [38], [39], cross-domain recommendation [40], sequential recommendation [41], etc. In this work, we only concentrate on the basic setting of social-aware recommendation.

In this paper, we focus on the most popular social recommendation method, social regularization.

5.2 Adversarial Recommendation

Adversarial machine learning has achieved great success in various tasks [18], [42], [43], [44], [45] with its high robustness and generalization ability. Recently, some researches applied adversarial machine learning methods for performing recommendation tasks, which can be categorized into two groups. On the one hand, some researches [46], [47], [48] utilized the GAN framework [18] and designed generative and discriminative models for recommendation. Wang *et al.* [46] utilized an MF as a generative model to select unobserved items to fool the discriminator. Chae *et al.* [47] adopted a neural network as the generative to generate user's historical interactions. Ding *et al.* [48] further considered the real false-negative samples with the help of exposure data in recommender systems and utilized neural networks to generate those samples. Recently, Fan *et al.* [49] applied the GAN network to social recommendation to generate fake user-user social-connected pairs, which still is based on the social-trust assumption.

On the other hand, there are some works [17], [50] adopting adversarial training [51], [52] for learning more robust feature during the training procedure. He *et al.* [17] first introduced adversarial training into recommender systems. To be specific, the authors propose to add a stage of feature obfuscation in the training stage of Bayesian personalized ranking, which is a competitive matrix factorization model for implicit recommender systems. Tang *et al*. [50] extended it to the task of fashion recommendation via adding noise to learned embedding of images during the training procedure. It is also worth-mentioning the denoising autoencoder (DAE) based models [53], [54] which extend traditional autoencoder (AE) [55] also aims to robust recommendation. Different from adversarial training based models mentioned above, DAE adds noise to the input data, user-item interaction, to improve the robustness of the model.

In general, the core idea of the first category, GAN-based recommendation models, is to utilize the generator-discriminator framework to better sample hard negative samples. For the second category, adversarial training based models, the main motivation is to improve robustness and generalization ability, which can further improve recommendation performance.

In this paper, we adopt adversarial training to improve the robustness of our proposed CSR model and further provide more accurate recommendation.

6 Conclusion and Future Work

This work contributes to a new solution for social recommendation. We argue the traditional assumption of social trust is not reasons from multiple perspectives. We develop a novel design of a regularization term for leveraging social relations in collaborative filtering. Two variants of models are proposed, and extensive experiments demonstrate both of them can improve the recommendation performance significantly. Further studies show that our proposed models can alleviate the data sparsity issue in two aspects, including sparse social relations and sparse interactions.

It is worth mentioning that this work modeling user preference based on user-item interaction and characteristic social effect. In fact, recommendation in real-world scenarios may be much more complicated, with diverse and multiple-source data input such as geographical knowledge, item attributes, temporal information, etc. This work emphasizes studying the characteristic social effect for social recommendation, leaving these extra data inputs unexplored.

In the future, we plan to extend our proposed CSR to more complicated models. First, we are interested in applying CSR in more complex interaction function, such as factorization machine [56] or neural matrix factorization [16]. Second, we will combine CSR with high-order social connectivity. For example, we can apply CSR in the training procedure of graph convolutional networks.

Acknowledgments

This work was supported in part by The National Key Research and Development Program of China under grant SQ2020AAA010130, the National Nature Science Foundation of China under U1936217, 61971267, 61972223, 61941117, 61861136003, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology. An early conference version of this work was published in ACM CIKM 2018 [57].

References

[1] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 116–142, 2004.

[2] A. Popescul, D. M. Pennock, and S. Lawrence, "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments," in *Proc. 17th Conf. Uncertainty Artif. Intell.*, 2001, pp. 437–444.

[3] J. Tang, X. Hu, and H. Liu, "Social recommendation: A review," *Soc. Netw. Anal. Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.

[4] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. ACM Recommender Syst. Conf.*, 2010, pp. 135–142.

[5] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.

[6] L. Guo, J. Ma, H.-R. Jiang, Z.-M. Chen, and C.-M. Xing, "Social trust aware item recommendation for implicit feedback," *J. Comput. Sci. Technol.*, vol. 30, no. 5, pp. 1039–1053, 2015.

[7] L. Du, X. Li, and Y.-D. Shen, "User graph regularized pairwise matrix factorization for item recommendation," in *Proc. Int. Conf. Adv. Data Mining Appl.*, 2011, pp. 372–385.

[8] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2008, pp. 931–940.

[9] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 203–210.

[10] J. Tang, H. Gao, and H. Liu, "mTrust: Discerning multi-faceted trust in a connected world," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2012, pp. 93–102.

[11] J. Tang, H. Gao, X. Hu, and H. Liu, "Exploiting homophily effect for trust prediction," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2013, pp. 53–62.

[12] J. Tang, X. Hu, H. Gao, and H. Liu, "Exploiting local and global social context for recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2712–2718.

[13] D. Rafailidis and F. Crestani, "Joint collaborative ranking with social relationships in top-N recommendation," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1393–1402.

[14] C.-Y. Liu, C. Zhou, J. Wu, Y. Hu, and L. Guo, "Social recommendation with an essential preference space," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 346–353.

[15] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme , "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[16] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[17] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 355–364.

[18] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[19] T.-H. Lin, C. Gao, and Y. Li, "CROSS: Cross-platform recommendation for social E-commerce," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 515–524.

[20] F. Xu, Z. Han, J. Piao, and Y. Li, "" i think you'll like it" modelling the online purchase behavior in social E-commerce," *Proc. ACM Hum.-Comput. Interaction*, 2019, vol. 3, pp. 1–23.

[21] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[22] K. Wei, J. Huang, and S. Fu, "A survey of E-commerce recommender systems," in *Proc. Int. Conf. Serv. Syst. Serv. Manage.*, 2007, pp. 1–5.

[23] M. Jamali and M. Ester, "Using a trust network to improve top-N recommendation," in *Proc. 3rd ACM Conf. Recommender Syst.*, 2009, pp. 181–188.

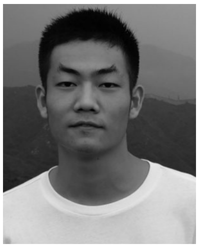
[24] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "NAIS: Neural attentive item similarity model for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2354–2366, Dec. 2018.

[25] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 5–14.

[26] C.-C. Hsu, M.-Y. Yeh, and S.-D Lin, "A general framework for implicit and explicit social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2228–2241, Dec. 2018.

[27] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 261–270.
[28] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong, "Attentive group recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 645–654.
[29] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-N recommendation," *ACM Trans. Inf. Syst.*, vol. 37, no. 3, 2019, Art. no. 33.
[30] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 566–577.
[31] A. Gulati and M. Eirinaki, "With a little help from my friends (and their friends): Influence neighborhoods for social recommendations," in *Proc. World Wide Web Conf.*, 2019, pp. 2778–2784.
[32] W. Fan *et al.*, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.
[33] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 235–244.
[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–14.
[35] J. J.-C. Ying, W.-N. Kuo, V. S. Tseng, and E. H.-C. Lu, "Mining user check-in behavior with a random walk for urban point-of-interest recommendations," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 1–26, 2014.
[36] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *Proc. 20th Int. Conf. Advances Geographic Inf. Syst.*, 2012, pp. 199–208.
[37] B. Berjani and T. Strufe, "A recommendation system for spots in location-based online social networks," in *Proc. 4th Workshop Soc. Netw. Syst.*, 2011, pp. 1–6.
[38] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks," in *Proc. 18th SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2010, pp. 458–461.
[39] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 325–334.
[40] T. Man, H. Shen, X. Jin, and X. Cheng, "Cross-domain recommendation: An embedding and mapping approach," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2464–2470.
[41] C. Cai, R. He, and J. McAuley , "SPMC: Socially-aware personalized Markov chains for sparse sequential recommendation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1476–1482.
[42] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–11.
[43] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–11.
[44] S.-M. Moosavi-Dezfooli , A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1765–1773.
[45] Y. Wu, D. Bamman, and S. Russell, "Adversarial training for relation extraction," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 1778–1783.
[46] J. Wang *et al.*, "IRGAN: A minimax game for unifying generative and discriminative information retrieval models," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 515–524.
[47] D.-K. Chae, J.-S. Kang, S.-W. Kim, and J.-T. Lee, "CFGAN: A generic collaborative filtering framework based on generative adversarial networks," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 137–146.
[48] J. Ding, Y. Quan, X. He, Y. Li, and D. Jin, "Reinforced negative sampling for recommendation with exposure data," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2230–2236.
[49] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, and Q. Li, "Deep adversarial social recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1351–1357.
[50] J. Tang, X. Du, X. He, F. Yuan, Q. Tian, and T.-S. Chua, "Adversarial training towards robust multimedia recommender system," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 5, pp. 855–867, May 2020.
[51] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *The J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.
[52] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2107–2116.
[53] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, 2015, pp. 811–820.
[54] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-N recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining*, 2016, pp. 153–162.
[55] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 111–112.
[56] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.
[57] T.-H. Lin, C. Gao, and Y. Li, "Recommender systems with characterized social regularization," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, 2018, pp. 1767–1770.

Chen Gao received the BS degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2016. He is currently working toward the PhD degree in the Department of Electronic Engineering, Tsinghua University, Beijing, China. His research interests include recommender systems, data mining and data privacy. He has publications appeared in several top journals and conferences such as the *IEEE Transactions on Knowledge and Data Engineering*, WWW, ICDE, KDD, SIGIR, IJCAI and UbiComp. His work on graph neural network based recommender system received the Best Short Paper Honorable Mention Award in SIGIR 2020.

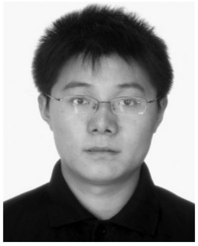


Nian Li is with the Department of Electrical Engineering, Tsinghua University, Beijing, China. His researches focus on recommender systems and deep learning.

Tzu-Heng Lin received the BS degree in electrical engineering from Tsinghua University, Beijing, China, in 2018. He is currently working toward the MS degree in Artificial Intelligence Innovation Center, Peking University, China. His researches focus on recommender systems and deep learning.

Dongsheng Li is currently working toward the undergraduate degree in Electronic Engineering Department, Tsinghua University, Beijing, China. His works mainly focus on recommender systems.

Jun Zhang is with the Department of Electrical Engineering, Tsinghua University, Beijing, China. His researches focus on recommender systems and deep learning.



Yong Li (Senior Member, IEEE) received the BS degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007 and the PhD degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a faculty member of the Department of Electronic Engineering, Tsinghua University, China. He has served as general chair, TPC chair, TPC member for several international workshops and conferences, and he is on the editorial board of two IEEE journals. His papers have total citations more than 5200. Among them, ten are ESI Highly Cited Papers in Computer Science, and four receive conference best paper (run-up) awards. He received IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers and Young Talent Program of China Association for Science and Technology.

Depeng Jin (Member, IEEE) received the BS and PhD degrees from Tsinghua University, Beijing, China, in 1995 and 1999 respectively both in electronics engineering. Currently he is an associate professor at Tsinghua University, China and vice chair of the Department of Electronic Engineering. He was awarded National Scientific and Technological Innovation Prize (Second Class), in 2002. His research fields include telecommunications, high-speed networks, ASIC design, and future internet architecture.