

---

Article

# A Social Recommendation based on metric learning and Users' Co-occurrence Pattern

Xin Zhang<sup>1,2</sup>, Jiwei Qin<sup>1,2,\*</sup> and Jiong Zheng<sup>3</sup>

<sup>1</sup> School of Information Science and Engineering, Xinjiang University, Urumqi, 830046, China; zhang-xin\_996@stu.xju.edu.cn

<sup>2</sup> Key Laboratory of Signal Detection and Processing, Xinjiang Uygur Autonomous Region, Xinjiang University, Urumqi, 830046, China; jwqin\_xju@163.com

<sup>3</sup> School of Information Science and Engineering, Xinjiang University, Urumqi, 830046, China; sky\_gingo@126.com

\* Correspondence: jwqin\_xju@163.com;

**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 pattern 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

---

## 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 [5,6,7] 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 [7] 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 [8]. 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 [9], 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 similarity, effectively enhancing the recommendation model. Tran's RME [9] further extends the Cofactor with the co-occurrence patterns of users and successfully proves that 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. In our work, 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 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 alleviate this problem. Social recommender systems assume that users are influenced by users with social relationships, resulting in some similarity in their preferences [10]. 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 [11] model utilized users' social networks to alleviate the data sparsity problem and improve the recommendation effect. Mohsen [12] introduces the trust propagation principle into the matrix factorization model. Guo exploited the implicit information and proposed TrustSVD [13]. Zhao introduced social information into the Bayesian personalized ranking algorithm and proposed SBPR. [14]

### 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 [15]. 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 (1),  $A \in R^{m \times m}$  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:

$$\begin{aligned} \min_{A \in R^{m \times m}} & \sum_{(x_i, x_j) \in S} d_A^2(x_i, x_j) \\ \text{s. t.} & \sum_{(x_i, x_j) \in D} d_A^2(x_i, x_j) \geq \theta, A \geq 0 \end{aligned} \quad (2)$$

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.  $\theta$  is the minimum distance between two different classes of data points.

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 [16] 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 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:

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

It can also be written as:

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

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 similarity. 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.

### 3. Proposed Methodology

In this section, we propose a social recommendation model based on metric learning and the users' co-occurrence pattern (SRMC). The inspiration for SRMC is that Cofactor



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.

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

$\omega_{ui}$  indicates the confidence level. For extremely high or extremely low ratings, we will assign greater weights.  $\varphi$  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:

$$e_{ui} = r_{ui} - \hat{r}_{ui} = r_{ui} - (\mu + b_u + b_i - \|x_u - y_i\|_A^2)W = H \cdot H^T + H^T \cdot H \quad (8)$$

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

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

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

$$\frac{\partial \mathcal{L}}{\partial b_u} = \lambda b_u - e_{ui} \quad (12)$$

$$\frac{\partial \mathcal{L}}{\partial b_i} = \lambda b_i - e_{ui} \quad (13)$$

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

### 3.2. Discover Potentially Similar Or 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 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:

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

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  $PMI(u, v)$  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  $PMI(u, v)$  is, the less relevant and less similar user  $u$  and user  $v$  are.

Another issue worth considering is what happens when the PMI 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:

$$SPPMI(u, v) = \max(PMI(u, v) - \log(k), 0) \quad (15)$$

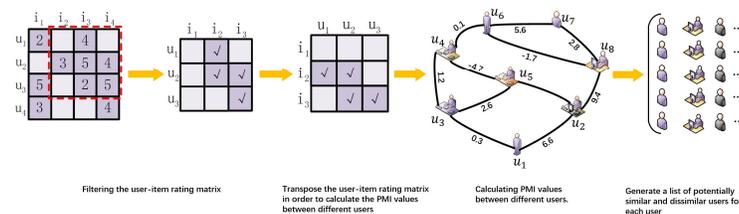
where  $k$  is the parameter that controls the size of the SPPMI matrix. Later, those models based on matrix factorization will jointly decompose the rating matrix and the SPPMI matrix. When  $PMI(u, v)$  is negative, SRMC considers user  $u$  and user  $v$  to be extremely dissimilar and increases their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

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

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

When the PMI value between user  $u$  and user  $v$  is positive, SRMC uses the 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  $u$ .  $\sigma$  is a small negative number, and the value of  $\sigma$  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.



**Figure 2.** Discovering potential similar or dissimilar users

#### 4. Experiment

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. FilmTrust[17]'s 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[18] dataset has a data sparsity of 99.21% and contains 2848 users, 39586 items, and 894887 ratings. Douban' rating range is [1, 5], and the step size is 1. The Epinions[19] 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	1,508	2,071	35,497	1,853
Douban	2,848	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:

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

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.

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

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 prediction of SRMC.

#### 4.2. Compared Algorithms

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

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

UE-SVD++: 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 ( $\alpha, \eta, \lambda, lr$ ) on SRMC. In our experiments, we use the 5-fold cross validation method to process the dataset. The whole dataset is randomly divided into five parts, with 80% of the data as the training set and the remaining 20% as the test set.

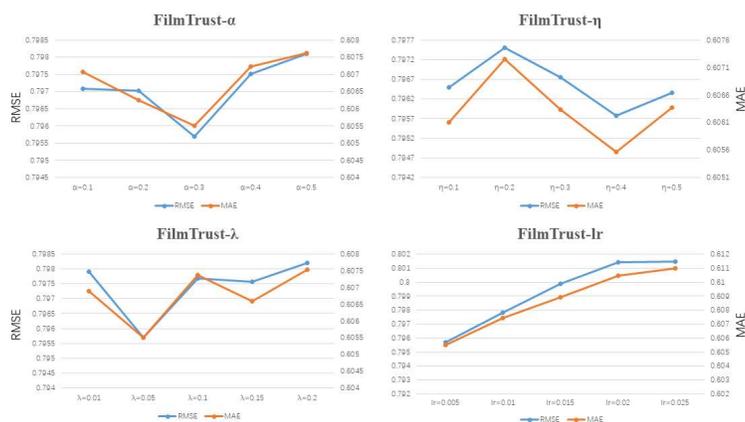


Figure 3. Changes of each parameter on FilmTrust

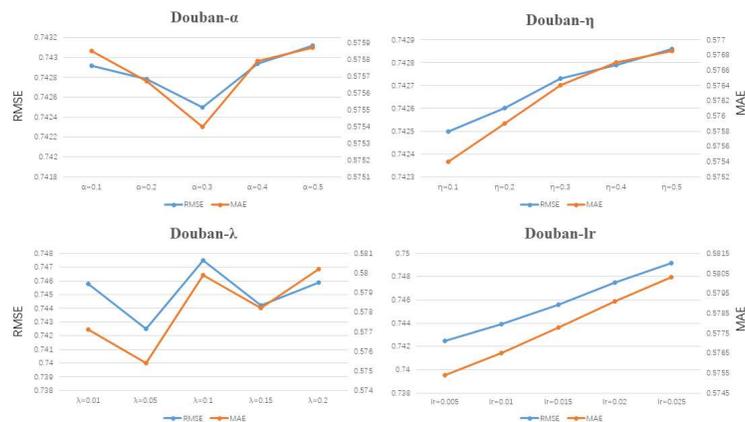
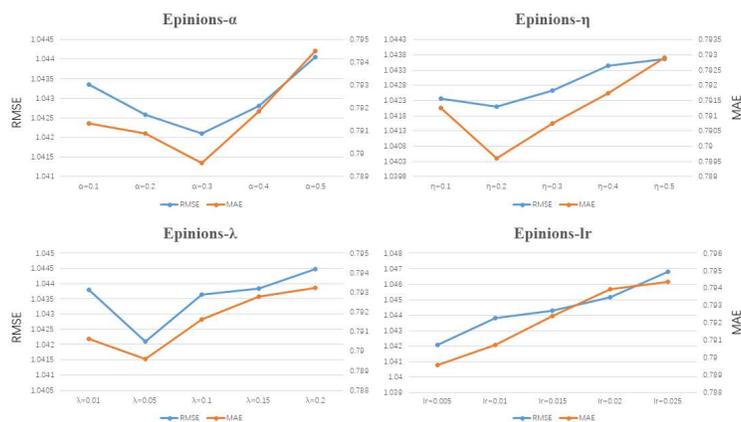


Figure 4. Changes of each parameter on Douban



**Figure 5.** Changes of each parameter on Epinions

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.

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

From Figures 3,4 and 5, we can see that the variation of rating prediction accuracy is mainly affected by  $\alpha$ ,  $\eta$  and  $\lambda$ , because these three parameters control the distribution of data points in the space. When  $\alpha$  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  $\alpha$  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  $\eta$  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.  $\lambda$  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  $\lambda$  will affect 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 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.

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

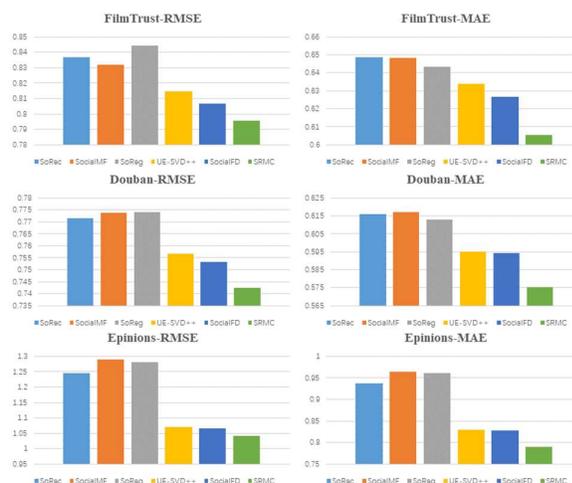
**Table 2.** Experimental Results

Dataset	FilmTrust		Douban		Epinions	
	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	<b>0.8065</b>	<b>0.6265</b>	<b>0.7533</b>	<b>0.5945</b>	<b>1.0650</b>	<b>0.8287</b>
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%

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.

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.



**Figure 6.** Performance Comparison

## 5. Conclusion and future work

This paper is inspired by Cofactor'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 user co-occurrence patterns. SRMC utilizes user co-occurrence patterns to distinguish users from extremely similar or dissimilar consumption behaviors and changes the relative positions of different users and items in the metric space. The obtained distances are used to generate understandable and reliable recommendations. In future work, we will try to extend SRMC with other auxiliary information (e.g., item content) and hope to validate SRMC on a more larger dataset.

**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 Grant under 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.

**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[J]. *Computers in Human Behavior*, **2016**, 55: 51-61.
2. Liu B, Zeng Q, Lu L, et al. A survey of recommendation systems based on deep learning[C]//*Journal of Physics: Conference Series. IOP Publishing*, **2021**, 1754(1): 012148.
3. Hsieh C K, Yang L, Cui Y, et al. Collaborative metric learning[C]//*Proceedings of the 26th international conference on world wide web*. 2017: 193-201.
4. Wang F, Sun J. Survey on distance metric learning and dimensionality reduction in data mining[J]. *Data mining and knowledge discovery*, **2015**, 29(2): 534-564.
5. Ram P, Gray A G. Maximum inner-product search using cone trees[C]//*Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012: 931-939.
6. Tay Y, Anh Tuan L, Hui S C. Latent relational metric learning via memory-based attention for collaborative ranking[C]//*Proceedings of the 2018 World Wide Web Conference*. 2018: 729-739.
7. Yu J, Gao M, Song Y, et al. Connecting factorization and distance metric learning for social recommendations[C]//*International Conference on Knowledge Science, Engineering and Management*. Springer, Cham, 2017: 389-396.
8. Sánchez-Moreno D, López Batista V, Vicente M, et al. Exploiting the user social context to address neighborhood bias in collaborative filtering music recommender systems[J]. *Information*, **2020**, 11(9): 439.

- 
9. Tran T, Lee K, Liao Y, et al. Regularizing matrix factorization with user and item embeddings for recommendation[C]//Proceedings of the 27th ACM international conference on information and knowledge management. 2018: 687-696.
  10. Qiu J, Tang J, Ma H, et al. DeepInf: Modeling influence locality in large social networks[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'18). 2018.
  11. Ma H, Yang H, Lyu M R, et al. Sorec: social recommendation using probabilistic matrix factorization[C]//Proceedings of the 17th ACM conference on Information and knowledge management. 2008: 931-940.
  12. Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks[C]//Proceedings of the fourth ACM conference on Recommender systems. 2010: 135-142.
  13. Guo G, Zhang J, Yorke-Smith N. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2015, 29(1).
  14. Zhao T, McAuley J, King I. Leveraging social connections to improve personalized ranking for collaborative filtering[C]//Proceedings of the 23rd ACM international conference on conference on information and knowledge management. 2014: 261-270.
  15. Yang L, Jin R. Distance metric learning: A comprehensive survey[J]. Michigan State University, 2006, 2(2): 4.
  16. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//*Advances in neural information processing systems*. **2013**: 3111-3119.
  17. FilmTrust. Available online: <https://guoguibing.github.io/librec/datasets.html>
  18. Douban. Available online: <https://book.douban.com>
  19. Epinions. Available online: <http://www.trustlet.org/epinions.html>