User Bias Drift Social Recommendation Algorithm based on Metric Learning

Jianli Zhao[1]*, Tingting Li[1], Shangcheng Yang[1], Hao Li[1], Baobao Chai[1]
[1]College of Computer Science and Engineering, Shandong University of Science and Technology,
Qingdao 266590, China
[e-mail: jlzhao@sdust.edu.cn]
* Corresponding Author: Jianli Zhao

Received April 9, 2022; revised November 10, 2022; accepted December 1, 2022;
published December 31, 2022

Abstract

Social recommendation algorithm can alleviate data sparsity and cold start problems in recommendation system by integrated social information. Among them, matrix-based decomposition algorithms are the most widely used and studied. Such algorithms use dot product operations to calculate the similarity between users and items, which ignores user's potential preferences, reduces algorithms' recommendation accuracy. This deficiency can be avoided by a metric learning-based social recommendation algorithm, which learns the distance between user embedding vectors and item embedding vectors instead of vector dot-product operations. However, previous works provide no theoretical explanation for its plausibility. Moreover, most works focus on the indirect impact of social friends on user's preferences, ignoring the direct impact on user's rating preferences, which is the influence of user rating preferences. To solve these problems, this study proposes a user bias drift social recommendation algorithm based on metric learning (BDML). The main work of this paper is as follows: (1) the process of introducing metric learning in the social recommendation scenario is introduced in the form of equations, and explained the reason why metric learning can replace the click operation; (2) a new user bias is constructed to simultaneously model the impact of social relationships on user's ratings preferences and user's preferences; Experimental results on two datasets show that the BDML algorithm proposed in this study has better recommendation accuracy compared with other comparison algorithms, and will be able to guarantee the recommendation effect in a more sparse dataset.

Keywords: Artificial Intelligence, Recommender system, Social recommendation, Metric learning, Matrix decomposition.

This study is supported by Natural Science Foundation of Shandong Province (No. ZR2021MF104, ZR2021MF113), National Natural Science Foundation (No.62072288), Key R&D Projects of Qingdao Science and Technology Plan (No.21-1-2-19-xx), Qingdao West Coast New District Science and Technology Plan (No.2020-1-6).

http://doi.org/10.3837/tiis.2022.12.001 **ISSN : 1976-7277**

1. Introduction

People are facing the problem of data overload as the Internet and information technology advance. Recommendation systems have emerged to assist users in quickly and accurately extracting the information they require. In the age of big data, recommender systems, which reduce the negative effects of information overload by providing users with the most attractive and relevant items, have been very successful in social media and e-commerce [1].

Traditional recommender systems are facing the cold starts and the data sparsity problems. The cold start problem occurs when users or items has no historical data associated with them. The data sparsity problem occurs when the quantity of ratings obtained from users is much smaller than the number of ratings needed to build an accurate prediction model [2], and most datasets are sparse, which has a significant impact on the recommendation system's accuracy. The theory of social homogeneity, which holds that users' interests would be similar with their social friends and will be influenced by them, has driven study on social recommendation [3]. To solve the data sparsity and the cold start problems, social information is incorporated into the recommendation process, resulting in social recommendation algorithms [4-5]. For example, the recommendation algorithm based on matrix decomposition [6-8], the recommendation algorithm based on graph, the recommendation algorithm based on clustering [9], etc. Among them, the recommendation algorithm based on matrix decomposition is a popular algorithm in social recommendation because of its advantages such as being flexible, not easily affected by data sparsity, and good interpretability. Jamali et al. [10] considered the trust propagation mechanism, so that user’s features are integrated with the characteristics of the direct neighbor. Ma et al. [11] added a regularization term integrating social relationships to the loss function, in this way can express the restrictions of social relations on users. By mining user preferences hidden in social information, social recommendation algorithms are usable even when the number of user ratings is sparse [12]. Matrix decomposition and its variants have emerged as the dominant technique for constructing recommender systems in collaborative filtering [13]. Matrix decomposition captures known ratings and predicts unknown ratings by dot product between user and item’s feature vectors. In this case, the dot product measures the similarity between user and item feature vectors. However, dot product does not satisfy the required triangular inequality in metric space [14], resulting in the matrix decomposition model's potential vectors not reliably capturing item-item or user-user similarity [15]. As a result, the use of matrix decomposition in recommendation algorithms can limit the accuracy of recommendations. Metric learning avoids the drawbacks of matrix decomposition by constructing distance functions that satisfy triangular inequalities to capture relationships between data, and is widely applied to classification and clustering tasks [16]. For example, Hsieh et al. proposed CML, CML encodes not only user preferences but also user-user and item-item similarities [15]. Compared with matrix decomposition techniques, CML has shown excellent performance in item recommendation, and there are also many studies exploring the combination of metric learning and recommender systems [17-19]. However, these metric learning-based methods usually directly use the distance between embedding vectors of users and items to replace the dot product operation in matrix decomposition, without theoretical demonstration. In addition, the statistical survey results show that user’s ratings are driven by social networking. For users with less social friends and items with less interactive, social influence is stronger. Newer and extremely negative ratings among social friends will have a more significant impact on users [20]. This shows that in practical applications, social relations will not only indirectly affect users' preferences, but also directly affect users' rating decisions [21], and even trigger social learning behavior [22].

However, the existing social recommendation methods only consider the indirect impact of social relations on users’ preferences, while ignoring the direct impact of social relations on users' ratings decisions. This effect is the impact of social friends on users' rating preferences. The data set is analyzed in Section 4.2 of this paper. The results show that user’s friends can impact user’s rating preferences.

This study focuses on the influence of social behavior on a user's rating preferences and thus proposes a User Bias Drift Social Recommendation Algorithm based on Metric Learning (BDML). The main contributions of this study are as follows:

(1) We deduced a formula for predicting user ratings by introducing metric learning, and theoretically explained the principle of using the metric distance between vectors to replace vector multiplication.

(2) We present a novel metric learning-based algorithm, BDML, which constructed a new user bias. On the basis of previous research, we consider the direct impact of social friends on users' rating preferences.

(3) We conduct extensive experiment on two public datasets. The results show that BDML algorithm outperforms baselines algorithms.

The rest of this paper is structured as follows: we briefly introduce related work in section 2. In section 3 we first derive a formula for predicting scores using metric learning, and then describe the BDML algorithm in detail. Experiments and analyses are presented in Section 4. Section 5 presents a summary and outlook.

2. Related work

Matrix decomposition is the most frequently used algorithms for personalized recommendation. In the Netflix competition, Simon Funk [13] was the first to use matrix decomposition in the recommendation domain and rating prediction, later researches improved MF. For example, Koren [23] introduced user and item bias to model user and item particular features and introduced item implicit feedback to improve the accuracy of recommendation algorithm. [4,24] et al. incorporated the variability of user preferences on the basis of matrix decomposition, but these algorithms could not work for the cold start problem.

For the cold start problem, social information is introduced into social recommendation to improve the recommendation accuracy of the algorithm. Y et al. [13] consider interactions between users and items based on matrix decomposition, and generate recommendations through user preference characteristics and item neighbor relationships. Ma et al. [25] combined users’ social network relationships with a rating matrix by sharing the same user feature space between social relationships and rating data. And then, they proposed RSTE [26], a social trust integration algorithm that linearly integrates matrix decomposition with a trust-based neighborhood model, considering only the user's preferences but also their friends' preferences in rating prediction for target users. Although the RSTE algorithm uses social information to alleviate the cold start problem, it only considers the influence of friends' preferences on the user's preferences without taking trust propagation into account. Jamali et al. [10] fused a trust propagation mechanism based on social recommendation and proposed a new social algorithm, SocialMF. Zhang et al. [3] fused the social relationship matrix and rating matrix into a large matrix, and based on the objective function of matrix decomposition, two new social regularization terms are added. In this way, the model can perceive the impact of social relations on user preferences. The above algorithm introduces social information into the matrix decomposition algorithm, which alleviates the cold start problem in recommendation systems and improves recommendation accuracy, but dot product operation in matrix decomposition does not satisfy the triangular inequality [14], which limits the ability of matrix decomposition to express user preferences [27]. Metric learning has been introduced into the recommendation domain to address this issue. Li et al. [28] directly used metric learning to replace the dot product of user and item features, and thus proposed the SREE algorithm to integrate user similarity information when predicting ratings. The predictive rating formula is as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i - \alpha \left(p_u - q_i \right) \left(p_u - q_i \right)' - \beta \sum_{v \in N(u)} S_{uv} \left(p_u - p_v \right) \left(p_u - p_v \right)' \tag{1}$$

where $\hat{r}_{ui}$ denotes prediction score, μ denotes global bias, b_u denotes the bias of the user u, b_i is the bias of the item i, p_u and q_i are the feature vectors of the user u and item i ,respectively. The set of friends of user u is denoted by $N(u)$, and the similarity between user u and user v is denoted by S_{uv}. Hao Wu et al. [14] proposed the CRML algorithm, which is based on CML and treats the optimization problem as a multitask learning problem with a primary task for optimal metric learning and two auxiliary tasks for representation learning, while using soft parameter sharing to pass parameters between the primary and auxiliary tasks.

However, existing social recommendation algorithms ignore the direct influence of social friend on users' rating decision, the user's rating preference. In addition, many studies have shown that metric learning-based recommendation algorithms can solve the problem, the dot product operation can affect the expression of potential user preferences, which reduces the recommendation accuracy of the algorithm. Hence, this study employs metric learning to investigate the direct of social behavior on users' rating preferences to improve prediction accuracy.

3. Method

In previous sections, we discussed social recommendation algorithms related to metric learning, and in this section, we present a User Bias Drift Social Recommendation Algorithm based on Metric Learning (BDML). To obtain user and item feature vectors via metric learning, we first map ratings to metric distances by using the metric method:

$$D_{ui} = r_{max} - r_{ui} \tag{2}$$

The r_{max} in the formula denotes the highest score that a user can give to an item in a system or dataset (if a system is a 0-5 rating system, r_{max} is 5), and r_{ui} denotes the rating of user u for item i.In this study, we use the Euclidean distance (or l_2-norm distance) as the prediction distance between feature vectors, P_u denotes the feature vector of user u and Q_i denotes the feature vector of item i.That is: $D_{ui} = \| P_u - Q_i \|_2$.Actually, squared distance is usually adopted to avoid the trouble of computing the square root [29]. As a result, the prediction distance $\hat{D}_{ui} = \| P_u - Q_i \|_2^2$.Through the above conversion, when the more user u likes item i, the smaller the prediction distance between the feature vectors of user u and item i.

The new prediction distance can be obtained by adding bias information to the prediction distance in the manner of BiasMF [13].

$$\hat{D}_{ui} = \mu_d + b_{ud} + b_{id} + \| P_u - Q_i \|_2^2 \tag{3}$$

where μ_d denotes the global distance bias, which is equal to the average distance of the data set, b_{ud} denotes the users ' distance bias, and b_{id} denotes the items' distance bias. Based on Eq. (2) and Eq. (3), the prediction rating formula $\hat{r}_{ui} = r_{max} - \hat{D}_{ui}$ can be obtained:

$$\hat{r}_{ui} = \mu_d - b_{ud} - b_{id} - \| P_u - Q_i \|_2^2 \tag{4}$$

In Eq.(4), $\mu_d = \frac{\sum_{u,i}(r_{max} - r_{ui})}{\| R \|} = \frac{\sum_{u,i} r_{max} - \sum_{u,i} r_{ui}}{\| R \|}$, where $\| R \|$ denotes the total number of ratings in training set and r_{max} is a constant, so the global distance bias can be written as:

$$\mu_d = \frac{\| R \| r_{\max} - \frac{\sum_{u,i} r_{ui}}{\| R \|}}{\| R \|} = r_{\max} - \frac{\sum_{u,i} r_{ui}}{\| R \|} \tag{5}$$

Substituting Eq. (5) into Eq. (4) and denoting the training set mean rating $\frac{\sum_{u,i} r_{ui}}{\| R \|}$ by μ, $-b_{ud}$ by b_u and $-b_{id}$ by b_i, Eq. (4) can be written as:

$$\hat{r}_{ui} = \mu + b_u + b_i - \| P_u - Q_i \|_2^2 \tag{6}$$

The analysis of the dataset in Section 4.2 shows that user's rating preferences will change during socialization due to the influence of their friends, and user's rating preferences will tend to converge with their friends' rating preferences, but since the users themselves are not identical, the target users will retain their own characteristics even after their rating preferences have drifted. As a result, in this section, we made a new definition of user bias is provided, as follows:

$$b_{ud} = b_{od} + b_{dd} \tag{7}$$

In Eq. (7), b_{ud} represents the user's distance bias (User distance bias), b_{od} represents the user's distance bias due to its own characteristics (Original distance bias), and b_{dd} indicates the drift of the distance bias due to the influence of friends (Distance bias drift).

We get the predicted scores by substituting Eq. (7) into Eq. (4):

$$\hat{r}_{ui} = r_{\max} \mu_d - b_{od} - b_{dd} - b_{id} - \| P_u - Q_i \|_2^2 \tag{8}$$

Use b_u to denote $-b_{od}$, b'_u to denote $-b_{dd}$, and other expressions are the same as Eq. (6), So the new prediction rating formula is as follows:

$$\hat{r}_{ui} = \mu + b_u + b'_u + b_i - \| P_u - Q_i \|_2^2 \tag{9}$$

Then the objective function can be written as follows:

$$L = \frac{1}{2} \sum_{u,i \in R} \left(r_{ui} - \left(\mu + b_u + b'_u + b_i - \left\| P_u - Q_i \right\|_2^2 \right) \right)^2 + \lambda_p \left\| P_u \right\|_2 + \lambda_q \left\| Q_i \right\|_2 + \lambda_b \left(b_u^2 + b_u'^2 + b_i^2 \right) + \sum_{u,v \in T} \left(b'_u - b'_v \right)^2 \tag{10}$$

The variables in public Eq. (10) are the same as in the previous section, and the parameters in Eq, where T is the set of users, b'_v is the drift of the distance bias of user v due to the influence of friends. The parameters in Eq. (10) are obtained by means of stochastic gradient descent, and the iterative formula is obtained as follows:

$$\begin{gathered} p_u = p_u - \eta\left(e_{ui} \cdot \left(p_u - q_i\right) + \lambda_p p_u\right) \\ q_i = q_i + \eta\left(e_{ui} \cdot \left(p_u - q_i\right) - \lambda_q q_i\right) \\ b_u = b_u + \eta\left(e_{ui} - \lambda_b b_u\right) \\ b_i = b_i + \eta\left(e_{ui} - \lambda_b b_i\right) \\ b_{ud} = b_{ud} + \eta\left(e_{ui} - \lambda_b b_{ud} + \sum\nolimits_{v \in T_u}\left(b_{ud} - b_{vd}\right)\right) \end{gathered} \tag{11}$$

Where e_{ui} denotes the difference between the true score and the predicted score, calculated as: $e_{ui} = r_{ui} - \hat{r}_{ui}$, and η denotes the learning rate.

The flow of the User Bias Drift Social Recommendation Algorithm based on Metric Learning is shown in **Table 1**.

Table 1. BDML algorithm Flow

```
Algorithm1 User Bias Drift Social Recommendation Algorithm based on Metric Learning
Input: train set $R$, regularization parameter $\lambda$, learning rate $\eta$
Output: global distance bias $\mu$, user feature matrix $P$, item feature matrix $Q$, user bias $b_u$, item bias $b_i$, item bias drift $b'_u$
Calculate $\mu$ by $R$
Initialize $P$、$Q$、$b_u$、$b_i$、$b'_u$ with random variables
Iteration:
  for $r_{ui}$ in $R$ do
  Update parameters according to Eq. (11):
      $P_u \leftarrow P_u - \eta \frac{dL}{dP_u}$
      $Q_i \leftarrow Q_i - \eta \frac{dL}{dQ_i}$
      $b_u \leftarrow b_u - \eta \frac{dL}{db_u}$
      $b_i \leftarrow b_i - \eta \frac{dL}{db_i}$
      $b'_u \leftarrow b'_u - \eta \frac{dL}{db'_u}$
   until converge
print $P$, $Q$, $b_u$, $b_i$, $b'_u$
```

4. Experiment

In order to evaluate the performance of the algorithms in this study, the experimental phase of this study used two publicly available datasets containing user rating information and social relationship information, FilmTrust and Ciao, to test different algorithms, and the social relationships in the datasets were all one-way social trust relationships. All experiments were performed on a PC equipped with an Intel(R) Core (TM) i5-8500 CPU processor and 8 GB RAM, running at 3.00 GHz.

4.1 Dataset

The FilmTrust dataset (http://www.librec.net/datasets.html) was crawled by Guo et al. from the FilmTrust website [24]. FilmTrust is a social trust relationship-based movie recommendation website that allows users to rate movies based on their preferences while building one-way social trust relationships. The Ciao dataset (http://www.jiliang.xyz/trust.html) was collected by Tang et al. on the item review website Ciao [30]. In the Ciao dataset, users can not only rate and add reviews to different items but also build one-way social trust relationships with other users.

The range of scores varies from dataset to dataset. FilmTrust has a scoring range of 0.5-4 with a step size of 0.5, while Ciao has a scoring range of 1-5 with a step size of 1. **Table 2** displays detailed statistics for both datasets.

Table 2. Descriptive Statistics for the Dataset

Dataset	Users	Items	Ratings	Density	Social relationship	Social relationship sparsity
FilmTrust	1641	2071	35497	1.044%	1853	0.069%
Ciao	7375	106997	284086	0.036%	111781	0.205%

4.2 Dataset Analysis

There are 1,641 users in the FilmTrust dataset, 609 of whom are socially engaged and each of whom follow a maximum of 59 other users. The Ciao dataset contains 7375 users, 1437 of whom have performed social behaviors and have a maximum of 100 social friends.

To observe the impact of social behavior on user ratings, this section counts the average rating of users with social behavior, the global average rating of dataset, the difference between the average rating of example users and their friends, and the difference between the average rating of example users and their non-friends, denoted by f1, f2, f3, and f4, respectively, and the statistical results are shown in **Fig. 1**.

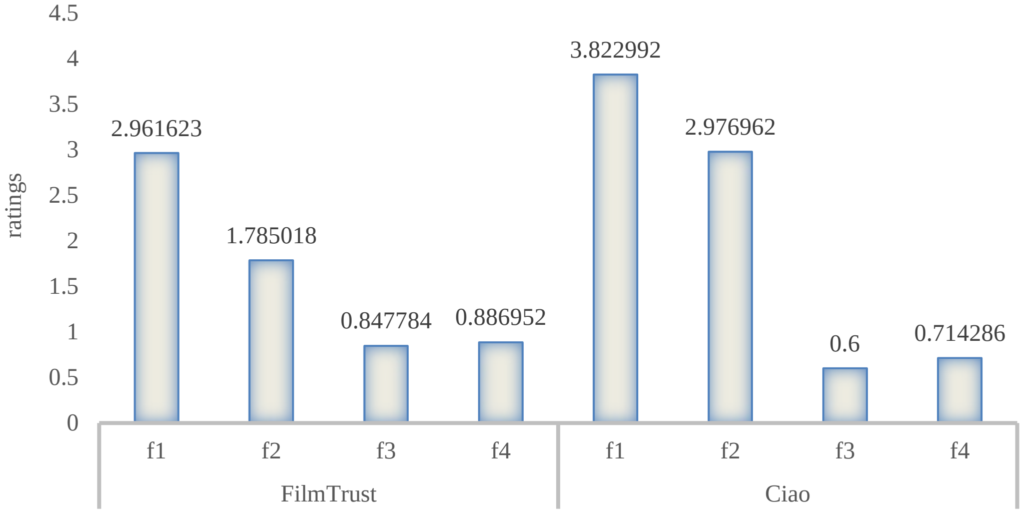


Fig. 1. User Rating Statistics

The average user rating in the personalized recommendation algorithm can reflect the user rating preference to some extent, and this preference is defined as user bias in the rating formula. As shown in **Fig. 1**, the average rating of users with social behavior in the FilmTrust dataset is 2.96, while the global average rating is 1.78. In the FilmTrust dataset, the average rating of users' social behavior is 65.9 percent higher than the global average rating. It implies that a user's rating preferences are influenced by their friends. In the Ciao dataset, the average rating of users with social behavior is 28.4 percent higher than the global average rating, indicating that users' rating preferences are also influenced by their social friends in the Ciao dataset.

This section selects a specific user (hereafter referred to as the sample user) in each of the two datasets and counts the rating data for the sample user to more clearly observe the impact of social behavior on user rating bias. The difference between the average rating of the sample users and their non-friends in the Ciao dataset is 0.71, but the distinction with their friends is 0.6. The difference between the average ratings of the sample users and their non-friends is 19% greater than the difference between their friends' average ratings. It indicates that the average rating of users in the Ciao dataset is more similar with their friends, the user's rating preferences are closer to those of their friends. In the FilmTrust dataset, the difference between the sample user's average rating and non-friends is 4.6% higher than the difference between the average rating of friends, which is small but still indicates that the user's rating preferences are also influenced by social behavior in the FilmTrust dataset.

4.3 Experimental Settings

4.3.1 Evaluation Metrics

Since the algorithm in this study is aimed at rating prediction, the experiment's evaluation criteria used two commonly used evaluation indicators in the recommendation system to measure rating prediction: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). The lower the RMSE and MAE values, the higher recommended accuracy is. The RMSE and MAE formulas are as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{u,i\in R}\left(r_{ui} - \hat{r}_{ui}\right)^2}{T}} \tag{12}$$

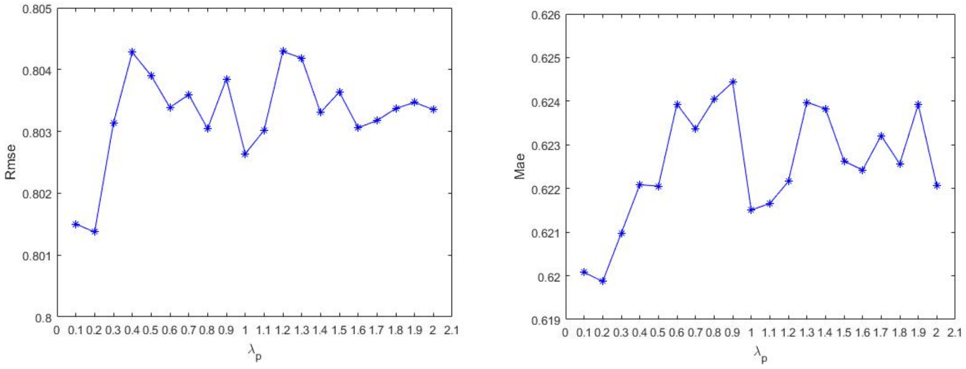
$$\text{MAE} = \frac{\sum_{u,i\in R}\left|r_{ui} - \hat{r}_{ui}\right|}{T} \tag{13}$$

where R denotes the test set, T is the total number of ratings in the test set, r_{ui} is the true rating, and $\hat{r}_{ui}$ is the predicted rating.

In addition, to ensure the statistical significance of our experiment, we use the Paired Samples Wilcoxon Test to test the significant level of the difference between the predicted results by the algorithms and the actual results. When $P < 0.05$, the predicted ratings of the model and the true ratings are considered to be significantly different, and when $P > 0.05$, the predicted ratings of the model and the true ratings are considered to be no significant difference. The larger the P value, the smaller the difference between the predicted ratings and the true ratings.

4.3.2 Parameter Settings

The BDML algorithm has four main parameters: user feature regularization parameter λ_p, item feature regularization parameter λ_q, bias regularization parameter λ_b, and feature vector dimension d .This section demonstrates how parameter changes affect the experimental results in the FilmTrust dataset. First, changed the parameter λ_p, and **Fig. 2** shows how the RMSE and MAE change as the parameters are changed.

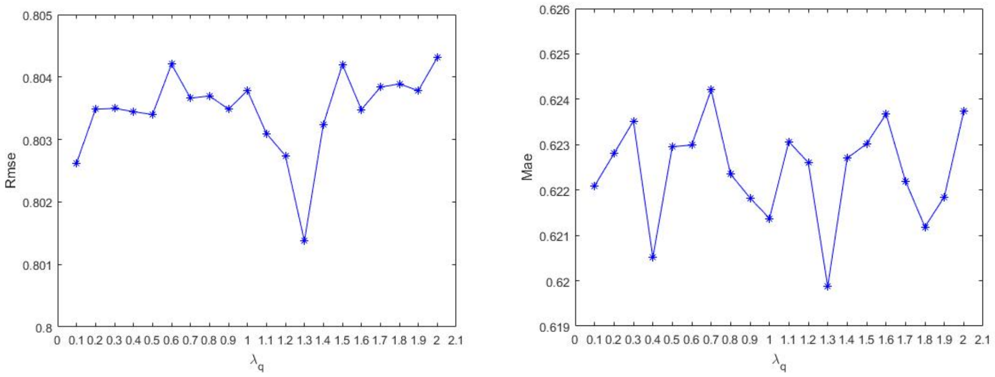


(a) (b)

Fig. 2. Impact of λ_p on Prediction Accuracy

As the parameter λ_p changes, the RMSE trend is shown in **Fig. 2**(a), and **Fig. 2**(b) depicts the MAE trend as the parameter λ_p is changed. The figure shows that when λ_p is set to 0.2, the optimal values of RMSE and MAE are obtained simultaneously, and the overall trend of both RMSE and MAE increases as the parameter is increased, so λ_p is set to 0.2.

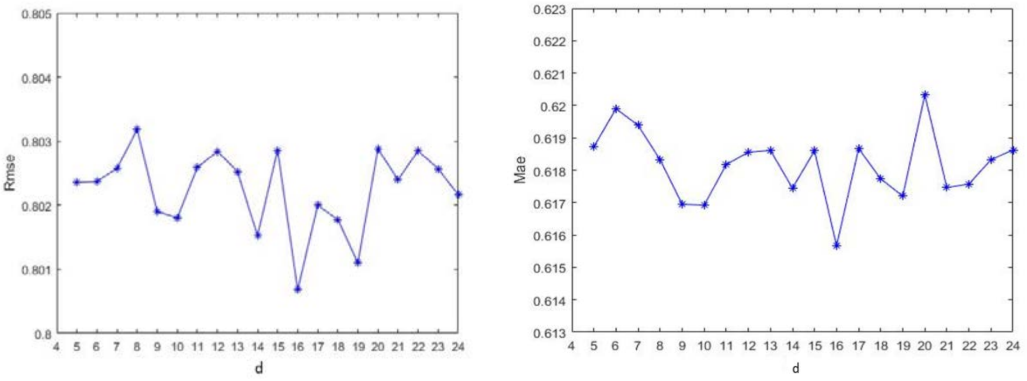
When adjusting the parameter λ_q, the RMS and MAE change with the parameter as shown in **Fig. 3**.



(a) (b)

Fig. 3. Impact of λ_q on Prediction Accuracy

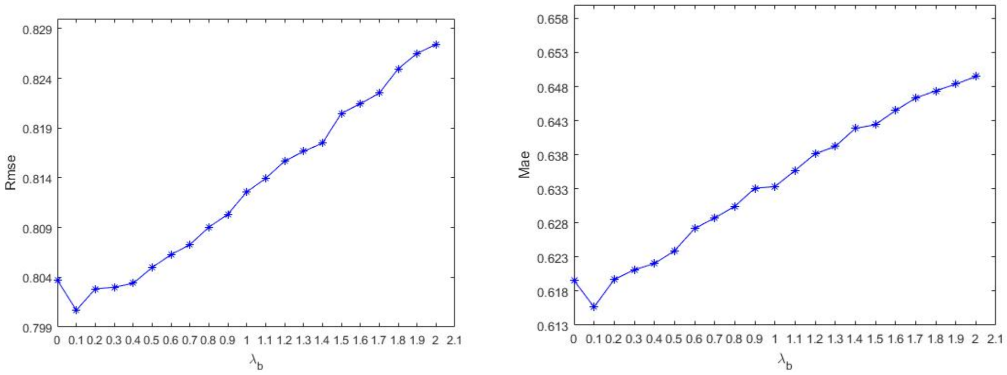
Fig. 3(a) shows the RMSE trend as the parameter λ_q varies, while **Fig. 3**(b) depicts the MAE trend as the parameter λ_q is changed. The figure shows that when λ_q is set to 1.3, both RMSE and MAE achieve higher values, so that value of λ_q is set to 1.3.



(a) (b)

Fig. 4. Impact of Embedding Size d on Prediction Accuracy

Fig. 4(a) depicts the degree of influence of various feature dimensions *d* on RMSE, while **Fig. 4**(b) depicts the influence of various feature dimensions *d* on MAE. The figure shows that when the feature dimension *d* is set to 16, both RMSE and MAE achieve better values, and as the feature dimension increases, the RMSE value fluctuates but the overall trend is higher, so the feature dimension is set to 16.



（a） （b）

Fig. 5. Impact of λ_b on Prediction Accuracy

Fig. 5(a) shows the impact of various λ_b on RMSE, while **Fig. 5**(b) shows the impact of various λ_b on MAE. The figure shows that the model can achieve a better value when λ_b is set to 0.1, and then both RMSE and MAE gradually increase as λ_b increases, so λ_b is set to 0.1.

4.4 Baseline Algorithms

In this study, the BDML algorithm is proposed based on metric learning while taking social information into account. Therefore, we select personalized recommendation algorithms based on metric learning and some social recommendation algorithms as baselines. Their brief descriptions of them are listed as follows， and the specific parameter settings of comparison algorithms are shown in **Table 3**.

SocialMF [10]: Social Matrix Decomposition (SocialMF) is a social recommendation algorithm with classical user feature re-representation. The algorithm considers that user preferences depend on the preferences of friends, and therefore re-represents user features by the features of friends.

SoReg [11]: SoReg’s concept is similar with SocialMF in that it assumes that user characteristics should be similar with their friends, and thus uses their fends' preference information to influence the user's final rating.

SREE [28]: Social Recommendation model using Euclidean Embedding (SREE) is a social recommendation algorithm based on metric learning that considers social information and uses it to re-represent user features.

TrustPMF [3]: TrustPMF considers both the influence of the trustor and the influence of trustees on user characteristics.

Table 3. The specific parameter settings of the competing algorithms

Algorithms	**Parameter Settings**	
	FilmTrust	Ciao
SocialMF	$\lambda_p = 0.05, \lambda_q = 0.05, d = 5$	$\lambda_p = 0.3, \lambda_q = 0.3, d = 9$
SoReg	$\lambda_p = 0.02, \lambda_q = 0.02, d = 10$	$\lambda_p = 0.2, \lambda_q = 0.2, d = 10$
SREE	$\lambda_p = 1.4, \lambda_q = 1.2, d = 10$	$\lambda_p = 0.2, \lambda_q = 0.2, d = 10$
TrustPMF	$\beta_1 = 10, \beta_2 = 10, d = 10$	$\beta_1 = 5, \beta_2 = 5, d = 10$

The special parameter β_1 of the TrustPMF algorithm in **Table 3** controls the similarity between user characteristics and trustee characteristics, β_2 controls the similarity between user characteristics and trustee characteristics, and the other parameters are consistent with the above meanings.

4.5 Experimental Results

Experiments are carried out on two real datasets in this study to demonstrate the efficacy of the proposed method. Throughout the experiments, the datasets were divided into 8:2 training and test sets. **Table 4**, **Fig. 6** and **Fig. 7** show the experimental results on two datasets, FilmTrust and Ciao.

Table 4. Comparison with different algorithms on FilmTrust and Ciao datasets

Dataset	Algorithm	RMSE	MAE	P
FilmTrust	SocialMF	0.8399	0.6373	1.612×10^{-9}
	SREE	0.8152	0.6346	0.061
	TrustPMF	0.8091	0.6327	0.027
	SoReg	0.8923	0.6706	9.065×10^{-8}
	BDML	0.7980	0.6158	0.2602
Ciao	SocialMF	1.0216	0.7903	3.074×10^{-93}
	SREE	0.9876	0.7560	6.552×10^{-28}
	TrustPMF	1.0079	0.7743	3.326×10^{-35}
	SoReg	1.0388	0.7996	3.942×10^{-62}
	BDML	0.9813	0.7414	5.243×10^{-5}

Table 4 shows numerical data from the experimental results, demonstrating three evaluation metrics, RMSE MAE and p, of various algorithms on two real datasets, FilmTrust and Ciao. The results of statistical experiments in FilmTrust dataset show that the P value of BDML algorithm is greater than 0.05. Although the P value is less than 0.05 on the Ciao dataset, but the BDML model had the highest P value and the best performance.

This section graphs the numerical data in **Table 4** to form **Fig. 6** and **Fig. 7**, and BDML algorithm is labeled with an orange grid in **Fig. 6** and **Fig. 7**. This allows for a clear comparison of the experimental results. **Fig. 6** compares the RMSE and MAE of various algorithms in the FilmTrust dataset, and **Fig. 7** compares the performance of various algorithms in the Ciao dataset.

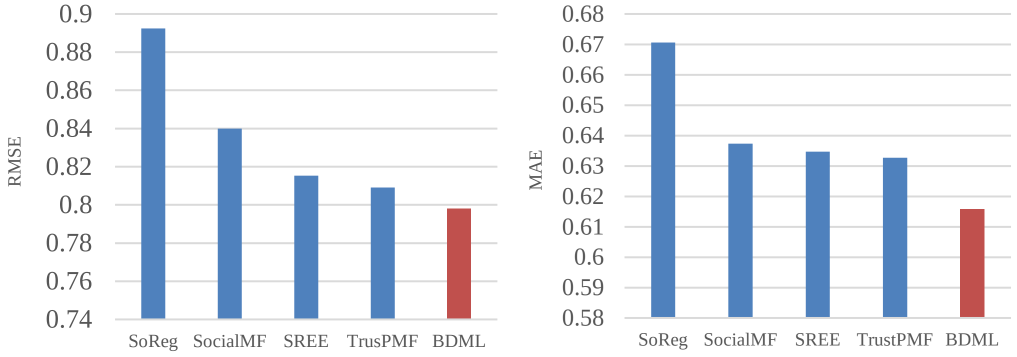


Fig. 6. Comparison with different algorithms on FilmTrust

All three methods, SREE, SoReg, and SocialMF use social information for the re-representation of user features, but as seen in **Table 4** and **Fig. 6**, SREE improves the RMSE metrics by 8.6% and 2.9%, respectively, over the two classical algorithms, SoReg and SocialMF, in the FilmTrust dataset, which indicates that the metric-based learning of user feature re-representation of social recommendation algorithms can improve the recommendation accuracy of social recommendation algorithms. However, the SREE recommendation is slightly less effective than TrustPMF, because TrustPMF considers both the influence of trusted friends on user characteristics and the influence of trusted friends on user characteristics.

In terms of RMSE metrics, the BDML algorithm proposed in this study reduces 10.5 percent versus the SoReg algorithm, 4.9 percent versus the SocialMF, 2.1 percent versus the SREE algorithm, and 1.4 percent versus the TrustPMF; In terms of MAE metrics, the BDML algorithm reduces 8.1 percent, 3.1 percent, 3.0 percent, and 2.6 percent, respectively, when compared to SoReg, SocialMF, SREE, and TrustPMF. The BDML algorithm achieves better improvement in RMSE and MAE metrics when compared to other algorithms, indicating that the BDML algorithm considers user’s rating preference influenced by friends based on metric learning and can improve prediction accuracy.

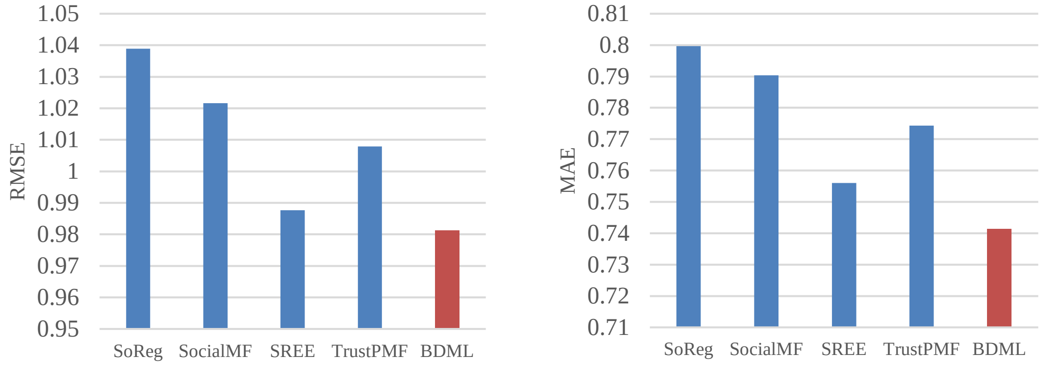


Fig. 7. Comparison with different algorithms on Ciao

As seen in **Table 4** and **Fig. 7**, the comparison trends of the three methods, SREE, SoReg, and SocialMF, are the same as in the FilmTrust dataset, indicating that metric learning in the

Ciao dataset can still be useful in social recommendation algorithms for user feature representation. The TrustPMF effect in the Ciao dataset is worse than in the FilmTrust dataset because the Ciao dataset is sparser, and the sparsity of the dataset has a greater impact on recommendation accuracy. In the Ciao dataset, SREE and BDML algorithms still outperform TrustPMF, indicating that the metric learning-based social recommendation algorithm can still provide users with more accurate recommendations even in sparse datasets.

In terms of RMSE metrics for the Ciao dataset, the BDML algorithm proposed in this chapter reduces 5.54 percent versus the SoReg algorithm, 3.94 percent versus the SocialMF algorithm, 0.6 percent versus the SREE algorithm, and 2.6 percent versus the TrustPMF algorithm, respectively. In terms of MAE metrics, the BDML algorithm reduces 7.3 percent, 6.2 percent, 2.0 percent, and 4.2 percent, respectively, when compared to SoReg, SocialMF, SREE, and TrustPMF, indicating that the BDML algorithm can improve prediction accuracy by considering the impact of social relationships on user rating decisions, and it can also adapt to sparser datasets.

5. Conclusion

Traditional social recommendation learning is mostly based on matrix decomposition, and the dot product operation in matrix decomposition makes recommendation by learning the similarity between user features and item features. This causes the algorithm to ignore the user's fine-grained preferences for items. Social recommendation algorithm based on metric learning, using the distance of users and items feature vector in feature space, to replace the dot product operation, alleviate this problem. We first explain this process in the form of a formula derivation, thus filling the theoretical gap in the application of metric learning in recommender systems. In addition, social relationships not only have an indirect impact on users' preferences, but also have a direct impact on users' rating preferences. We use the new user bias term to study the direct impact of social relationships on users' rating preferences. The proposed BDML algorithm outperforms other baseline algorithms in recommendation.

As people's living standard improves and they are exposed to more and more things, users' preferences will change accordingly. How to capture the changes in users' preferences and thus make real-time recommendations is a problem we need to solve urgently. In the future, we will try to make real-time social recommendations.

References

[1] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, and L. Sun, “Dual-regularized matrix factorization with deep neural networks for recommender systems,” *Knowledge-Based Systems*, vol. 145, pp. 46–58, 2018. Article (CrossRef Link)

[2] J. Shokeen and C. Rana, “A review on the dynamics of social recommender systems,” *International Journal of Web Engineering and Technology*, vol. 13, no. 3, 2018. Article (CrossRef Link)

[3] T. Zhang, W. Li, L. Wang, and J. Yang, “Social recommendation algorithm based on stochastic gradient matrix decomposition in social network,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 601-608, 2020. Article (CrossRef Link)

[4] J. Tang, X. Hu, and H. Liu, “Social recommendation: a review,” *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013. Article (CrossRef Link)

[5] Q. Zhang, J. Wu, H. Yang, W. Lu, G. Long, and C. Zhang, “Global and local influence-based social recommendation,” in *Proc. of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1917–1920, 2016. Article (CrossRef Link)

[6] B. Yang, Y. Lei, J. Liu, and W. Li, “Social collaborative filtering by trust,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017. Article (CrossRef Link)

[7] H. Tahmasbi, M. Jalali, and H. Shakeri, “Modeling user preference dynamics with coupled tensor factorization for social media recommendation,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 9693-9712, 2021. Article (CrossRef Link)

[8] A. Pujahari and D. S. Sisodia, “Pair-wise Preference Relation based Probabilistic Matrix Factorization for Collaborative Filtering in Recommender System,” *Knowledge-Based Systems*, vol. 196, pp. 1-12, 2020. Article (CrossRef Link)

[9] Y. Yang, H. Yao, R. Li, and S. Wang, “A collaborative filtering recommendation algorithm based on user clustering with preference types,” *Journal of Physics: Conference Series*, vol. 1848, no. 1, pp. 12043-12047, 2021. Article (CrossRef Link)

[10] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in *Proc. of the fourth ACM conference on Recommender systems*, pp. 135–142, 2010. Article (CrossRef Link)

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, “Recommender systems with social regularization,” in *Proc. of the fourth ACM international conference on Web search and data mining*, pp. 287–296, 2011. Article (CrossRef Link)

[12] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005. Article (CrossRef Link)

[13] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” *Computer*, vol. 42, no. 8, pp. 30–37, 2009. Article (CrossRef Link)

[14] H. Wu, Q. Zhou, R. Nie, and J. Cao, “Effective metric learning with co-occurrence embedding for collaborative recommendations,” *Neural Networks*, vol. 124, pp. 308–318, 2020. Article (CrossRef Link)

[15] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, “Collaborative metric learning,” in *Proc. of the 26th international conference on world wide web*, pp. 193–201, 2017. Article (CrossRef Link)

[16] F. Wang and J. Sun, “Survey on distance metric learning and dimensionality reduction in data mining,” *Data mining and knowledge discovery*, vol. 29, no. 2, pp. 534–564, 2015. Article (CrossRef Link)

[17] Y. Tay, L. Anh Tuan, and S. C. Hui, “Latent relational metric learning via memory-based attention for collaborative ranking,” in *Proc. of the 2018 world wide web conference*, pp. 729–739, 2018. Article (CrossRef Link)

[18] J. Yu, M. Gao, W. Rong, Y. Song, Q. Fang, and Q. Xiong, “Make users and preferred items closer: Recommendation via distance metric learning,” in *Proc. of International Conference on Neural Information Processing*, Springer, pp. 297–305, 2017. Article (CrossRef Link)

[19] S. Zhang, L. Yao, Y. Tay, X. Xu, X. Zhang, and L. Zhu, “Metric factorization: Recommendation beyond matrix factorization,” *arXiv preprint arXiv:1802.04606*, 2018. Article (CrossRef Link)

[20] C& A. Wang, X& M. Zhang, and I. Hann, “Social Nudged: A Quasi Empirical Study of Friends' Social Influence in Online Product Ratings,” *Information Systems Research*, vol. 29, no. 3, 2018. Article (CrossRef Link)

[21] T. Xu, H. Zhong, H. Zhu, H. Xiong, E. Chen and G. Liu, “Exploring the impact of dynamic mutual influence on social event participation,” *SDM*, pp. 262−270, 2015. Article (CrossRef Link)

[22] T. Xu, H. Zhu, X. Zhao, Q. Liu, H. Zhong, E. Chen and H. Xiong, “Taxi driving behavior analysis in latent vehicle-to-vehicle networks: A social influence perspective,” in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1285−1294, 2016. Article (CrossRef Link)

[23] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” in *Proc. of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426–434, 2008. Article (CrossRef Link)

[24] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," *IJCAI*, vol. 13, pp. 2619–2625, 2013. Article (CrossRef Link)

[25] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proc. of the 17th ACM conference on Information and knowledge management*, pp. 931–940, 2008. Article (CrossRef Link)

[26] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 203–210, 2009. Article (CrossRef Link)

[27] C. Ma, L. Ma, Y. Zhang, R. Tang, X. Liu, and M. Coates, "Probabilistic metric learning with adaptive margin for top-k recommendation," in *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1036–1044, 2020. Article (CrossRef Link)

[28] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, R. Jia, and T. Dou, "Social recommendation using euclidean embedding," in *Proc. of 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 589–595, 2017. Article (CrossRef Link)

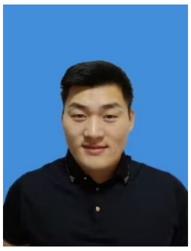
[29] S. Zhang, L. Yao, Y. Tay, X. Xu, X. Zhang, and L. Zhu, "Metric Factorization: Recommendation beyond Matrix Factorization," *arXiv: Information Retrieval*, 2018. Article (CrossRef Link)

[30] J. Tang, H. Gao, and H. Liu, "mtrust: Discerning multi-faceted trust in a connected world," in *Proc. of the fifth ACM international conference on Web search and data mining*, pp. 93–102, 2012. Article (CrossRef Link)

Jianli Zhao received his PhD degree in 2006 from the Department of Computer Application Technology, Northeastern University, China. In 2011, He served as an associate professor in the College of Computer Science and Engineering, Shandong University of Science and Technology, Major Research Direction in pervasive computing, personalized recommendation, indoor location.

Tingting Li received her B.E. degree in College of Computer Science and Engineering from Shandong University of Science and Technology of Taishan College of Science and Technology, Taian, China, in 2019.

Shangcheng Yang received his Master's degree in College of Computer Science and Engineering from Shandong University of Science and Technology, Qingdao, China, in 2021.



Hao Li received his Master's degree in Computer Science and Engineering from Shandong University of Science and Technology in Qingdao, China, in 2021. He is now working on his PhD, focusing on tensor decomposition and recommender systems.

Baobao Chai is now pursuing his Ph.D. degree in the School of Computer Science and Engineering at Shandong University of Science and Technology, Qingdao, China. His research interests include federated learning, traffic prediction and blockchain.