User Embedding for Rating Prediction in SVD++-Based Collaborative Filtering

Wenchuan Shi, Liejun Wang and Jiwei Qin *

School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; wechshi@163.com (W.S.); wljxju@xju.edu.cn (L.W.)

* Correspondence: jwqin_xju@163.com

Abstract: The collaborative filtering algorithm based on the singular value decomposition plus plus (SVD++) model employs the linear interactions between the latent features of users and items to predict the rating in the recommendation systems. Aiming to further enrich the user model with explicit feedback, this paper proposes a user embedding model for rating prediction in SVD++-based collaborative filtering, named UE-SVD++. We exploit the user potential explicit feedback from the rating data and construct the user embedding matrix by the proposed user-wise mutual information values. In addition, the user embedding matrix is added to the existing user bias and implicit parameters in the SVD++ to increase the accuracy of the user modeling. Through extensive studies on four different datasets, we found that the rating prediction performance of the UE-SVD++ model is improved compared with other models, and the proposed model's evaluation indicators root-mean-square error (RMSE) and mean absolute error (MAE) are decreased by 1.002–2.110% and 1.182–1.742%, respectively.

Keywords: recommendation system; rating prediction; SVD++; user embedding

1. Introduction

The past decades witnessed strong growth of internet use worldwide. The amount of information on the internet is increasing explosively, giving rise to the problem of "information overload". Recommendation systems help users automatically retrieve and filter information or services required by users from the massive available data. For a long time, recommendation systems recommended items in which a particular user may be interested from user–item interaction information such as ratings. Under these circumstances, the accuracy of the rating prediction directly affects the recommended effect.

In recent years, the collaborative filtering (CF) algorithms [1–4] was one of the most widely used and classic technologies for rating prediction in recommendation systems, among which the matrix factorization (MF) model [2,5,6] was the most commonly used technique. The MF model expresses the rating data in a matrix form to mine the low-dimensional implicit feature space. The correlation is characterized by the inner product between the user and item vectors. However, the rating data have the characteristics of high sparsity and uneven distribution, leading to several problems such as low recommendation performance. To solve this problem, researchers introduced additional information about the user or item into the MF model, and they obtained good recommendation results to some extent. For example, the FUNK-SVD (A model proposed in the context of the Netflix Prize by Simon Funk) model decomposed the sparse and high-dimensional rating matrix into two low-dimensional matrices and predicted the rating using the reconstructed low-dimensional matrices. Then, the BIAS-SVD [7] model not only considered the user-independent or item-independent factors as the biased part, but also considered the interaction between the user and item as a personalized

part, and it considered that the bias component of the MF model plays a more essential role in improving the accuracy of the rating prediction than the personalized component. The singular value decomposition plus plus (SVD++) model [7] added the implicit feedback to the users based on the BIAS-SVD. Guo et al. [8] considered the user's explicit trust relationship and the item's rating as the implicit feedback information, and they added the social feedback information into the original SVD++ model to reconstruct the TRUST-SVD model of the rating prediction.

Clearly, the bias and the implicit feedback of the user are considered to model the user features, while the explicit feedback of the user in the rating is ignored in the existing traditional SVD++ model. Therefore, the development of an effective method for further enriching the user features and improving the rating prediction precision is a key issue for the future improvement of recommendation systems. In this paper, we propose a novel rating prediction model based on user embedding and SVD++ named UE-SVD++. The UE-SVD++ model exploits the user potential correlation from the rating and constructs the user embedding matrix to enrich the user modeling. To build the user embedding matrix, we primarily calculate and extract a collection of strongly favored users for each item in the dataset. Inspired by Adomavicius et al. [9,10], we define a user with a rating of greater than 70% of the highest rating on the given item as the favored user of this item. Then, we calculate the User-wise Mutual Information (UMI) values to obtain the relationship between different users, and derive the user embedding matrix according to the set UMI threshold k. Finally, the user embedding matrix is added to the existing user bias and implicit parameters in the SVD++ to model the user features more accurately on rating prediction. The experimental results on four real-world datasets demonstrate that the UE-SVD++ model outperforms the state-of-the-art rating prediction models.

The rest of this paper is organized as follows: we review the related work in Section 2. Then, we describe the design and construction of the UE-SVD++ model in Section 3. The selection of the optimal model parameters, and the performance comparison and discussion of the UE-SVD++ model with other models using extensive experiments are described in Section 4. Finally, we conclude this paper in Section 5.

2. Related Work

The MF model received extensive attention as a popular and effective rating prediction method in recommendation systems for the past few years. The basic MF model splits the rating matrix into the user and item potential feature matrix and improves the rating prediction accuracy by learning the potential relationship between the user and item.

With the advent of the information interconnection era, the basic MF model no longer met the requirements of the recommendation systems, which prompted the development of many variants of this model. To improve the accuracy of the rating prediction, Levy et al. [11] proposed a semi-supervised non-negative matrix method that can fuse prior information kernels based on the MF model. Dong-Kyu et al. [12] and Salakhutdinov et al. [13] added a graph probability model to better adapt to the real data of the sparse environment. Salakhutdinov et al. [14] interpreted the MF model as a probabilistic graph model to alleviate the sparsity and imbalance of the real dataset. Koren et al. [15] introduced the user and item biases to model specific characteristics of the user and item. Kim et al. [16] proposed a method based on the content metadata to effectively utilize content metadata to alleviate the problem of data sparsity. Koren et al. [17] regarded all of the unobserved user ratings as a negative feedback process. Then, Devooght et al. [18] considered the user and item bias and the impact of the rated items other than the user- and item-specific vectors on the rating prediction in order to objectively describe the user preferences.

In addition to the above MF models that aimed to further improve recommendation system performance based on rating data, researchers integrated the trust relationship between users into the MF model. Jamali et al. [19] proposed a new social recommendation model named SocialMF based on the trust propagation mechanism in social networks. Ma et al. [20] introduced the SoRec model that adds the user's social relationship information to the original Probabilistic Matrix Factorization

(PMF) recommendation model. Different from the social information construction method of the SoRec model, the recommended model TrustMF was proposed by Yang et al. [21] based on the relationship between the trustees and trusters. The TrustMF model maps each user to two different K-dimensional feature vectors according to the directionality of trust relationship. Different from the above models which decompose social information into two feature vector inner product forms, Tang et al. [22] proposed a social recommendation model named LOCABAL (local and global) that combines the local and global social information. Guo et al. regarded the user's explicit trust relationship and the item's ratings as the implicit feedback information and added the social feedback information to the original SVD++ model to reconstruct the TRUST-SVD rating prediction model. Hao et al. [23] proposed a new collaborative filtering method based on social information fusion. This approach firstly relies on the social information fusion to search for similar users, and then uses the similar users to update the product user ratings to generate recommendations.

The user embedding model proposed in this paper was inspired by the word-embedding models [24] in the field of natural language processing. Among them, Word2vec is often used to solve the similarity problem between words. In recommendation systems, we can think of the collection of all users as one document and each user as an individual word. Therefore, Levy et al. [11] used the implicit similarity of Word2vec technology for the MF model. Mikolov et al. [25] used the word embedding model to model the item features in recommendation systems. Guardiasebaoun et al. [26] applied the item embedding technique to the CF algorithm. Barkan et al. [27] not only introduced item embedding technology into the MF model, but also quantified the items that users like or dislike, improving the rating prediction accuracy.

3. Proposed UE-SVD++ Model

In this section, we firstly introduce the fundamental theory of the proposed UE-SVD++ model. Then, we describe the user embedding method. Ultimately, the details and the training process of the UE-SVD++ model are illustrated.

3.1. Fundamental Theory

The abovementioned SVD-based collaborative filtering method is widely used in recommendation systems. In particular, the SVD++-based CF algorithms add the bias and implicit feedback of the user into the MF model to improve the rating prediction accuracy. Hence, we adopt the popular SVD++ model into our proposed rating prediction method. This method firstly analyzes the potential correlation between the different users based on the proposed UMI values. Thus, the user embedding matrix is extracted from the rating matrix. Finally, the user embedding matrix is added to the existing user bias and implicit parameters to model the user features more accurately for rating prediction.

Generally, the proposed UE-SVD++ model for rating prediction has several components, including the process of generating the favored users of every item, the user embedding matrix, the SVD++ model, and predicting the rating. Figure 1 illustrates the framework of the proposed UE-SVD++ model. In Figure 1, the data used are the four datasets commonly used in the recommendation systems that contain a rating of "0–5" used to represent the user satisfaction. A value of "5" means the highest positive rating, while a value of "0" means the lowest negative rating. Therefore, inspired by previous works, we define a user with a rating of greater than "3.5" on the given item as the favored user of this item and calculate the favored user set for each item. Then, the user embedding matrix is constructed based on the proposed UMI values. In addition, based on the preprocessed user embedding matrix and the original rating matrix, the SVD++ model is introduced to complete matrix factorization, and both the user matrix and the item matrix are obtained sequentially. The user matrix represents the latent information of the user. The item matrix represents the latent information of the item. Using these two matrices, it is easy to predict the rating of the target user for all of the items, and the dot product between these matrices is computed to complete the rating prediction.
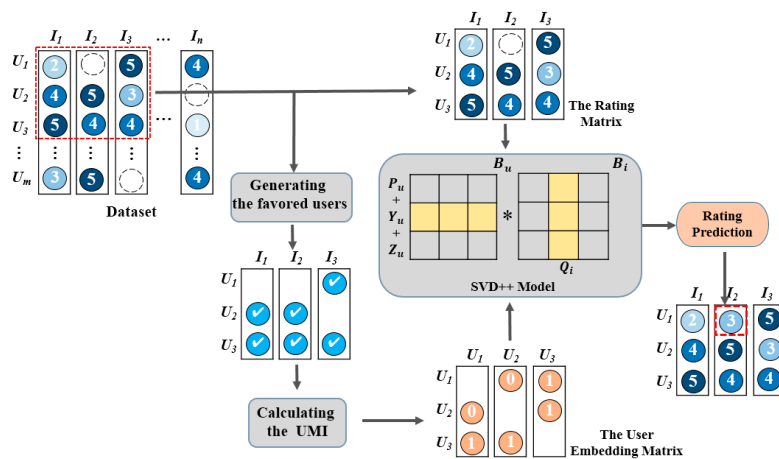
Figure 1. The framework of the proposed user embedding model for rating prediction in singular value decomposition plus plus (SVD++)-based collaborative filtering (UE-SVD++).

3.2. User Embedding Matrix

Generally, we consider that users who frequently consume the same item are similar, just as the words that appear in a document are related to the same topic. For example, "organic food" and "fitness" may appear in a series of health articles at the same time. Due to the excellent performance of the Word2vec technology in the field of natural language processing, Barkan et al. added the Word2vec technology into the CF algorithm for recommendation systems. Inspired by Adomavicius et al., when users A and B give the same item a higher rating (>3.5/5), we can set users A and B as having similar interests. Therefore, if user A gives a higher rating on an item that is not in user B's evaluation item list, we can recommend the item to user B.

Based on the above description, modeling with the original rating data, all users of item 1 to N were statistically evaluated, and the users who rated an item greater than 70% of the highest rating were identified as the favored user data for a particular item. For instance, if we consider the rating data of three users and three items, the calculation process of the favored users of the three items is as shown in Figure 2.
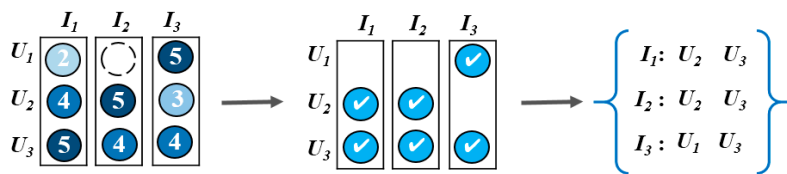

Figure 2. The rating matrix is converted into a list of favored users for each item.

In Figure 2, $\{U_2, U_3\}$, $\{U_2, U_3\}$, and $\{U_1, U_3,\}$ are the favored user collections of I_1, I_2, and I_3. Here, we use the user-wise mutual information $UMI(i, j)$ value to measure the potential relationship between U_i and U_j. Let a specific user be a word in the context; the total number of words and word context pairs is D. Then, the $UMI(i, j)$ between the word U_i and its context U_j is calculated as follows:

$$UMI(i, j) = log \frac{P(U_i, U_j)}{P(U_i) * P(U_j)}, \tag{1}$$

$$P(U_i, U_j) = \frac{\#(U_i, U_j)}{|D|}, \tag{2}$$

$$P(U_i) = \frac{\#(U_i)}{|D|}, \tag{3}$$

$$P(U_j) = \frac{\#(U_j)}{|D|}, \tag{4}$$

where $P(U_i, U_j)$ is the probability that word U_i and word U_j appear simultaneously in a fixed window, $P(U_i)$ is the probability that U_i appears in set D, and $P(U_j)$ is the probability that U_j appears in set D. Therefore, substituting Equations (2)–(4) into Equation (1) yields the following equation:

$$UMI(U_i, U_j) = log\frac{\#(U_i, U_j) * |D|}{\#(U_i) * \#(U_j)}. \tag{5}$$

By calculating the *UMI* value of the specific U_i and U_j in the user set, we can form a *UMI* matrix with dimension $m \times m$. Here, m is the total number of users. To increase the credibility of the *UMI* matrix, we introduce the threshold parameter k to primarily control the credibility of the matrix. When k is larger, the calculated *UMI* value is larger, and the similarity between U_i and U_j is greater. Then, we only reserve $KUMI(U_i, U_j)$ greater than the threshold k, and the calculation method is as shown in Equation (6).

$$KUMI(U_i, U_j) = max\{UMI(U_i, U_j),\ log(k)\}, \tag{6}$$

where the $max\{UMI(U_i, U_j),\ log(k)\}$ means that only the $UMI(U_i, U_j)$, which is greater than the threshold k, is reserved; finally, the user embedding matrix is constructed. Algorithm 1 shows the details of the construction process of the user embedding matrix.

Algorithm 1: The process of computing the user embedding matrix.

Input: the rating matrix
Output: the user embedding matrix
1. Count all user collections that rated on the item based on the rating matrix.
2. Calculate the number of the favored users for a particular item. The users who rated for an item with a rating greater than 70% of the highest rating are the favored user data.
3. Compute the user-wise mutual information (*UMI*) value of specific U_i and U_j using Equations (1)–(4).
4. Filter the UMI values using Equation (6).
5. Generate the user embedding matrix.

3.3. Proposed UE-SVD++ Model

Compared with the traditional MF algorithm, the singular value decomposition plus plus (SVD++) model adds the user bias information and implicit parameters to objectively describe the user preferences. User ratings imply preference information; some ratings are low (or high). Therefore, we design our model based on the SVD++ to better mine user preferences and accurately calculate predicted rating. The rating prediction calculation in the SVD++ model is given by Equation (7).

$$\widehat{R}_{u,i} = B_u + B_i + \mu + Q_i^T\left(P_u + |N_u|^{-\frac{1}{2}}\sum_{n\in N_u}Y_n\right), \tag{7}$$

where $\widehat{R}_{u,i}$ represents the predicted rating given by user u to item i, and μ represents the mean of the global ratings. B_u represents the user bias (the rate habits of a particular user), which is independent of the items' characteristics. B_i represents the item bias (a rating given for a particular item), which is independent of the user interest. Then, μ represents the mean of the global ratings. Q_i and P_u are the latent feature vectors of item i and user u. For a particular user u, the collection of items that provides the implicit feedback is defined as N_u. Y_n denotes the implicit influence of items rated by user u in the past on the ratings of unknown items in the future, and all of the user u's ratings revised values are $\sum_{n\in N_u}Y_n$. These are usually expressed as $Q_i^T Y_n$, and $|N_u|^{-1/2}$ is introduced to eliminate the difference

caused by different N_u numbers. To sum up, the SVD++ model describes the user preferences based on bias information, implicit parameters, etc. The object function in the SVD++ model is given by Equation (8).

$$J_0 = \frac{1}{2} \sum_{(u,i)\in R} \left\{ R_{u,i} - \mu - B_u - B_i - Q_i^T \left(P_u + |N_u|^{-1/2} \sum_{n\in N_u} Y_n \right) \right\}^2 \\ + \frac{\lambda}{2} \left\{ \sum_u \left(B_u^2 + \|P_u\|^2 \right) + \sum_i \left(B_i^2 + \|Q_i\|^2 + \|Y_n\|^2 \right) \right\}. \tag{8}$$

The loss value is calculated in the first part of Equation (8), where R is the rating matrix, the latter part $\lambda \left\{ \sum_u \left(B_u^2 + \|P_u\|^2 \right) + \sum_i \left(B_i^2 + \|Q_i\|^2 + \|Y_n\|^2 \right) \right\}$ is a regularization term for preventing the over-fitting of the model, and λ is the regularization coefficient.

The UE-SVD++ model adds the user embedding matrix to the user structure based on the existing user bias information and implicit parameters of the SVD++ model to improve the accuracy of the rating prediction. The rating prediction calculation method is given by Equation (9).

$$\widehat{R}_{u,i} = B_u + B_i + \mu + Q_i^T \left(P_u + |N_u|^{-1/2} \sum_{n\in N_u} Y_n + |D_u|^{-1/2} \sum_{\omega\in D_u} Z_\omega \right), \tag{9}$$

where D_u is the dependent dataset of user u, and Z_ω is the specific user potential feature vector on which user u depends. Therefore, $Q_i^T |D_u|^{-1/2} \sum_{\omega\in D_u} Z_\omega$ represents the influence of the dependent user on the rating of a particular user based on a certain item. In simple terms, we model the user feature vectors based on the item rating matrices and the user embedding matrices. The optimization objective function of the *UE-SVD++* model is given by Equation (10).

$$J_1 = \frac{1}{2} \sum_u \sum_{i\in I_u} \left(R_{u,i} - \widehat{R}_{u,i} \right)^2 + \frac{\theta_d}{2} \sum_u \sum_{\omega\in D_u} \left(D_{u,\omega} - \widehat{D}_{u,\omega} \right)^2, \tag{10}$$

where $\widehat{D}_{u,\omega}$ is the prediction dependence between users u and ω, and θ_d is a parameter that controls the degree of dependence between different users.

Guo et al. proposed the use of smaller penalties for the popular users and items to decrease the likelihood of over-fitting. In contrast, those items which rarely get a rating or users with very few ratings have a large likelihood of over-fitting, and they should be weighted. Then, the new objective function is given by Equation (11).

$$J_2 = \frac{1}{2} \sum_u \sum_{i\in I_u} \left(R_{u,i} - \widehat{R}_{u,i} \right)^2 + \frac{\theta_d}{2} \sum_u \sum_{\omega\in D_u} \left(D_{u,\omega} - \widehat{D}_{u,\omega} \right)^2 \\ + \frac{\theta}{2} \left\{ \sum_u |N_u|^{-1/2} \left(B_u^2 + \|P_u\|^2 + \|Y_n\|^2 \right) + \sum_i |N_i|^{-1/2} \left(B_i^2 + \|Q_i\|^2 \right) \right\} \\ + \frac{\theta_d}{2} \left\{ \sum_u |D_u|^{-1/2} \|P_u\|^2 + \sum_\omega |D_\omega|^{-1/2} \|Z_\omega\|^2 \right\}. \tag{11}$$

The first part of J_2 is the loss based on the least square method. The second part of J_2 is the regularization term. In addition, the initial values of B_u and B_i are set as the null vectors, and the initial values of P_u and Q_i are obtained from the normal distribution by sampling with the zero mean value. The UE-SVD++ model is optimized by the stochastic gradient descent (SGD) [28] method.

$$e_{u,i} = \left| R_{u,i} - \widehat{R}_{u,i} \right|, \tag{12}$$

$$B_u^{(t)} \leftarrow B_u^{(t-1)} + \gamma \left(e_{u,i} - \theta |N_u|^{-1/2} B_u^{(t-1)} \right), \tag{13}$$

$$B_i^{(t)} \leftarrow B_i^{(t-1)} + \gamma \left(e_{u,i} - \theta |N_i|^{-1/2} B_i^{(t-1)} \right), \tag{14}$$

$$P_u{}^{(t)} \leftarrow P_u{}^{(t-1)} + \gamma\left\{e_{u,i} - \left(\theta|N_u|^{-1/2} + \theta_d|D_u|^{-1/2}\right)P_u{}^{(t-1)}\right\}, \tag{15}$$

$$Q_i{}^{(t)} \leftarrow Q_i{}^{(t-1)} + \gamma\left\{e_{u,i}\left(P_u{}^{(t-1)} + |N_u|^{-1/2}\sum_{n\in N_u}Y_n{}^{(t-1)} + |D_u|^{-1/2}\sum_{\omega\in D_u}Z_\omega{}^{(t-1)}\right) - \theta Q_i{}^{(t-1)}\right\}, \tag{16}$$

$$Y_n{}^{(t)} \leftarrow Y_n{}^{(t-1)} + \gamma\left(e_{u,i}|N_u|^{-1/2}Q_i{}^{(t-1)} - \theta Q_i{}^{(t-1)}\right), \tag{17}$$

$$Z_\omega{}^{(t)} \leftarrow Z_\omega{}^{(t-1)} + \gamma\left\{e_{u,i}\left(|D_u|^{-1/2}Q_i{}^{(t-1)} + \theta P_u{}^{(t-1)}\right) - \theta_d Z_\omega{}^{(t-1)}\right\}, \tag{18}$$

where $e_{u,i}$ is the prediction error, γ is the learning rate, and θ is the regularization parameter. Additionally, (t) represents the current iteration, and (t − 1) represents the previous iteration update. Algorithm 2 shows the details of the proposed UE-SVD++ model based on the SVD++ model.

Algorithm 2: The Proposed UE-SVD++ algorithm.

Input: the rating matrix, the user embedding matrix
Output: the predicted rating matrix
1.　Calculate the mean rating μ based on the rating matrix.
2.　Initialize the "bias information" B_u and B_i. Initialize the user vector P_u and the item vector Q_i. Initialize the user embedding hidden feature Z_w. Initialize the implicit parameters Y_n.
3.　Calculate the inner product of the user vector P_u and the item vector Q_i using Equation (9). User ratings are predicted.
4.　Calculate the prediction error $e_{u,i}$ based on the real rating and the predicted rating. The stochastic gradient descent (SGD) method is utilized to complete optimization, as shown in Equations (12)–(18).
5.　Repeat the third step and fourth step to get the prediction rating $\widehat{R}_{u,i}$. Update the predicted rating matrix.

---

4. Experiment

In this section, we firstly introduce the four datasets used in our experiment and the evaluation indexes of the model. Secondly, we carry out several experiments to select the optimal parameters of our proposed model. Finally, we compare the performance of the UE-SVD++ with other rating prediction models.

4.1. Experimental Datasets and Evaluations

The experiment used four widely used datasets in the recommended field, FilmTrust, Epinions, MovieLens-100K, and EachMovie, to evaluate the effectiveness of the UE-SVD++ model. The statistics of the four datasets are presented in Table 1.

Table 1. Statistics of the experimental datasets.

	Users	**Items**	**Ratings**	**Ratings Range**	**Density**
FilmTrust	1508	2071	35,497	{0.5, … , 4.0}	1.136%
Epinions	49,289	139,738	664,823	{1.0, … , 5.0}	0.051%
MovieLens-100K	943	1682	100,000	{1.0, … , 5.0}	6.304%
EachMovie	29,520	1648	1,048,575	{0.2, … , 1.0}	2.155%

To measure the accuracy of the rating prediction, we used two commonly used evaluation indicators, namely, the root-mean-square error (*RMSE*) and the mean absolute error (*MAE*), which were obtained by calculating the difference between the true rating and the predicted rating to measure the accuracy of the rating prediction of recommendation systems. A smaller value corresponds to a higher accuracy of the rating prediction. RMSE and MAE are defined by the following equations:

$$RMSE = \sqrt{\frac{1}{T}\sum_{u,i}\left(R_{u,i} - \widehat{R}_{u,i}\right)^2}, \tag{19}$$

$$MAE = \frac{1}{T} \sum_{u,i} \left| R_{u,i} - \widehat{R}_{u,i} \right|,$$

(20)

where T is the number of the rating records of the test set, $R_{u,i}$ is the true rating value, and $\widehat{R}_{u,i}$ is the predicted rating value.

4.2. Model Parameter Selection

The selection of the UE-SVD++ model parameters has great impact on rating prediction performance. To obtain more accurate data, we used the average values obtained from five experiments as our data. As shown in Figures 3–6, we listed the effects of the changes in the parameters on each dataset during training on the performance of the UE-SVD++ model.

It is worth mentioning that the method to select optimal parameters was the control variable method in this section. For example, when we wanted to find the optimal value of the parameter k, we fixed the other three parameters unchanged, and so on. Of course, the selection of the initial value of the parameter was not random. We determined the value range of the parameter based on previous work, so as to try to find the optimal value of this parameter.
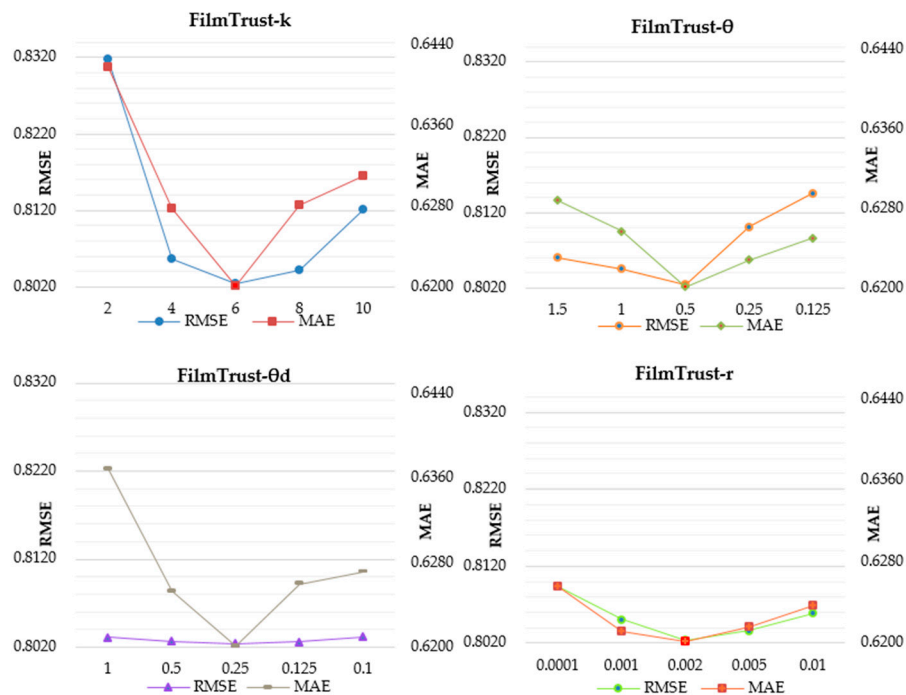

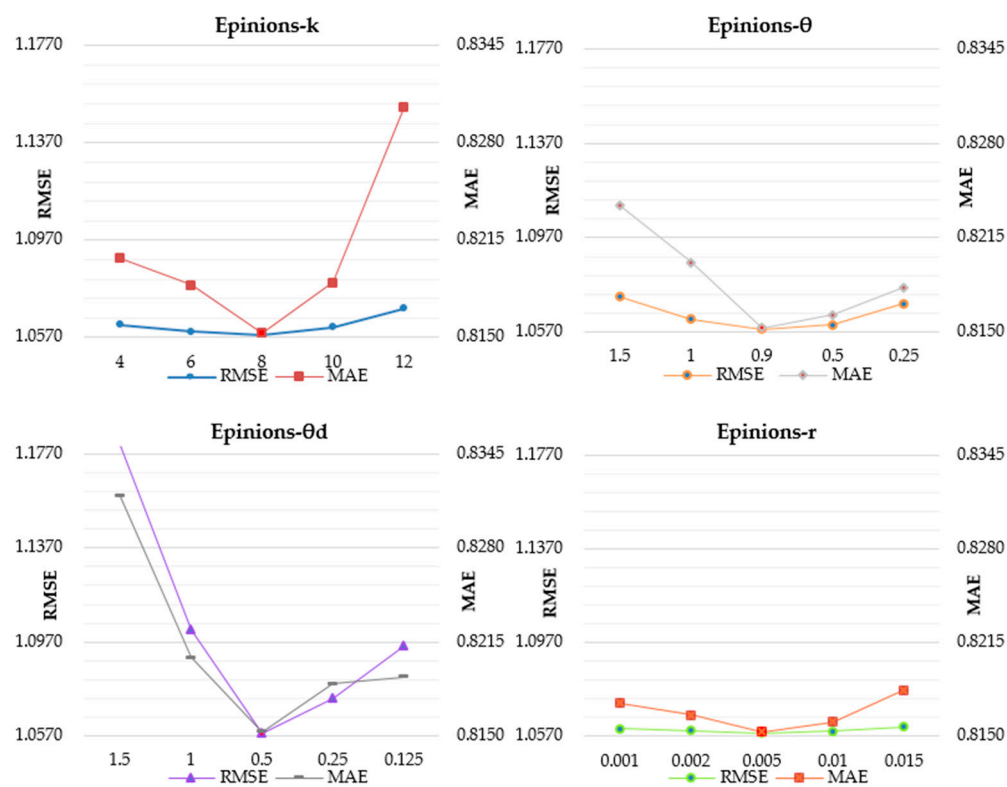**Figure 3.** Parameter optimization for UE-SVD++ in FilmTrust dataset.

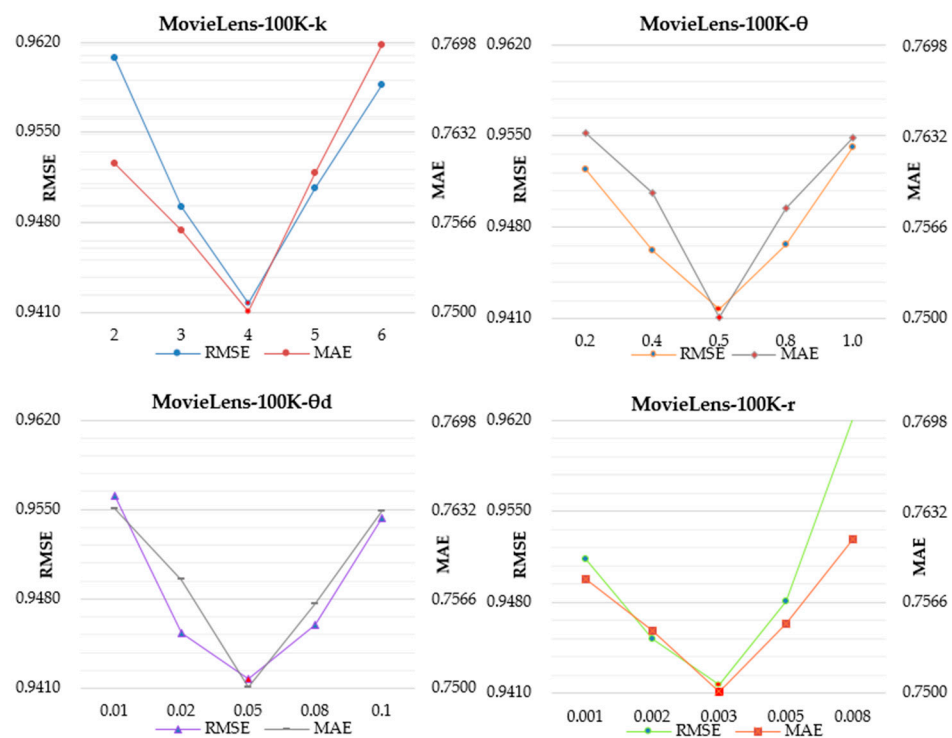**Figure 4.** Parameter optimization for UE-SVD++ in Epinions dataset.

Figure 5. Parameter optimization for UE-SVD++ in MovieLens-100K dataset.
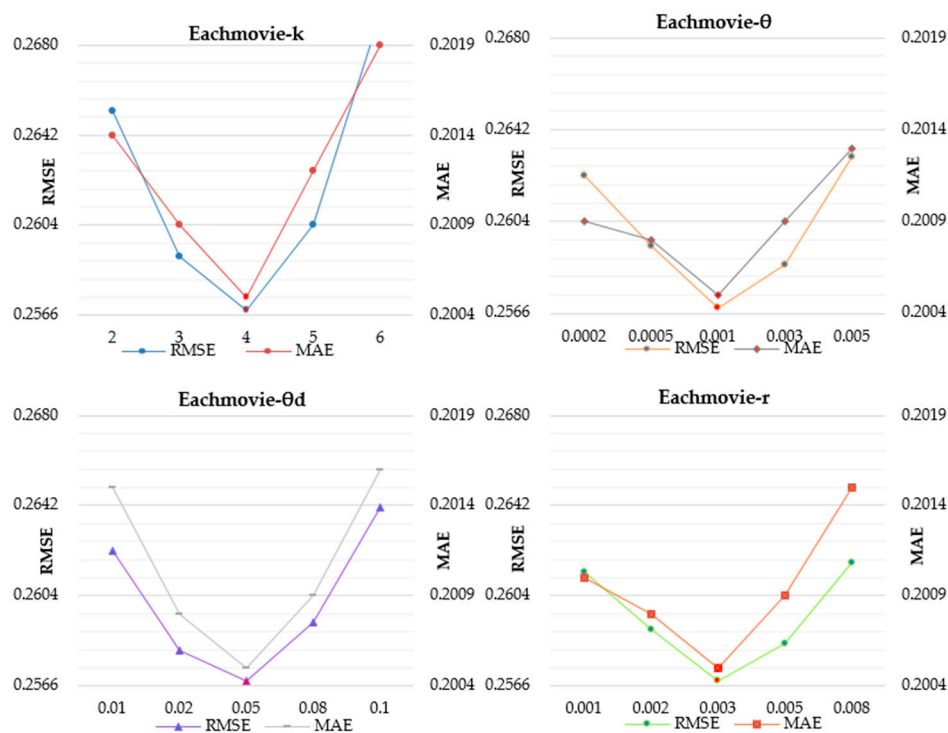
Figure 6. Parameter optimization for UE-SVD++ in Eachmovie dataset.

As can be seen in Figures 3–6, the change in rating prediction accuracy of UE-SVD++ was mostly caused by the change in parameter k. The reason is that parameter k controls the density of the user embedding matrix in the model. When the value of k is small, the density of the embedding matrix is very high, and users with low correlation are also added, which leads to inaccurate modeling of users and low prediction accuracy. When the value of parameter k is large, only a few users are added into the embedding matrix, which also leads to the weak generalization ability of the model and the decline in prediction accuracy. θ is the parameter used to prevent the model from over-fitting, θd is the parameter used to control the weight of the relationship between users in the model, and γ is the model's learning rate. From the above description, it can be seen that, when the value of parameter k is determined, the changes in the other three parameters only cause slight changes in the model.

Based on the experimental results, we can draw a few conclusions. The optimal parameters of the FilmTrust were as follows: UMI threshold parameter k = 6, regularization parameter θ = 0.5, θd = 0.25, learning rate γ = 0.002. The optimal parameters of the Epinions were as follows: UMI threshold parameter k = 8, regularization parameter θ = 0.9, θd = 0.5, learning rate γ = 0.005. The optimal parameters of MovieLens-100K were as follows: UMI threshold parameter k = 4, regularization parameter θ = 0.5, θd = 0.05, learning rate γ = 0.003 The optimal parameters of the EachMovie were as follows: UMI threshold parameter k = 9, regularization parameter θ = 0.001, θd = 0.05, learning rate γ = 0.003. The training order of parameters was k, θ, θd, γ. For instance, when the parameter θ is trained, the remaining parameters were fixed.

4.3. Compared Models

To further prove the good performance of the UE-SVD++ model in rating prediction, we used 10 classical models based on the CF algorithm for comparison. The comparison models used in this paper are described below.

USER-MF (user based matrix factorization) and ITEM-MF (item based matrix factorization) perform rating prediction based on the user's given item rating and the average of the ratings received by the items. SOCIAL-MF uses the direct social relationship (user trust) to model the user feature matrix to achieve a better effect of the rating prediction. PMF introduces a probabilistic

model based on the regularization matrix factorization to further optimize the effect of the rating prediction. A social recommendation model (LOCABAL) [22] combines the rating matrix and the social relation matrix by using local and global social relations. FUNK-SVD decomposes the sparse and high-dimensional rating matrix into two low-dimensional matrices and predicts the user's rating for the item using the reconstructed low-dimensional matrices. BIAS-SVD considers the user-independent or item-independent factors as the biased part, and considers the interaction between the user and the item as a personalized part, and it is considered that the bias part in the MF model plays a more essential role in improving the accuracy of rating prediction than the personalized part. A recommendation model (MFC) [29] of regularization and integration of social information in overlapping communities considers that users classified into the same communities have similar tastes. The SVD++ model adds the implicit feedback to the users based on BIAS-SVD. The TRUST-SVD model follows the approach of SVD++, while adding the implicit social feedback information to represent the user feature matrix, and the social relationship matrix is also decomposed to form the user feature matrix and the social feature matrix.

Since the MovieLens-100K and EachMovie datasets do not contain user trust data, the SOCIAL-MF, LOCABAL, MFC, and TRUST-SVD models were not applied to the above two datasets. To make the experimental results more accurate and representative, we used the classic evaluation indicators (RMSE and MAE) in the rating prediction. Each group of data was tested five times to obtain the mean value. An examination of the results presented in Table 2 shows that the RMSE and MAE values of the UE-SVD++ model on the four different datasets were superior to the other compared models, and it is clear that the evaluation performance was stable.

Table 2. Comparison of rating prediction performance of different models. RMSE—root-mean-square error; MAE—mean absolute error.

Datasets	FilmTrust		Epinions		MovieLens-100K		EachMovie	
Metrics Models	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
ITEM-MF	0.9274	0.7138	1.2001	0.9134	1.0872	0.8717	0.2842	0.2228
USER-MF	0.8960	0.6646	1.1894	0.9128	1.0946	0.8648	0.2847	0.2296
PMF	0.8541	0.6554	1.1152	0.8426	0.9573	0.7602	0.2627	0.2058
BIAS-SVD	0.8199	0.6311 *	1.0809 *	0.8342	0.9629	0.7587 *	0.2597	0.2029 *
FUNK-SVD	0.8487	0.6546	1.0970	0.8331	0.9587	0.7594	0.2615	0.2054
SVD++	0.8340	0.6315	1.1194	0.8315	0.9521 *	0.7624	0.2594 *	0.2035
LOCABAL	0.8297	0.6519	1.1316	0.8477				
SOCIAL-MF	0.8506	0.659	1.0823	0.8314				
MFC	0.8198	0.6496	1.1263	0.836				
TRUST-SVD	0.8197 *	0.6349	1.0908	0.8258 *				
UE-SVD++	**0.8024**	**0.6201**	**1.0583**	**0.8153**	**0.9417**	**0.7501**	**0.2568**	**0.2005**
Improve	**2.110%**	**1.742%**	**2.091%**	**1.271%**	**1.092%**	**1.133%**	**1.002%**	**1.182%**

*: It's to indicate the second best performance.

4.4. Performance Comparison

In this section, we compare our proposed UE-SVD++ with the above comparison models. The model was trained and tested by the method of five-fold cross validation. Specifically, the datasets were divided equally at random, with 80% of the dataset used as the training set and 20% as the test set in each test. The use of five experiments ensured that all of the data were tested, and the final test result was the average of the results of the five experiments.

In Table 2, we show the performance of all the comparison models and our proposed model on the four datasets. We use a bold font to represent the best performance on each of the four datasets, and the symbol * to indicate the second best performance. Among the models tested, TRUST-SVD and BIAS-SVD performed better on the FilmTrust and Epinions datasets. SVD++ and BIAS-SVD performed better on Movielens-100k and EachMovie datasets. The main reason is that the above models added the implicit feedback information to re-express the user feature matrix. It is clear that our model

(UE-SVD++) was superior to the other models in terms of each metric for both datasets. Relative to the second best performance, we showed an improvement of our model. The RMSE index of UE-SVD++ improved by 1.002%–2.110%, and the MAE index improved by 1.133–1.742%. The above experimental results show that our model has a high accuracy when effectively rating prediction.

4.5. The Influence of the Training Data Volume on Model Performance and the Execution Time

In this section, we used the FilmTrust dataset to study the influence of the amount of training data on the model performance and the execution time, and the results are shown in Table 3. We reduced the amount of training data during model training randomly. For example, 50% of the training data meant that we randomly selected 50% data from the total rating data for the training of the model. The experimental results show that, when the training data were more than or equal to 60%, the prediction accuracy did not fluctuate too much. When the data were lower than 60%, the fluctuation of accuracy increased obviously. Then, with the increase in training data, the time used to train the model also increased gradually. In conclusion, it is apt to take 80% of the training data as the benchmark of the model. Although the model takes a long time, it is acceptable to have the best prediction accuracy.

Table 3. The impact of less training data on performance in the FilmTrust dataset.

	30% Training	40% Training	50% Training	60% Training	70% Training	80% Training
RMSE	0.8634	0.8406	0.8291	0.8196	0.8109	0.8024
MAE	0.6742	0.6684	0.6521	0.6324	0.6259	0.6201
Time (s)	1394.2	1420.6	1482.4	1549.7	1636.4	1719.1

5. Conclusions

The rating prediction is one of the key tasks in recommendation systems, and the user preference mining is a key factor that restricts the accuracy of rating prediction. In this paper, we propose a novel rating prediction model based on user embedding and SVD++ named UE-SVD++.

As a classical rating prediction model in the CF algorithm, the SVD++ model ignores the explicit feedback of the user in the rating. To enrich the user features and improve the accuracy of rating prediction, the user embedding matrix based on the rating is proposed in this paper to enhance the correlation between the users. The user embedding matrix represents the potential association between two users. Based on the rating data, we calculate the number of times that any two users consumed the same item at the same time to obtain the user-wise mutual information values, and then construct the user embedding matrix. Eventually, the user embedding matrix is added to the existing user bias and implicit parameters to model the user features on rating prediction. The experimental results on four real-world datasets demonstrate that the UE-SVD++ model outperforms the state-of-the-art rating prediction models.

Despite the good rating prediction effect, there are still some areas we ignored in this method. In future work, we will consider the introduction of new metrics to measure the proposed approach more comprehensively. We will also introduce user assistance information as much as possible, such as the social relationship data between users.

Author Contributions: Conceptualization, J.Q.; data curation, W.S. and L.W.; formal analysis, J.Q.; funding acquisition, L.W. and J.Q.; methodology, J.Q.; resources, W.S. and L.W.; supervision, J.Q.; validation, W.S. and L.W.; writing—original draft, W.S.; writing—review and editing, W.S., L.W., and J.Q. All authors have read and agreed to the published version of the manuscript.

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

References

1. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative filtering for implicit feedback datasets. In Proceedings of the Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272.
2. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *2009*, 30–38. [CrossRef]
3. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. 1994. Available online: https://dl.acm.org/doi/10.1145/192844.192905 (accessed on 11 December 2019).
4. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**, *2009*, 1–19. [CrossRef]
5. Deshpande, M.; Karypis, G. Item-Based top-n, recommendation algorithms. *ACM Trans. Inf. Syst.* **2004**, *22*, 143–177. [CrossRef]
6. Linden, G.; Smith, B.; York, J. Amazon.com recommendations: Item-to-Item collaborative filtering. *IEEE Internet Comput.* **2003**, *7*, 76–80. [CrossRef]
7. Koren, Y. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data* **2010**, *4*, 1. [CrossRef]
8. 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 29th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–29 January 2015; AAAI Press: Menlo Park, CA, USA, 2015; pp. 123–129.
9. Adomavicius, G.; Kwon, Y.O. Overcoming accuracy-diversity tradeoff in recommender systems: A variance-based approach. In Proceedings of the WITS, Paris, France, 1 January 2008; p. 8.
10. Adomavicius, G.; Kwon, Y. Improving recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* **2009**, 896–911. [CrossRef]
11. Levy, O.; Goldberg, Y. Neural Word Embedding as Implicit Matrix Factorization. 2014. Available online: http://papers.nips.cc/paper/5477-neural-word-embedding-as (accessed on 11 December 2019).
12. Chae, D.K.; Shin, J.A.; Kim, S.W. Collaborative Adversarial Autoencoders: An Effective Collaborative Filtering Model under the GAN Framework. *IEEE Access* **2019**, *7*, 37650–37663. [CrossRef]
13. Mnih, A.; Salakhutdinov, R. Probabilistic Matrix Factorization. 2008. Available online: http://papers.nips.cc/paper/3208-probabilistic-matrix-factorization.pdf (accessed on 11 December 2019).
14. Salakhutdinov, R.; Mnih, A. Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo. 2008. Available online: https://dl.acm.org/doi/10.1145/1390156.1390267 (accessed on 11 December 2019).
15. Koren, Y.; Bell, R. Advances in collaborative filtering. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015; pp. 77–118.
16. Kim, K.S.; Chang, D.S.; Choi, Y.S. Boosting Memory-Based Collaborative Filtering Using Content-Metadata. *Symmetry* **2019**, *11*, 561. [CrossRef]
17. Koren, Y. Collaborative Filtering with Temporal Dynamics. 2009. Available online: https://dl.acm.org/doi/10.1145/1557019.1557072 (accessed on 11 December 2019).
18. Devooght, R.; Kourtellis, N.; Mantrach, A. Dynamic Matrix Factorization with Priors on Unknown Values. 2015. Available online: https://dl.acm.org/doi/10.1145/2783258.2783346 (accessed on 11 December 2019).
19. Jamali, M.; Ester, M. A Matrix Factorization Technique with Trust Propagation for Recommendation in Social Networks. 2010. Available online: https://dl.acm.org/doi/10.1145/1864708.1864736 (accessed on 11 December 2019).
20. Ma, H.; Yang, H.; Lyu, M.R.; King, I. SoRec: Social Recommendation Using Probabilistic Matrix Factorization. 2008. Available online: https://dl.acm.org/doi/10.1145/1458082.1458205 (accessed on 11 December 2019).
21. Yang, B.; Lei, Y.; Liu, D.Y.; Liu, J.M. Social Collaborative Filtering by Trust. 2013. Available online: https://ieeexplore.ieee.org/abstract/document/7558226/ (accessed on 11 December 2019).

22. Tang, J.; Hu, X.; Gao, H.; Liu, H. Exploiting Local and Global Social Context for Recommendation. 2013. Available online: https://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/viewPaper/6936 (accessed on 11 December 2019).

23. Wang, H.; Song, Y.; Mi, P.; Duan, J. The Collaborative Filtering Method Based on Social Information Fusion. *Math. Probl. Eng.* **2019**. Available online: https://www.hindawi.com/journals/mpe/2019/9387989/abs/ (accessed on 11 December 2019).

24. Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge discovery and data mining, Las Vegas, NV, USA, 24–27 August 2008; ACM Press: New York, NY, USA, 2008; pp. 426–434.

25. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. 2013. Available online: http://papers.nips.cc/paper/5021-distributed-representations-of-words-andphrases (accessed on 11 December 2019).

26. Guàrdia-Sebaoun, E.; Guigue, V.; Gallinari, P. Latent Trajectory Modeling: A Light and Efficient Way to Introduce Time in Recommender Systems. 2015. Available online: https://dl.acm.org/doi/10.1145/2792838.2799676 (accessed on 11 December 2019).

27. Barkan, O.; Koenigstein, N. Item2vec: Neural Item Embedding for Collaborative Filtering. 2016. Available online: https://ieeexplore.ieee.org/abstract/document/7738886/ (accessed on 11 December 2019).

28. Yang, X.; Steck, H.; Liu, Y. Circle-Based Recommendation in Online Social Networks. 2012. Available online: https://dl.acm.org/doi/10.1145/2339530.2339728 (accessed on 11 December 2019).

29. Li, H.; Wu, D.; Tang, W.; Mamoulis, N. Overlapping Community Regularization for Rating Prediction in Social Recommender Systems. 2015. Available online: https://dl.acm.org/doi/10.1145/2792838.2800171 (accessed on 11 December 2019).