symmetry

MDPI

Article

A Social Recommendation Based on Metric Learning and Users' Co-Occurrence Pattern

Xin Zhang [1,2], Jiwei Qin [1,2,*] and Jiong Zheng [1]

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; zhangxin_996@stu.xju.edu.cn (X.Z.); zhengjiong@xju.edu.cn (J.Z.)

[2] Key Laboratory of Signal Detection and Processing, Xinjiang Uygur Autonomous Region, Xinjiang University, Urumqi 830046, China

* Correspondence: jwqin@xju.edu.cn

Abstract: For personalized recommender systems, matrix factorization and its variants have become mainstream in collaborative filtering. However, the dot product in matrix factorization does not satisfy the triangle inequality and therefore fails to capture fine-grained information. Metric learning-based models have been shown to be better at capturing fine-grained information than matrix factorization. Nevertheless, most of these models only focus on rating data and social information, which are not sufficient for dealing with the challenges of data sparsity. In this paper, we propose a metric learning-based social recommendation model called SRMC. SRMC exploits users' co-occurrence patterns to discover their potentially similar or dissimilar users with symmetric relationships and change their relative positions to achieve better recommendations. Experiments on three public datasets show that our model is more effective than the compared models.

Keywords: recommender systems; social recommendation; metric learning

check for updates

Citation: Zhang, X.; Qin, J.; Zheng, J. A Social Recommendation Based on Metric Learning and Users' Co-Occurrence Pattern. *Symmetry* **2021**, *13*, 2158. https://doi.org/10.3390/sym13112158

Academic Editor: José Carlos R. Alcantud

Received: 28 September 2021
Accepted: 3 November 2021
Published: 11 November 2021

1. Introduction

With the rapid development of the internet, information overload [1] has become a common problem. To help users find truly valuable information better and faster, recommender systems have been widely applied in recent decades. Traditional recommender systems are mainly divided into two categories: content-based recommendation and collaborative filtering recommendation [2]. In collaborative filtering-based recommendation models, matrix factorization (MF) plays an important role due to its efficiency and scalability. In MF, each user or item is represented by a user or item latent vector, and the dot product between them is used to capture known ratings and predict unknown ratings. Since the dot product does not satisfy the triangle inequality, the MF model cannot reliably capture the item–item or user–user similarity, nor can it capture the fine-grained preferences present in user feedback, as Hsieh has proved [3].

Metric learning-based models produce distance functions that both satisfy the triangle inequality and capture important relationships among data. They have been widely used in various tasks for classification and clustering [4]. Accordingly, some works [3,5,6] utilize metric learning to overcome the disadvantages of the MF models. These works project users and items into a low-dimensional metric space, where user preferences are measured by the distance between the user and the item. Specifically, Hsieh's CML [3] minimizes the distance between the users and their positively rated items, making them closer to their preferred items. Tay's LRML [6] incorporates a memory network to further learn relations between users and items in metric space. Yu's SocialFD [5] changed users' spatial location, bringing users closer to their positively rated items and trusted friends.

Existing metric learning-based models have achieved satisfactory results, but these models still face the challenge of data sparsity. To alleviate this problem, they have introduced social information and achieved a certain degree of success [7]. However, all of these models overlook an important issue: social information is often as sparse as rating data, and most users' social information is still very sparse. An exploratory work in matrix factorization models is Liang's Cofactor [8], which jointly decomposes the user–item rating matrix and the item–item co-occurrence matrix with shared item latent factors. The Cofactor considers that the items that users often consume in tandem have some similarities, effectively enhancing the recommendation model. Tran's RME [9] further extends the Cofactor with the co-occurrence patterns of users and successfully proves the effectiveness of the co-occurrence pattern for the matrix factorization model.

Although the co-occurrence pattern of users or items effectively improves the matrix factorization model, the way to utilize this co-occurrence pattern information in a metric learning-based model is still a problem that needs to be solved. Refs. [10,11] inspired our work and we propose a metric learning-based social recommendation model called SRMC. SRMC exploits information about the users' co-occurrence patterns to discover users with symmetric relationships, whose consumption behavior is extremely similar or dissimilar and changes their relative positions in the metric space to achieve better recommendations. Our main contributions are shown as follows:

1. We propose a metric learning-based social recommendation model (SRMC), which provides a new idea of how to exploit users' co-occurrence pattern information in a metric learning-based model.
2. We provide an idea of how to exploit the user's co-occurrence pattern information to discover their potentially similar or dissimilar users with symmetric relationships.
3. We conducted extensive experiments on three datasets to demonstrate the superiority of SRMC over comparative algorithms for rating prediction tasks.

2. Related Work

2.1. Social Recommender System

Traditional recommender systems have been facing the problem of data sparsity. With the development of social network platforms, social recommender systems have emerged and effectively alleviated this problem. Social recommender systems assume that users are influenced by users with social relationships, resulting in some similarity in their preferences [12]. Specifically, if a user interacts with only a few items, we can infer his preferences based on his friends' interactions and then generate better recommendations. Early explorations of this idea focused on matrix factorization (MF) and achieved satisfying results. Ma's SoRec [13] model utilized users' social networks to alleviate the data sparsity problem and improve the recommendation effect. Mohsen [14] introduced the trust propagation principle into the matrix factorization model. Guo exploited the implicit information and proposed TrustSVD [15]. Zhao introduced social information into the Bayesian personalized ranking algorithm and proposed SBPR [16]. However, the above methods only utilize the sparse social network and ignore the additional auxiliary information hidden in the rating data, limiting the recommendation effectiveness.

2.2. Metric Learning in Recommender System

The goal of metric learning is to learn a suitable distance metric under the condition of a given set of constraints to ensure that the distribution of similar samples is more compact and the distribution of different samples is more spread out [17]. There are many distance functions that can be utilized, such as Euclidean distance and Mahalanobis distance. The Mahalanobis distance between any two points x_i and x_j can be expressed by:

$$d_A(x_i, x_j) = \| x_i - x_j \|_A = \sqrt{(x_i - x_j) A (x_i - x_j)^T} \quad (1)$$

In Equation (1), $A \epsilon R^{m \times m}$ has to be a positive semidefinite matrix to keep the distance non-negative and symmetric. The global optimization problem with constraints can be stated as:

$$\min_{A \epsilon R^{m \times m}} \sum_{(x_i, x_j) \epsilon S} d_A^2(x_i, x_j)$$

$$s.t. \sum_{(x_i, x_j) \epsilon D} d_A^2(x_i, x_j) \geq \theta, A \geq 0 \tag{2}$$

where S denotes the set of equivalent constraints in which x_i and x_j belong to the same class, and D denotes the set of inequivalent constraints in which x_i and x_j belong to different classes. θ is the minimum distance between two different classes of data points.

Heish's proposed CML closes the spatial distance between users and their positively rated item by minimizing the loss function, and pushes the spatial distance between users and other items. In this process, users who share a common liking for the same item also gather together. Tay's proposed LRML learns the latent relations of each user–item pair instead of a simple push-pull mechanism, mitigating the potential geometric inflexibility of existing metric learning models. Yu's SocialFD introduces social networks to change the spatial location of users and items, which alleviates the data sparsity problem to some extent. Although all these works have achieved satisfactory results, none of them has considered how to exploit the co-occurrence patterns of users in a metric-based learning model. It has been proven that co-occurrence patterns of users or items can effectively alleviate the data sparsity problem in the matrix factorization-based models [9].

Compared to the dot product in matrix factorization, the metric learning-based model reflects the user's preferences more accurately. For example, let user u be at u = (1, 1) in the matrix factorization model and let user v be at v = (1, 1) in the metric learning-based model. Next, we are going to recommend the most suitable items for both users. Obviously, for user v, we only need to recommend the closest item in the metric space, i.e., the closest space point to v = (1, 1). However, we cannot recommend the most suitable item for user u, but it is certain that q = (1, 1) will be worse than q = (2, 2) to recommend for user u.

2.3. Co-Occurrence Pattern in Matrix Factorization

It is well known that Word2vec [18] has achieved substantial success in NLP tasks. Essentially, Word2vec uses word vectors to represent semantic information in a massive corpus, making similar words closer in the word vector space. Word2vec has two main models: Skip-Gram to predict context words from input words and CBOW to predict input words from given context words. In the Skip-Gram model, Pointwise mutual information (PMI) values measure the association between a word w and its context word c by calculating the log of the ratio between their joint probabilities $P(w, c)$ and their marginal probabilities $P(w)\ and\ P(c)$. The formula of PMI is shown as:

$$\mathrm{PMI}(w, c) = \log \frac{P(w, c)}{P(w)P(c)} \tag{3}$$

It can also be written as:

$$\mathrm{PMI}(w, c) = \log \frac{\#(w, c) \cdot |D|}{\#(w)\#(c)} \tag{4}$$

Here $\#(i, j)$ is the number of times word j appears in the context of word i. $\#(w) = \sum_c \#(w, c)$ and $\#(c) = \sum_w \#(w, c)$. $|D|$ is the total number of word-context pairs.

Liang's Cofactor suggested that items that were frequently consumed by users in tandem also have some similarities. It utilized the PMI formula to measure the similarity of two items, where w and c denoted these two items, $P(w)$ and $P(c)$ denote the probability of these two items being purchased separately, and $P(w, c)$ denoted the probability of these two items being purchased together. After the calculation is completed, the Cofactor created a matrix for all item–item pairs, where the values in the i-th row and j-th column were the PMI values of item i and item j. After that, Liang jointly decomposed the rating matrix and the item–item co-occurrence matrix, making them share the same item latent factors, achieving satisfactory results. Similar to Cofactor, Nguyen [19] utilizes items' co-occurrence patterns to extract relationships between items and embeds them into the latent vectors of the factorization model. Although these works provide some enhancements to matrix factorization-based models, the drawback that the dot product does not satisfy the triangle inequality still limits these models' performance.

3. Proposed Methodology

Traditional metric learning-based social models ignore the potential relationships between users. With the rapid development of recommender systems, more and more auxiliary information can be used to enhance the recommendation model. Users' co-occurrence pattern information has been successfully enhanced by the matrix factorization-based model. However, the dot product in matrix factorization does not satisfy the triangle inequality and therefore fails to capture fine-grained information. To make the recommendations reliable, we consider that the influence of potential relationships between users should be considered in the metric learning-based model. In this section, we propose a social recommender that combines metric learning and users' co-occurrence patterns, called SRMC.

3.1. Motivation

The inspiration for SRMC is that RME can effectively enhance matrix factorization models. Different from the RME which jointly decomposes the user–item rating matrix and the user–user co-occurrence matrix, we utilize the users' co-occurrence patterns to distinguish sets of users with extremely similar or dissimilar consumption behaviors and combine social information to change their relative positions in the metric space. It is worth mentioning that negative sampling is not usually used when calculating the similarity of users' consumption behavior, but several works [20,21] have studied the implications of negative sampling as well as various methods to improve the quality of recommendation. Contrary to previous studies [8,9] which only used the PMI formula to capture the positive similarity between users, we consider negative sampling of user similarity and find the list of users with extremely dissimilar consumption behavior for each user.

In a metric learning-based model, distance reflects preference or similarity, and two users with very dissimilar consumption behavior should be further apart. When constructing the two types of constraints for metric learning, we add user–user pairs with relatively low PMI values to the inequality constraint and keep them at an appropriate distance. This helps to make the distribution of data points in the metric space more reasonable and makes the recommendation results more interpretable. Specifically, the set of users with extremely similar consumption behaviors will be added to the set of equality constraints in metric learning, while the set of users with extremely dissimilar consumption behaviors will be added to the set of inequality constraints. This means that even if user u has only a few ratings, his historical behavior can help determine his location, pull him near his potentially similar users and keep him away from users with extremely dissimilar consumption behavior. Then, items liked by users with extremely similar consumption behavior will be recommended to user u, and items liked by users with extremely dissimilar consumption behavior will not be recommended to user u.

Finally, each user moves closer to trusted friends, preferred items and potentially similar users and away from disliked items and potentially dissimilar users in the space. After that, SRMC will generate more reasonable recommendation results based on the changed distance.

3.2. Model Definition

In SRMC, we construct two constraints in metric learning by this approach: given a user u with several potentially similar users and several potentially dissimilar users and an item i, if user u gives a positive rating to item i, then we add that user u and item i, user u and its potentially similar users to the set of equivalent constraints; otherwise we add user u and item i, user u and its potentially dissimilar users to the set of inequality constraints. The framework of the SRMC model is shown in Figure 1.

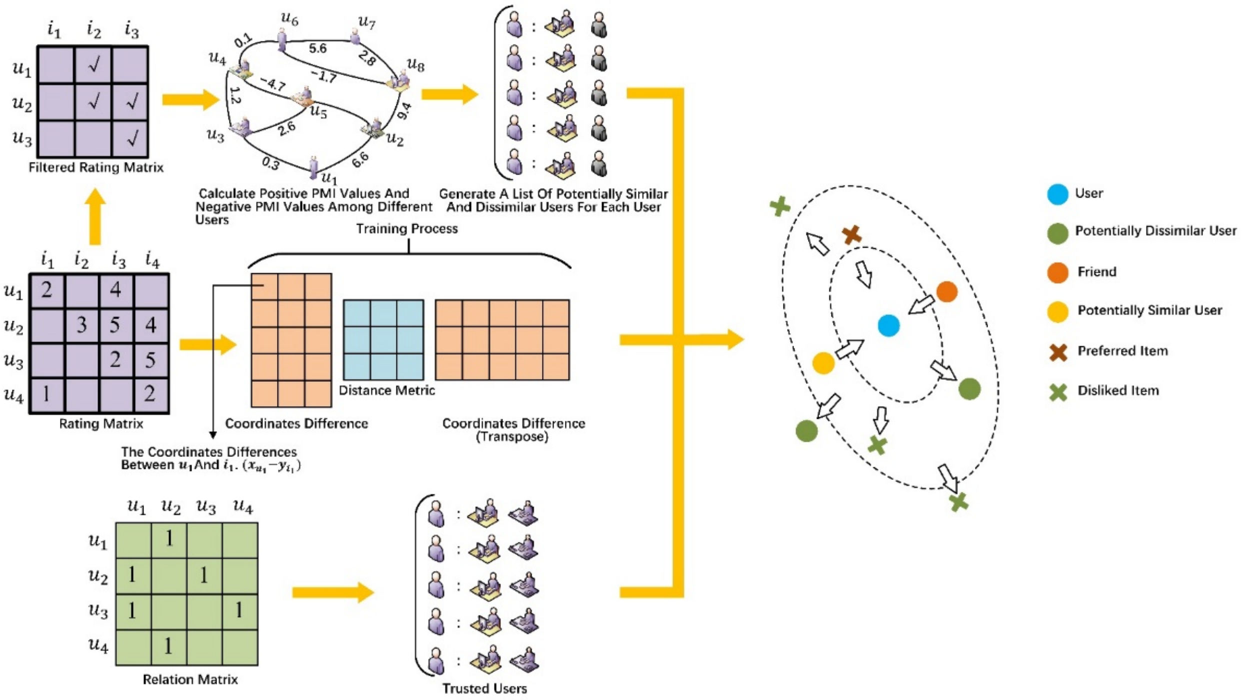


Figure 1. The framework of SRMC.

The calculation process of SRMC is shown below (Algorithm 1).

Algorithm 1 The calculation process of SRMC

```
Algorithm: The Proposed SRMC Algorithm
Input: the user–item rating matrix
Output: the predicted user–item rating matrix
    Step 1    Filter the user–item rating matrix $R$ separately, with matrix $A$ retaining only the positive ratings and matrix $B$ retaining only the negative ratings.
    Step 2    Calculate Positive PMI values between any two users in matrix $A$ and Negative PMI values between any two users in matrix $B$.
    Step 3    Generate a list of potentially similar users and potentially dissimilar users for each user based on the results of step 2.
    Step 4    Map each user and item into the metric space. During the training process, SRMC closes the spatial distance between each user and its positively rated items, trusted users and potentially similar users. At the same time, SRMC pushes away the spatial distance between each user and its negatively rated items and potentially dissimilar users.
    Step 5    After several rounds of training, SRMC places each user and item into the suitable spatial location and utilizes the distance between them to predict ratings in the testing set.
```

The predicted rating of user u and item i is determined by the distance between them and can be defined as:

$$\hat{r}_{ui} = \mu + b_u + b_i - \parallel x_u - y_i \parallel_A^2 \tag{5}$$

where μ is the global mean, b_u is user bias, b_i is item bias and x_u and y_i are point vectors of user u and item i in the metric space. $\parallel x_u - y_i \parallel_A^2$ is the squared Mahalanobis distance between user u and item i. The reason why we utilize squared Mahalanobis distances is that they are cheaper to calculate than Mahalanobis distances and the impact on accuracy is minimal. A is a positive semi-definite matrix that can be calcalated by $A = HH^T$. The loss function of the SRMC is shown below:

$$\begin{aligned}\mathcal{L} = \tfrac{1}{2} \sum_{(u,i)\in R_{m\times n}} \omega_{ui}(r_{ui} - \hat{r}_{ui})^2 + \tfrac{\eta}{2} \sum_{(u,v)\in S_u \cup T_u} \parallel x_u - x_v \parallel_A^2 + \tfrac{\eta}{2} \sum_{(u,v)\in D_u} [\theta - \parallel x_u - x_v \parallel_A^2]_+ \\ + \tfrac{\alpha}{2} \sum_{(u,i)\in P_u} \parallel x_u - y_i \parallel_A^2 + \tfrac{\alpha}{2} \sum_{(u,i)\in N_u} [\theta - \parallel x_u - y_i \parallel_A^2]_+ + \tfrac{\lambda}{2} \left(\sum_{u=1}^{m} b_u^2 + \sum_{i=1}^{n} b_i^2 \right)\end{aligned} \tag{6}$$

where S_u and T_u denote the potentially similar users and trusted users of user u. D_u is the potentially dissimilar users of user u. P_u and N_u are the sets of positively and negatively rated items of user u, respectively. $[\mathcal{Z}]_+ = \max(\mathcal{Z}, 0)$ is the standard hinge loss. λ controls the magnitudes of biases. η and α control the magnitude of two constraints. With these two constraints, users are guaranteed to be closer to their trusted users, positively rated items and potentially similar users, but further away from their negatively rated items and potentially dissimilar users.

$$\omega_{ui} = 1 + \varphi \left| r_{ui} - \frac{R_{\max}}{2} \right| \tag{7}$$

ω_{ui} indicates the confidence level. For extremely high or extremely low ratings, we will assign greater weights. φ controls the size of the confidence level, and $R_{\max}/2$ indicates the median rating of the current dataset.

SRMC optimizes the loss function using stochastic gradient descent and updates, x_u, y_i, H, b_u, b_i by:

$$e_{ui} = r_{ui} - \hat{r}_{ui} = r_{ui} - \left(\mu + b_u + b_i - \parallel x_u - y_i \parallel_A^2 \right) W = H \cdot H^T + H^T \cdot H \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial x_u} = e_{ui}(x_u - y_i)W \pm \eta(x_u - x_v)W \pm \alpha(x_u - y_i)W \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial y_i} = -e_{ui}(x_u - y_i)W \pm \alpha(x_u - y_i)W \tag{10}$$

$$\frac{\partial \mathcal{L}}{\partial H} = e_{ui}H(x_u - y_i)^T(x_u - y_i) \pm \eta H(x_u - x_v)^T(x_u - x_v) \pm \alpha H(x_u - y_i)^T(x_u - y_i) \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial b_u} = \lambda b_u - e_{ui} \tag{12}$$

$$\frac{\partial \mathcal{L}}{\partial b_i} = \lambda b_i - e_{ui} \tag{13}$$

where e_{ui} is the difference between the true rating and the predicted rating.

3.3. Discover Potentially Similar and Dissimilar Users

Considering that our aim is to find potentially similar or dissimilar users, we first filter the ratings in the rating matrix by a threshold, and any rating below the threshold will be removed. For different datasets, the threshold is also different. The first purpose of filtering is to ensure that both user u and user v have positively or negatively rated the items they have jointly rated, rather than just rated them. The second purpose is to ensure that the found users' consumption behavior is extremely similar or extremely dissimilar. For instance, for item i, user u has a positive rating and user v has a negative rating. It is obvious that for item i, user u and user v do not have the same preference. However, if the PMI value between user u and user v is calculated directly without filtering the rating matrix, the wrong situation of considering user u and user v. as potentially similar users may occur.

After filtering the rating matrix, the PMI values between users are calculated as follows:

$$\mathrm{PMI}(u_i, u_j) = \log \frac{\#(u_i, u_j) \cdot |D|}{\#(u_i)\#(u_j)}. \tag{14}$$

where $|D|$ indicates all user–user pairs for which a rating is in the filtered rating matrix. $\#(u_i)$ and $\#(u_j)$ represent the number of interactions between user i and user j, respectively. $\#(u_i, u_j)$ represents the number of items that are positively rated by both u_i and u_j. For instance, if u_i and u_j both have positive ratings for $[i_1, i_2, i_3]$ and they have no common ratings for other items in all item sets, then $\#(u_i, u_j)$ = 3.

The higher the $\mathrm{PMI}(u, v)$ is, the more positive the rated items that both user u and user v have rated, and the more similar the consumption behavior of user u and user v is. Conversely the lower $\mathrm{PMI}(u, v)$ is, the less relevant and less similar user u and user v are.

Previous studies such as Cofactor, RME only focused on the case where the PMI value was positive, but we think it is necessary to consider when the PMI value is negative. When calculating PMI values in the Skip-Gram model, there are cases where PMI values are negative or even negative infinity. This means that the (w, c) word-context pairs rarely or never appear together in the sliding window. In SRMC, the above case can be interpreted as the number of items with positive ratings from both user u and user v being very few or even zero. In other words, the consumption behaviors of user u and user v are extremely dissimilar. Previous studies tend to ignore the situation when PMI is negative and filter the PMI values by:

$$\mathrm{SPPMI}(u, v) = \max(\mathrm{PMI}(u, v) - \log(k), 0) \tag{15}$$

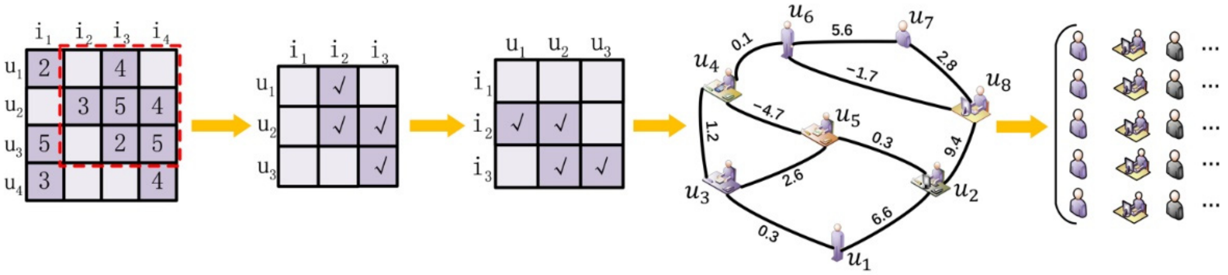
where k is the parameter that controls the size of the SPPMI matrix. Later, those models based on matrix factorization will jointly decompose the rating matrix and the SPPMI matrix. When $\mathrm{PMI}(u, v)$ is negative, SRMC considers user u and user v to be extremely dissimilar and increases their distance in metric space. Therefore, we filter the PMI values by the following equation:

$$\text{Positive PMI}(u, v) = \max(\mathrm{PMI}(u, v) - \log(k_1), 0) \tag{16}$$

$$\text{Negative PMI}(u, v) = \max(\mathrm{PMI}(u, v) + \log(k_2), \sigma) \tag{17}$$

When the PMI value between user u and user v is positive, SRMC uses Formula (16) to determine whether user v is a potentially similar user to user u. Similarly, when the PMI value between user u and user v is negative, SRMC uses the Formula (17) to determine whether user v is a potentially dissimilar user for user u. σ is a small negative number, and the value of σ varies depending on the datasets. k_1 and k_2 are also two constants that change with the datasets, ensuring that the users have extremely similar or extremely dissimilar consumption behaviors.

After calculating the PMI values among all of the users in the filtered rating matrix, we can obtain the set of potentially similar or dissimilar users for each user. The process of discovering potentially similar or dissimilar users is shown in Figure 2.



Filtering The User-Item Rating Matrix

Transpose The User-Item Rating Matrix In Order To Calculate The PMI Values Between Different Users

Calculate The Positive PMI Values And Negative PMI Values Among Different Users.

Generate A List of Potentially Similar And Dissimilar Users For Each User

Figure 2. Discovering potential similar or dissimilar users.

4. Experiments

In this section, we present the experimental results of the SRMC on three public datasets and conduct two main experiments: (1) the first experiment includes the recommendation quality of SRMC compared to other algorithms and (2) the second experiment examines the effect of parameters on SRMC.

4.1. Datasets and Evaluation Metrics

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

We used three public datasets that provide user–item ratings and trust relationships to validate SRMC's recommendation effect (Table 1). FilmTrust's [22] data were obtained from the FilmTrust website, containing 1508 users and 2071 movies. The sparsity of its data is 98.86%. FilmTrust's rating range is [0.5, 4], and the step size is 0.5. The Douban [23] dataset has a data sparsity of 99.21% and contains 2848 users, 39,586 items and 894,887 ratings. Douban's rating range is [1, 5], and the step size is 1. The Epinions [24] dataset has a data sparsity of 99.98% and contains 49,286 users, 139,738 items and 664,824 ratings. Epinions' rating range is [1, 5], and the step size is 1.

Table 1. Dataset Information.

Datasets	Users	Items	Ratings	Relations
FilmTrust	1508	2071	35,497	1853
Douban	2848	39,586	894,887	35,770
Epinions	49,290	139,738	664,824	487,181

We use root mean square error (RMSE) and mean absolute error (MAE), the two most commonly used evaluation metrics in recommender systems, as our evaluation criteria.

RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i}(r_{ui} - \hat{r}_{ui})^2}{N}} \tag{18}$$

We use root mean square error (RMSE) and mean absolute error (MAE), the two most commonly used evaluation metrics in recommender systems, as our evaluation criteria.

$$\mathrm{MAE} = \frac{\sum_{u,i}|r_{ui} - \hat{r}_{ui}|}{N} \tag{19}$$

where N indicates the number of ratings in the test set, r_{ui} is the true rating and $\hat{r}_{ui}$ is the predicted rating. A lower RMSE/MAE indicates that missing ratings are predicted more precisely. Lower values of RMSE and MAE indicate more accurate rating predictions of SRMC.

4.2. Algorithm Comparisons

To demonstrate the performance improvement of SRMC, we conducted a series of experiments to test our proposed SRMC. We chose several representative and relatively new models incorporating social relationships as our comparison algorithms. These models are shown below:

SoRec: This model shares the same user latent space to factorize the user–item rating matrix and the user–user social matrix based on the probability matrix factorization (PMF).

SocialMF [14]: This introduces a trust propagation mechanism based on matrix factorization to alleviate the cold start problem.

SoReg [25]: This model uses social regularization to denote social constraints, making two potentially similar users more similar in terms of latent feature vectors.

UE-SVD++ [26]: This method is a matrix factorization-based model that jointly decomposes the rating matrix and the user–user co-occurrence matrix.

SocialFD: In the metric space, SocialFD brings users closer to their preferred items, pushes them farther away from their disliked items, and brings them closer to their friends in space.

4.3. Model Parameter Selection

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

The selection of model parameters has a substantial impact on the recommendation performance of SRMC. In this section, we investigate the impact of several important parameters (α, η, λ, lr) on SRMC. In our experiments, we used the 5-fold cross validation method to process the dataset. The whole dataset was randomly divided into five parts, with 80% of the data as the training set and the remaining 20% as the testing set.

In this section, we utilize the control variables method to determine the most suitable parameters for SRMC. First, we determined the range of values for each parameter based on previous work, fixed the other three parameters and continuously adjusted the current parameter α until the optimal value was found. Then, we proceeded in a similar manner to find the optimal values for other parameters of SRMC.

From Figures 3–5, we can see that the variation of rating prediction accuracy is mainly affected by α, η and λ, because these three parameters control the distribution of data points in the space. When α is too small, SRMC cannot effectively reduce the spatial distance from the preferred items and increase the spatial distance from the disliked items for each user. Conversely, when α is too large, SRMC reduces the distance that should not be reduced smaller and increases the distance that should not be increased larger, which impacts the recommendation effect. Similarly, when η is too small, SRMC cannot effectively reduce the spatial distance for each user with trusted users and potentially similar users, as well as increase the spatial distance with potentially dissimilar users, and vice versa. λ ensures that the biases and the distance between two points in the space are in a suitable range. If either too large or too small λ will affect the recommendation effect.

It is also worth mentioning how SRMC defines preferred and undesired items and how it defines potential similar users and potential dissimilar users. For these two questions, we conducted many experiments; for the question asking whether users like an item, we made a judgment based on the ratings given by users; and for the question asking whether users are potentially similar or dissimilar to each other we made a judgment based on the PMI values from users to users.

In the FilmTrust dataset, if a user gives a rating of 3.0, 3.5 or 4.0, we assume that the user likes the item; if a user gives a rating of 0.5, 1.0 or 1.5, we assume that the user does not like the item. In the following, we consider how to distinguish potentially similar users from dissimilar users. If the PMI value between users is higher than 4.5, we consider these two users to be potentially similar users. If the PMI value between users is less than -0.6, we consider these two users to be potentially dissimilar users.

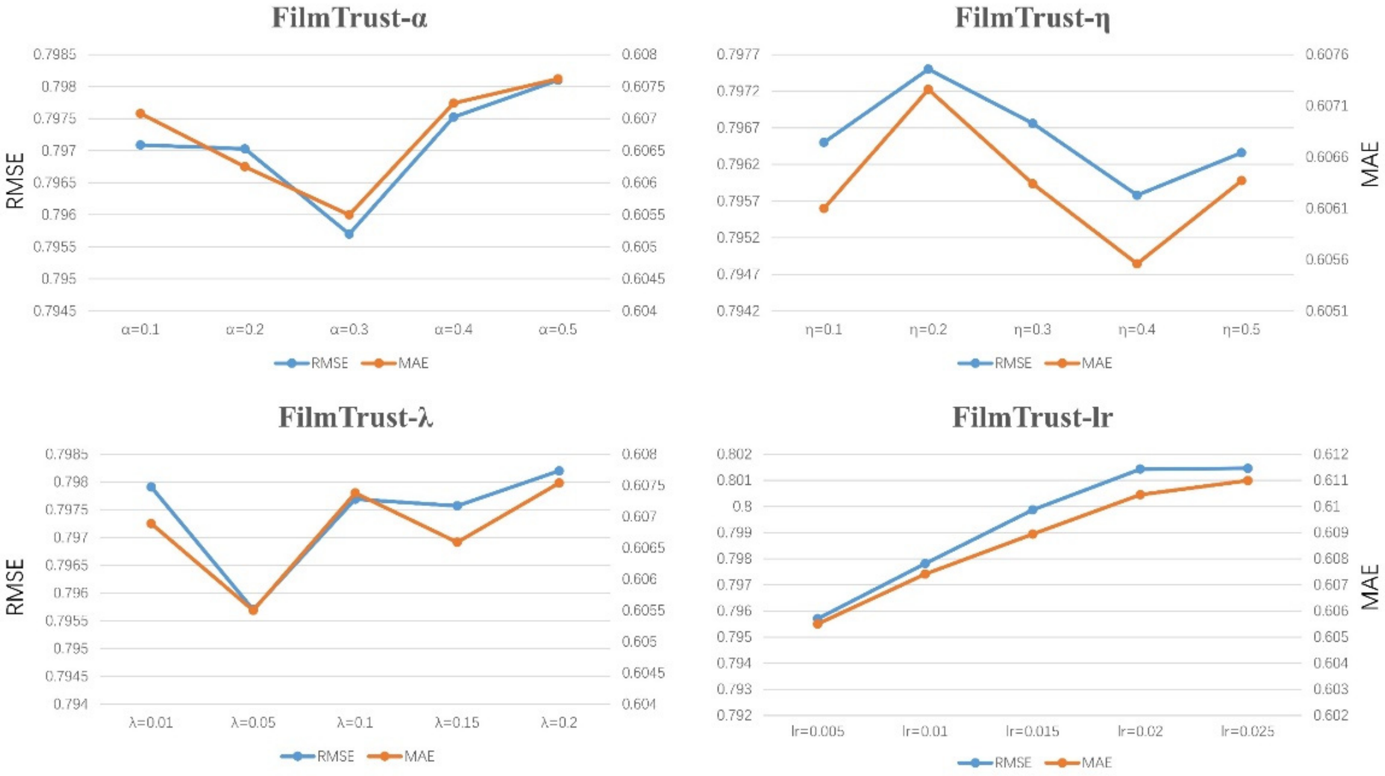


Figure 3. Changes of each parameter on FilmTrust.

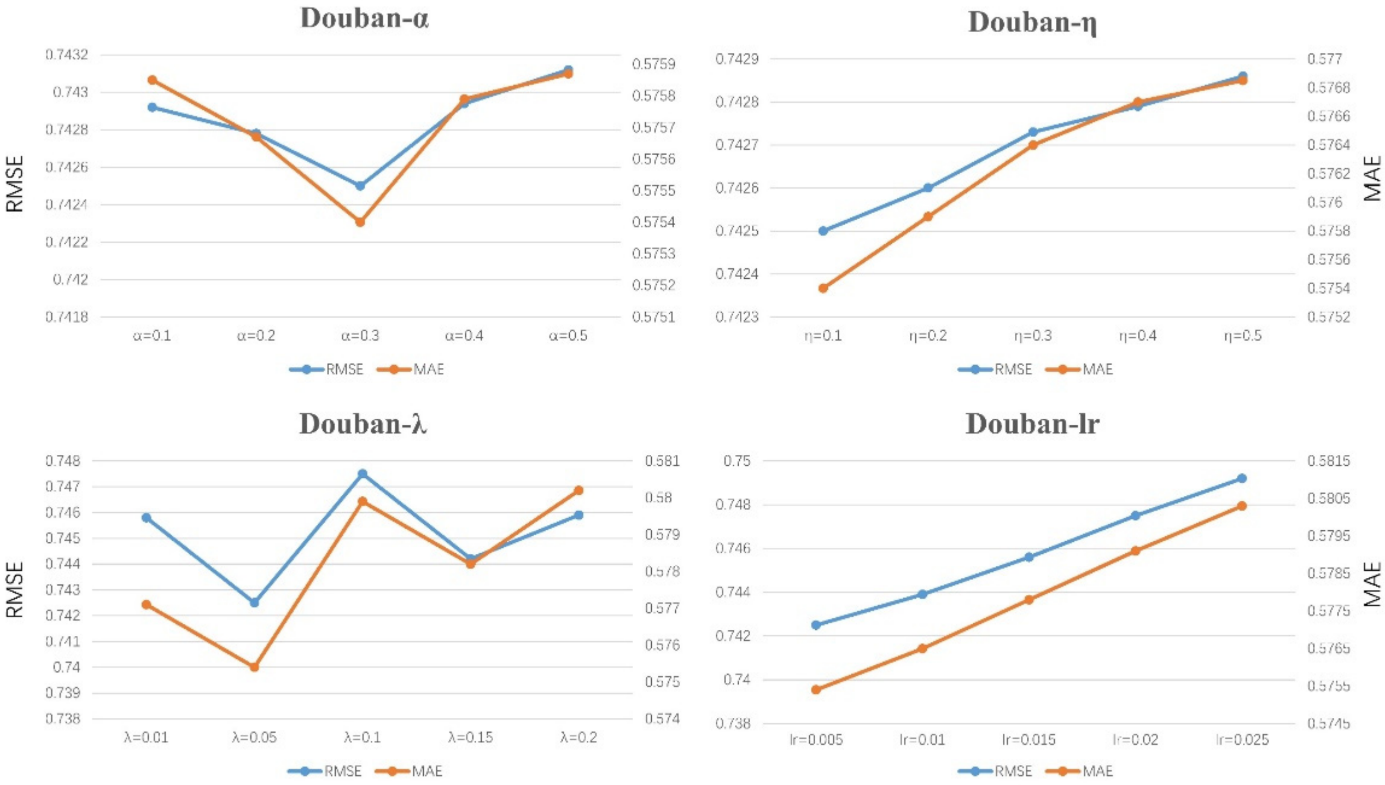


Figure 4. Changes of each parameter on Douban.

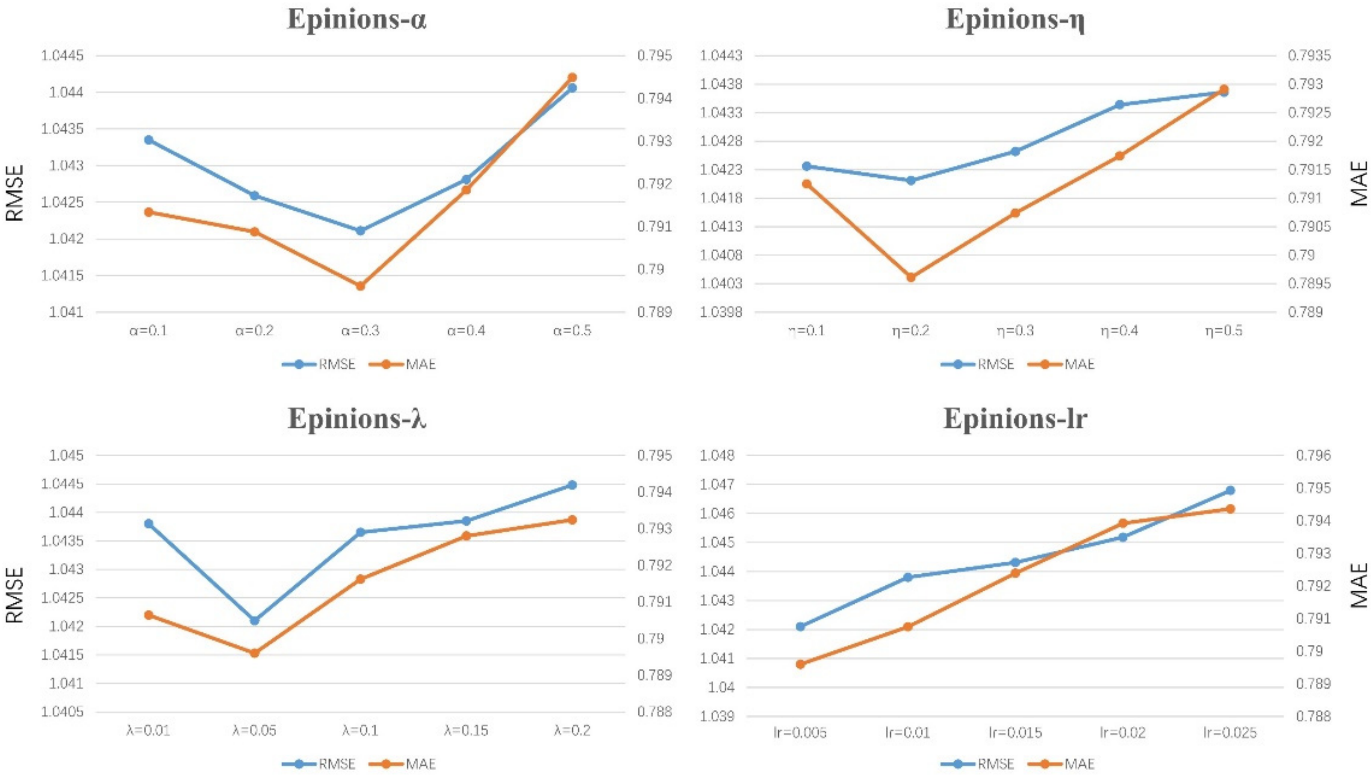


Figure 5. Changes of each parameter on Epinions.

In the Douban dataset, if a user gives a rating of 4 or 5, we assume that the user prefers the item, and if the user gives a rating of 1 or 2, we assume that the user does not prefer the item. In the following, we consider how to distinguish potentially similar users from dissimilar users. If the PMI value between users is higher than 2.5, we consider these two users to be potentially similar users. If the PMI value between users is less than −1.7, we consider these two users to be potentially dissimilar users.

In the Epinions dataset, if a user gives a rating of 4 or 5, we assume that the user likes the item, and if the user gives a rating of 1 or 2, we assume that the user does not like the item. In the following, we consider how to distinguish potentially similar users from dissimilar users. If the PMI value between users is higher than 6, we consider these two users as potentially similar users. If the PMI value between users is less than −0.7, we consider these two users as potentially dissimilar users.

4.4. Performance Comparison

Table 2 shows the experimental results of SRMC and other baseline algorithms. After extensive experiments, we determined the most suitable parameters for SRMC on different datasets. The parameters of SRMC on FilmTrust are: $\alpha = 0.3$, $\eta = 0.4$, $\lambda = 0.05$ and $lr = 0.005$, while the parameters on Douban are: $\alpha = 0.3$, $\eta = 0.1$, $\lambda = 0.05$ and $lr = 0.005$ and the parameters on Epinions are: $\alpha = 0.3$, $\eta = 0.2$, $\lambda = 0.05$ and $lr = 0.005$.

Compared with the baseline algorithm, SRMC has the best recommendation results among all of the datasets. Compared with the suboptimal algorithm SocialFD, SRMC improves RMSE by 1.34% and MAE by 3.35% in the FilmTrust dataset and improves RMSE by 2.15% and MAE by 4.72% in Epinions. It is worth noting that similar to SRMC, SocialFD maps all users and items into the metric space and brings users closer to trusted users and user-preferred items, and pushes users further away from unpreferred items. Nevertheless, SRMC is more effective than SocialFD. In addition, we control the relative distance between data points in the regularization term, which makes the data distribution in the whole metric space more compact and achieves a better recommendation effect. The comparison chart between SRMC and other algorithms is shown in Figure 6.

Table 2. Experimental Results.

Dataset	FilmTrust		Douban		Epinions	
Metrics	RMSE	MAE	RMSE	MAE	RMSE	MAE
SoRec	0.8368	0.6485	0.7714	0.6158	1.2452	0.9369
SocialMF	0.8317	0.6482	0.7737	0.6171	1.2904	0.9638
SoReg	0.8445	0.6432	0.7741	0.6128	1.2803	0.9608
UE-SVD++	0.8145	0.6338	0.7567	0.5951	1.0694	0.8293
SocialFD	0.8065	0.6265	0.7533	0.5945	1.0650	0.8287
SRMC	0.7957	0.6055	0.7425	0.5754	1.0421	0.7896
Improved	1.34%	3.35%	1.43%	3.21%	2.15%	4.72%

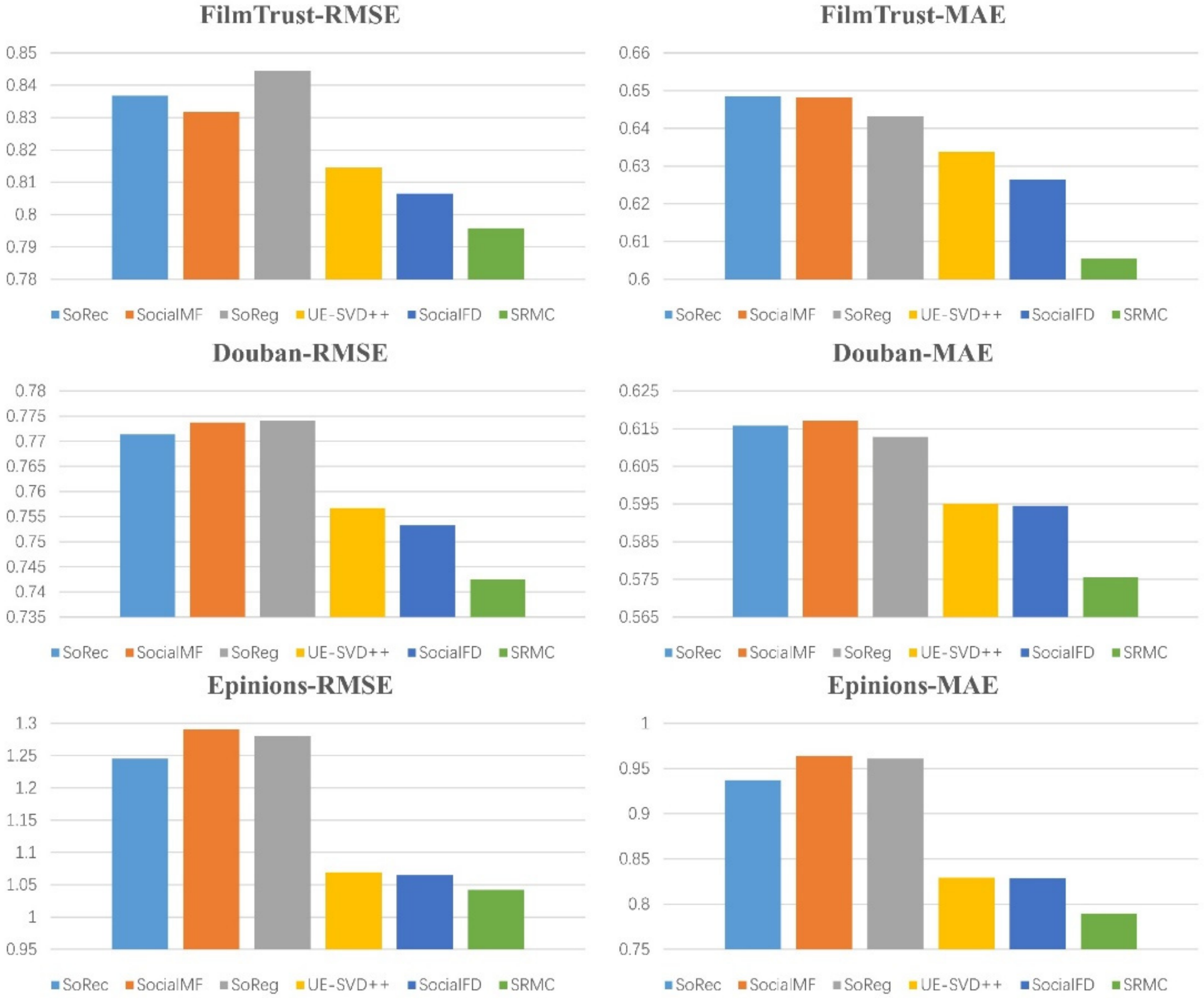


Figure 6. Performance Comparison.

4.5. Performance for Predicting Ratings

The prediction of missing ratings in the rating matrix is an important task for recommender models. In this section, we compare the performance of SRMC with the baseline models. To ensure the reliability and fairness of the experimental results, we compare them with the optimal performance of each model based on the authors' recommended parameters. From Table 2, we can see that the performance of SRMC is the best in all three public datasets and shows a significant advantage on the Epinions dataset. It is worth noting that UE-SVD++ also exploits the user's co-occurrence model and extends the matrix factorization-based model. In addition, UE-SVD++ outperforms most traditional recommendation models such as SVD++ [27], MFC [28], FUNK-SVD, etc. Different from our work, UE-SVD++ only filters the high rating data in the user–item rating matrix. It then generates a user co-occurrence matrix by calculating the PMI values among these positively rated users and jointly decomposes the user co-occurrence matrix and the rating matrix. The filtering of high rating data by UE-SVD++ can be seen as only filtering out users with similar consumption behaviors but ignoring users with dissimilar consumption behaviors, which limits its performance. SocialFD is similar to SRMC, but it only brings the user closer to its trusted friends and positively rated items, and only pushes the user further away from the negatively rated items. By utilizing users' co-occurrence patterns to discover potentially similar users and potentially dissimilar users, SRMC more reasonably adjusts the relative position of each user and item in the metric space. Thus, SRMC outperforms SocialFD in all three public datasets. By comparing with other social recommendation models, we validate the effectiveness of SRMC compared with other matrix factorization-based models.

5. Conclusions and Future Work

This paper is inspired by RME's effective enhancement for matrix factorization models. Considering the limitations of the matrix factorization model, we propose a new recommendation model, called SRMC, which is based on metric learning and users' co-occurrence patterns. SRMC not only utilizes the traditional auxiliary information such as social networks, but also utilizes the neighborhood effect in the rating data to explore the user's consumption behavior and find a list of users who are not necessarily socially related but have extremely similar and extremely dissimilar users for each user. During the training process, SRMC constructs two types of constraints in metric learning as such: (user-preferred items), (user-trusted friends) and (user-potentially similar users) will be used as the set of equivalent constraints; (user-disliked items), (user-potentially dissimilar users) will be used as the set of inequivalent constraints. The position of users and items is jointly determined by ratings, social relationships, potentially similar users and potentially dissimilar users, which can help to alleviate the data sparsity problem in the recommendation system. At the end of training, the obtained distances are used to generate understandable and reliable recommendations.

Most of the recommendation models based on metric learning, including SRMC, rely on a fixed margin θ. The margin θ in SRMC is similar to the user bias in matrix factorization models. Different users tend to have different criteria, and the number of items interacted by each user also varies greatly. If the fixed margin is too large, the user will be surrounded by too many positively rated items, and the items that should be recommended will most likely be negative samples. If the fixed margin is too small, the positive and negative rated items will be too close together for them to be distinguished, resulting in a false recommendation. Adopting a fixed margin may limit the expressiveness of the model, especially when the data distribution is complex. Therefore, different users should have adaptive margins. We will explore this potential direction in the future. In addition, how to identify potentially similar users and potentially dissimilar users more effectively will also be our future work.

Author Contributions: Conceptualization, J.Q. and X.Z.; methodology, J.Q.; software, X.Z.; validation, X.Z., J.Z.; formal analysis, J.Q.; investigation, X.Z.; resources, J.Z.; data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, J.Q.; visualization, J.Z.; supervision, J.Q.; project administration, J.Q.; funding acquisition, J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science Fund for Outstanding Youth of Xinjiang Uygur Autonomous Region under Grant No. 2021D01E14, the National Science Foundation of China under Grant No. 61867006, the Major Science and Technology Project of Xinjiang Uygur Autonomous Region under Grant No. 2020A03001, the Innovation Project of Sichuan Regional under Grant No. 2020YFQ2018 and the Key Laboratory Open Project of Science & Technology Department of Xinjiang

Uygur Autonomous Region named Research on video information intelligent processing technology for Xinjiang regional security.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable. No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, A.R.; Son, S.M.; Kim, K.K. Information and Communication Technology Overload and Social Networking Service Fatigue: A Stress Perspective. *Comput. Hum. Behav.* **2016**, *55*, 51–61. [CrossRef]
2. Liu, B.; Zeng, Q.; Lu, L.; Li, Y.; You, F. A Survey of Recommendation Systems Based on Deep Learning. *J. Phys. Conf. Ser.* **2021**, *1754*, 012148. [CrossRef]
3. Hsieh, C.-K.; Yang, L.; Cui, Y.; Lin, T.-Y.; Belongie, S.; Estrin, D. Collaborative Metric Learning. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017.
4. Wang, F.; Sun, J.M. Survey on Distance Metric Learning and Dimensionality Reduction in Data Mining. *Data Min. Knowl. Discov.* **2015**, *29*, 534–564. [CrossRef]
5. Yu, J.; Gao, M.; Song, Y.; Zhao, Z.; Rong, W.; Xiong, Q. Connecting Factorization and Distance Metric Learning for Social Recommendations. *Int. Conf. Knowl. Sci. Eng. Manag.* **2017**, *10412*, 389–396.
6. Yi, T.; Tuan, L.A.; Hui, S.C. Latent Relational Metric Learning via Memory-Based Attention for Collaborative Ranking. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018.
7. Sánchez-Moreno, D.; Batista, V.L.; Vicente, M.; Lázaro, Á.L.S.; Moreno-García, M.N. Exploiting the User Social Context to Address Neighborhood Bias in Collaborative Filtering Music Recommender Systems. *Information* **2020**, *11*, 439. [CrossRef]
8. Liang, D.; Altosaar, J.; Charlin, L.; Blei, D.M. Factorization Meets the Item Embedding: Regularizing Matrix Factorization with Item Co-Occurrence. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–16 September 2016.
9. Tran, T.; Lee, K.; Liao, Y.; Lee, D. Regularizing Matrix Factorization with User and Item Embeddings for Recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Turin, Italy, 22–26 October 2018.
10. Roman, C.R.; Precup, R.E.; Petriu, E.M. Hybrid Data-Driven Fuzzy Active Disturbance Rejection Control for Tower Crane Systems. *Eur. J. Control.* **2021**, *58*, 373–387. [CrossRef]
11. Zhu, Z.; Pan, Y.; Zhou, Q.; Lu, C. Event-Triggered Adaptive Fuzzy Control for Stochastic Nonlinear Systems with Unmeasured States and Unknown Backlash-Like Hysteresis. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 1273–1283. [CrossRef]
12. Qiu, J.; Tang, J.; Ma, H.; Dong, Y.; Wang, K.; Tang, J. Deepinf: Modeling Influence Locality in Large Social Networks. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18), London, UK, 19–23 August 2018.
13. Ma, H.; Yang, H.; Lyu, M.R.; King, I. Sorec: Social Recommendation Using Probabilistic Matrix Factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, CA, USA, 26–30 October 2008.
14. Jamali, M.; Ester, M. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010.
15. 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 Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
16. Zhao, T.; McAuley, J.; King, I. Leveraging Social Connections to Improve Personalized Ranking for Collaborative Filtering. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014.
17. Yang, L.; Jin, R. Distance Metric Learning: A Comprehensive Survey. *Mich. State Univ.* **2006**, *2*, 4.
18. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
19. Nguyen, T.; Aihara, K.; Takasu, A. Collaborative Item Embedding Model for Implicit Feedback Data. In Proceedings of the International Conference on Web Engineering, Rome, Italy, 5–8 June 2017.
20. Yang, J.; Yi, X.; Cheng, D.Z.; Hong, L.; Li, Y.; Wang, S.X.; Xu, T.; Chi, E.H. Mixed Negative Sampling for Learning Two-Tower Neural Networks in Recommendations. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020.
21. Tran, V.-A.; Hennequin, R.; Royo-Letelier, J.; Moussallam, M. Improving Collaborative Metric Learning with Efficient Negative Sampling. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019.
22. Filmtrust. Available online: https://guoguibing.github.io/librec/datasets.html (accessed on 16 August 2021).
23. Douban. Available online: https://book.douban.com (accessed on 16 August 2021).

24. Epinions. Available online: http://www.trustlet.org/epinions.html (accessed on 16 August 2021).
25. Ma, H.; Zhou, D.; Liu, C.; Lyu, M.R.; King, I. Recommender Systems with Social Regularization. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011.
26. Shi, W.; Wang, L.; Qin, J. User Embedding for Rating Prediction in Svd++-Based Collaborative Filtering. *Symmetry* **2020**, *12*, 121. [CrossRef]
27. Koren, Y. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. *ACM Trans. Knowl. Discov. Data* **2010**, *4*, 1–24. [CrossRef]
28. Li, H.; Wu, D.; Tang, W.; Mamoulis, N. Overlapping Community Regularization for Rating Prediction in Social Recommender Systems. In Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria, 16–20 September 2015.