TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 19, No. 2, April 2021, pp. 471~478
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v19i2.18149

Adopting explicit and implicit social relations by SVD++ for recommendation system improvement

Mohsin Hasan Hussein, Akeel Abdulkareem Alsakaa, Haydar A. Marhoon
Department of Computer Science, College of Science, University of Kerbala, Iraq

Article Info

Article history:

Received Jul 18, 2020
Revised Oct 1, 2020
Accepted Oct 16, 2020

Keywords:

Recommendation system
Social Relation
Sparsity
SVD++

ABSTRACT

Recommender systems suffer a set of drawbacks such as sparsity. Social relations provide a useful source to overcome the sparsity problem. Previous studies have utilized social relations or rating feedback sources. However, they ignored integrating these sources. In this paper, the limitations of previous studies are overcome by exploiting four sources of information, namely: explicit social relationships, implicit social relationships, users' ratings, and implicit feedback information. Firstly, implicit social relationships are extracted through the source allocation index algorithm to establish new relations among users. Secondly, the similarity method is applied to find the similarity between each pair of users who have explicit or implicit social relations. Then, users' ratings and implicit rating feedback sources are extracted via a user-item matrix. Furthermore, all sources are integrated into the singular value decomposition plus (SVD++) method. Finally, missing predictions are computed. The proposed method is implemented on three real-world datasets: Last.Fm, FilmTrust, and Ciao. Experimental results reveal that the proposed model is superior to other studies such as SVD, SVD++, EU-SVD++, SocReg, and EISR in terms of accuracy, where the improvement of the proposed method is about 0.03% for MAE and 0.01% for RMSE when dimension value (d) = 10.

Corresponding Author:

Mohsin Hasan Hussein
Department of Computer Science
University of Kerbala
Kerbala 56001, Iraq
Email: mohsin.h@uokerbala.edu.iq

1. INTRODUCTION

Due to the exponential growth of Internet e-commerce, the information provided by the Internet has increased and this phenomenon is called information overflow. Accordingly, finding preferred products on websites has become a ponderous task [1, 2]. Therefore, a recommendation system is utilized to solve the aforementioned problem by reducing search times and predicting a user's preferences of products. A recommendation system (RS) can be defined as a tool that can recommend a list of items to interested users [3]. A recommendation system comes in a variety of types, collaborative filtering being the most common and most widely used [4, 5]. Collaborative filtering estimates unknown ratings based on user history ratings and behaviors of similar users. Such systems are called user-oriented. An item-oriented approach was proposed by [6] where predictions are generated on the basis of similarities between items. However, most recent datasets are rather huge and have comprised thousands of users and items, including Netflix, Facebook, and YouTube, so computing similarity becomes costly. There are two ways to implement this task: first, by finding the similarity between users or items, and second by finding the latent factors model of users and items which can be implemented by converting both users and items to the same latent factor followed by comparisons between them [7].

Matrix factorization is a common technique of extracting latent features from a user-item matrix. Several methods have been applied in this technique such as SVD [7], PMF [8], SVD++ [9], non negative matrix factorization [10-12]. A Recommendation system aims to predict items accurately so that users can find their preferences among thousands of products. However, the increasing size of the information presents obstacles to implementing an ideal recommendation system. In a real dataset, the sparsity ratio is so high due to users leave most items without rating. Moreover, computing prediction values depending on users' similarities would be futile. Therefore, matrix factorization methods are a suitable choice to overcome such limitations. Although matrix factorization techniques work well with sparse data, they still need improvement to increase prediction accuracy.

Consequently, social information has recently spread rapidly and many applications now use social relations [13, 14]. A social network produces new factors for further promoting the performance of the recommendation system [15, 16]. In reality, users are often influenced by their friends' choices, so for example when they try to decide on a specific movie to watch, they check those movies that are already rated by their friends. Social relationships are the main factor here and exploiting explicit relationships can enhance the prediction process. Many studies have been performed in this area, such as [17-20]. In [17] proposed two models, namely SR1 and SR2. In that study the authors integrated the matrix factorization method and social relations. Both models were implemented by computing the similarity between a target user and a set of that user's friends. The results enhanced the accuracy of the prediction. In [18] proposed a model that depended on community detection to alleviate the cold-start problem. Additionally, users' preferences may be influenced by another kind of relationship known as "friends of friends" or implicit relationships. Research [19] introduced a new system that exploited the implicit relationship of users. In that study, explicit relationships were ignored. In [20], explicit and implicit relationships were utilized with and incorporated with the user-item matrix into the probabilistic matrix factorization (PMF) method.

Although the previous studies enhanced the RS performance, still there is a shortage in terms of prediction accuracy. As mentioned in [20], the authors proposed the EISR model, which exploited implicit social relations, explicit social relations and explicit users' ratings. In that study, they overcame the drawbacks of [17, 21] by using implicit social relations as an extra source, but they overlooked the implicit feedback information that is available in the user-item matrix. Implicit feedback information is vital as it can reduce the sparsity ratio, which is considered a major problem in RS. In this paper, the limitation of [20] is overcome and the sparsity problem is alleviated by exploiting the implicit feedback information source with the other three sources that are already used in [20]. All sources are incorporated into the SVD++ method (rather than PMF method). Accordingly, the contributions of this paper include; first, exploiting four sources of information namely: explicit social relations, implicit social relations, explicit rating, and implicit feedback. Second, the sources of information are incorporated into the SVD++ method. Third, the proposed method is evaluated via three real-datasets: Ciao, FilmTrust, and Last.Fm. The remainder of this paper includes related work in the next section and the proposed method is explained in the third section followed by the results and analysis. Finally, the conclusions of the study are presented in the last section.

2. RESEARCH METHOD

The backbone of the proposed method is the SVD++ method. SVD++ is an extension of the SVD proposed by [9]. SVD++ exploits extra information called implicit feedback as well as the explicit rating. In this study four sources of information are adopted: explicit social relations, implicit social relations, rating values, and implicit feedback rating. The users' ratings are divided into two parts: training and testing. The testing part is used in the evaluation process, while the training part is utilized to implement the proposed method. Figure 1 shows a summary of the proposed method.

After determining the training set, explicit users' ratings (explicit feedback) is created in a user-item matrix. Implicit feedback can be obtained from users' ratings regardless of whether an item is of interest to the user. Accordingly, another matrix will be created with the same size as the previous user-item matrix; this matrix will be a binary matrix. The one value refers to the item rated by a user and zeros for unrated items. In [22] asserts that even if users do not like an item, there is still an opportunity for them to like an item similar to the previous one. Second, the implicit social relationships are extracted from hidden information in social networks, which may contain vital information. The implicit relationships are also called Friends of Friends. Consequently, the link prediction technique uses the resource allocation index (RAI) algorithm to derive the hidden information using as shown in (1). For example, for graph G (*V, E*), *V* denotes the users in the social network and *E* refers to the explicit relationship between the users. Let (*x, y*) be two unfriended users; RAI will compute the probability of x and y being friends by using the common users between them. If the result is greater than a certain threshold, x and y are deemed to be implicit friends.

$$pro(x,y) = \sum_{z \in |\Gamma(x)\ \cap \Gamma(y)|} \frac{1}{|\Gamma(z)|} \tag{1}$$

Where $\Gamma(x)$ and $\Gamma(y)$ refer to the sets of friends of x and y respectively. After extracting implicit social relations, a user-user matrix is created. Then the similarity between each pair of users who have explicit or implicit social relations is computed using the Pearson correlation coefficient as follows:

$$sim(u,v) = \frac{\sum_{h=1}^{n}(r_{u,jh}-\overline{r_u})(r_{v,jh}-\overline{r_v})}{\sqrt{\sum_{h=1}^{n}(r_{u,jh}-\overline{r_u})^2}\ \sqrt{\sum_{h=1}^{n}(r_{v,jh}-\overline{r_v})^2}} \tag{2}$$

$r_{u,j}$ is the rating value of item j that is rated by user u. $\overline{r_x}$ refers to the average rating of the user x, and n implies the common items between users u and v. The similarity ranges between [-1, 1]; 1 means matching between users and -1 indicates completely dissimilar users. The four sources are incorporated into the SVD++ algorithm. The SVD++ prediction formula depends on the baseline, as follows:

$$r_{u,i} = \mu + b_u + b_i + q^T\left(p_u + \frac{1}{\sqrt{|N_u|}}\sum_{j \in N_u} y_j\right) \tag{3}$$

$$\sum_{j \in N_u} y_j = \frac{\sum_{j \in N(u)} I(r_{u,j}>0)}{|N_u|} \tag{4}$$

Here, μ is the average rating of the entire dataset. b_u and b_i are the mean values of user u and item i respectively, that are derived from the observed deviations. p_u is the left orthogonal values of the user-item matrix, which implies the users. q_u implies the right orthogonal of the user-item matrix, which indicates the items. $|N_u|$ is the number of items rated by user u and y_j is the left orthogonal value of the binary matrix array. $I(r_{u,j} > 0)$ equals one if $r_{u,j}$ has a value, otherwise zero.

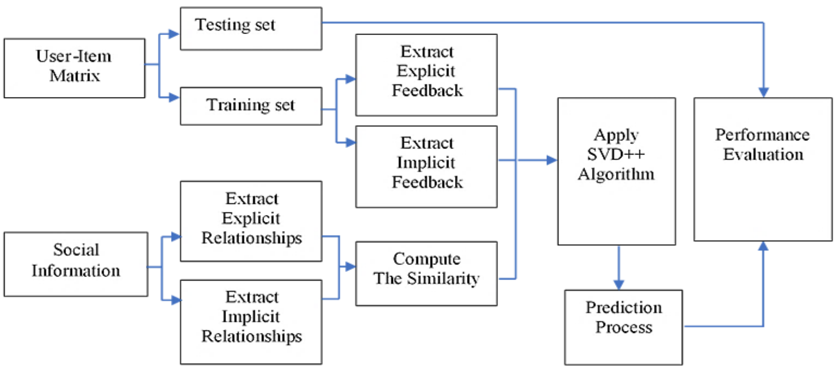

Figure 1. Overview of the proposed method

The stochastic gradient descent (SGD) algorithm is adopted to optimize and reduce the regularized square error. The optimization technique is performed to regularize the square error via the SGD algorithm. The result of square error should be close to a particular threshold; this process is called regularization. Initially, μ, b_u, and b_i are computed. Then, random values are set to vectors $p_{u,k}$ and $q_{i,k}$, where $p_{u,k}$ indicates a vector of users, u refers to the number of users, and k is the dimension value. $q_{i,k}$ implies a vector of items and i stands for the number of items. The similarity of explicit and implicit social relations is utilized to enhance and reduce the regularization error of vector p_u and the value of user-bias (b_u). As shown in (5) and (6) are used to reduce the regularization error:

$$\frac{\partial E}{\partial U_u} = \sum_{(u,i)\in K}\left(r_{u,i} - \mu - b_i - b_u - q^t\left(p_u + \frac{1}{\sqrt{|N_u|}}\sum_{j\in N_u} y_j\right)\right)^2 + \lambda\left(b_u^2 + b_i^2 + \|p_u\|^2 + \|q_i\|^2 + \|y_j\|^2\right) + \beta_e \sum_{i=1}^{u}\sum_{f\epsilon F(i)} sim(i,f)\|p_i - p_f\|_{fro}^2 + \beta_i \sum_{i=1}^{u}\sum_{f^*\epsilon F^*(i)} sim(i,f^*)\|p_i - p_{f^*}\|_{fro}^2 \tag{5}$$

$$\frac{\partial E}{\partial V_i} = \sum_{(u,i)\in K} \left(r_{ui} - \mu - b_u - b_i - q^t\left(p_u + \frac{1}{\sqrt{|N_u|}}\sum_{j\in N_u} y_j\right)\right)^2 + \lambda\left(b_i^{\,2} + b_u^{\,2} + \|p_u\|^2 + \|q_i\|^2 + \|y_j\|^2\right) \quad (6)$$

where *βe* and *βi* are constant values that control the impact degrees of the explicit and implicit relationships respectively; *F(i)* and *F*(i)* respectively denote the numbers of explicit friends and implicit friends of user *u*; *sim(x,y)* refers to the similarity value between users *x* and *y*; *K* is the number of ratings, and λ is the constant value that determines the degree of the constraint. The last two equations are utilized to find the best value for the vectors p_u, q_u, b_u, b_i and y_j. The following equations show the updating values after each loop:

$$p_u = p_u + \gamma\left(\frac{\partial E}{\partial U_u} q_i - \lambda p_u\right) \quad (7)$$

$$q_i = q_i + \gamma\left(\frac{\partial \mathrm{E}}{\partial V_i}\left(p_u + \frac{1}{\sqrt{|N_u|}}\sum_{j\in N_u} y_j\right) - \lambda q_i\right) \quad (8)$$

$$y_j = y_j + \gamma\left(\frac{\partial E}{\partial v_i}\left(\frac{q_i}{\sqrt{|N_u|}}\right) - \lambda y_j\right) \quad (9)$$

$$b_u = b_u + \gamma(\frac{\partial E}{\partial U_i} - \lambda b_u) \quad (10)$$

$$b_i = b_i + \gamma(\frac{\partial E}{\partial V_i} - \lambda b_i \quad (11)$$

where γ is the learning rate, which is a constant value that controls the regularization error during the training stage. After optimizing the values of the vectors, prediction values are computed using as shown in (3). Finally, the evaluation process is performed to compute the accuracy of the prediction by comparing the results of the proposed method with the testing part through the mean absolute error (MAE) and root mean square error (RMSE).

3. RESULTS AND ANALYSIS

In this section, the dataset, parameters, evaluation metric, and analysis of the results are explained. In subsection 3.1, the environment setup includes description of the datasets, the evaluation metrics are determinded, and the previous studies that are utilized in the comparison are addressed as well. Moreover, in subsection 3.3, the results are explained and the advantages of the proposed method are highlighted. Finally, in subsection 3.4, the impact of exploiting social relations is demonstrated.

3.1. Environment setup

Environment setup includes dataset selection, evaluation metrics, parameter settings, and training size. All of them are explained in detail. Three datasets were utilized in this study: Last.Fm, Ciao, and FilmTrust. Last.Fm was released by HetRec in 2011 in the framework of the second international workshop on information [23]. This dataset contains 2,100 users and 18,745 items. Each user rated 50 items. The data include listening counts for each item. As shown (12) was adopted to map the counting values over a range (1-5) of rating values, which was proposed by [24]. Moreover, this dataset contains a social information file on the relationships between the users.

$$\mathrm{r} = \begin{cases} \lfloor \log_{10} \mathrm{l} \rfloor + 1, & \text{if} \lfloor \log_{10} \mathrm{l} \rfloor + 1 \leq 5 \\ 5 & \text{, otherwise} \end{cases} \quad (12)$$

Here, *r* is the rating integer value, *l* the listening count, and ⌊x⌋ the operation of rounding down towards zero. In this study, to compare the proposed method with other studies, users with fewer than five relationships were removed. Therefore, 1,123 users remained. Ciao was the second dataset generated by [25]. This dataset enables users to create their relationships. It is a product review website on which users can rate and review products. This dataset consists of 7,375 users and 99,746 items. In matrix factorization methods, when a user or an item vector is not rated, the entire vector values are zero. Therefore, it becomes impossible to find the best low-rank linear representation of the user-item matrix [26]. Thus, in this study, every user with at least one relationship, and who had a rating greater than or equal to 1, was selected. Similarly, every item rated at greater than 1 was selected. Subsequently, 6,767 users with 22,229 items remained. FilmTrust, produced by [27], was the last dataset used in this study. This dataset, also containing social information, involves 1,508 users and 2,071 items. Accuracy metrics were adopted to evaluate prediction accuracy in this study; these being mean absolute error (MAE) and root mean absolute error (RMSE) [28, 29]. The following formulas compute these metrics.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| r_{u,i} - \bar{r}_{u,i} \right| \tag{13}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(r_{u,i} - \bar{r}_{u,i} \right)^2} \tag{14}$$

where *N* is the total number of the prediction, $r_{u,i}$ and $\overline{r}_{u,i}$ are the actual rating and the rating value computed by the method of item *i* and by user *u* respectively. Zero value implies an optimal result. In contrast, the high value implies lower accuracy. Our proposed method is compared with other matrix factorization studies including:
- SVD proposed by [30]: This method makes predictions by exploiting information only and it is used to compare evaluations for every dataset.
- Probabilistic matrix factorization (PMF) proposed by [8]: This method utilizes only the user-item matrix for prediction and it is used for results with every dataset.
- SVD++ proposed by [9]: This method is again only used for rating information, which is applied to all datasets.
- Social recommendation (SoReg) proposed by [17]: The author here uses explicit social information in addition to the rating information.
- EISR proposed by [20]: This study is close to our work since explicit and implicit relationships are adopted. The comparison is achieved for Last.Fm and Ciao.
- User Embedding UE-SVD++ in [22]. This is constructed and utilized for both explicit and implicit rating information with the user embedding matrix by the proposed user-wise mutual information. It was used for FilmTrust.

For impartial comparison, every method is implemented with the same parameter values. The parameter settings used in this study are demonstrated as follows:
- d: the dimensions of the latent features equalling 5 and 10.
- β_e and β_i are the weights of explicit and implicit social relationships respectively. The values of both weights are 0.05 for Last.Fm, 0.001 for Ciao and 0.005 for FilmTrust.
- λ is the regularization parameter set at 0.001 for Last.Fm and Ciao, and at 0.0001 for FilmTrust.
- α is the learning rate set at 0.05 for Last.Fm, 0.006 for Ciao and 0.015 for FilmTrust.

3.2. Discussions

The aim of this study is to show the impact of integrating social relations (explicit and implicit) with the rating information (explicit rating and implicit feedback information) into the SVD++ method to alleviate the sparcity issue. To implement the proposed method, the datasets are divided into two parts: training and testing. The training part is 80% of the entire users, which are selected randomly. The other part (20%) is for testing. The experiments are implemented 10 times and the means of the outcomes for the MAE and RMSE values are computed. Tables 1 to 3 show the results of MAE and RMSE when *d*=5 and *d*=10 and they are benchmarked with other studies applied to the Last.Fm, Ciao, and FilmTrust datasets. The best results are presented in boldface.

As shown in Tables 1 to 3, social relations have a significant impact on enhancing prediction accuracy. In the FilmTrust dataset (Table 1), the closest results to our study are UE-SVD++, where the proposed method outperformed UE-SVD++ with an improvement of about 0.02% for MAE and 0.0008% for RMSE when *d*=5. Whereas when *d*=10, the improvement is about 0.03% for MAE and 0.01% for RMSE. Moreover, the improvements in using social relations are persistent, especially by adding implicit social relations that further boost the prediction process. Therefore, EISR and the proposed method reached peak results for both metrics (RMSE and MAE) by using this source. Additionally, the proposed method achieved the best results and exceeded all the aforementioned studies where the difference in Table 2 between the proposed method and the nearest study (EISR) is 0.0017 for MAE and 0.0047 for RMSE when *d*=5. The same preference can be seen in Table 3, where the proposed method also outperformed other studies for both metrics. To sum up, the effectiveness of this study mitigates the sparsity ratio by exploiting four sources of information, namely, explicit social relations, implicit social relations, explicit users' ratings, and implicit feedback information. Therefore, the results of the tables show that the proposed method outperform the previous studies in terms of accuracy and achieve the objective of this study by further improving prediction accuracy.

Table 1. Performance comparisons for the FilmTrust dataset (80% training). The best results are presented in boldface

All users	Metrics	d=5	d=10
PMF	MAE	0.714	0.735
	RMSE	0.949	0.968
SVD	MAE	0.709	0.709
	RMSE	0.925	0.954
SVD++	MAE	0.613	0.611
	RMSE	0.804	0.802
SoReg	MAE	0.674	0.668
	RMSE	0.878	0.875
UE SVD++	MAE	0.6203	0.6280
	RMSE	0.8026	0.8120
Proposed method	MAE	**0.6072**	**0.6096**
	RMSE	**0.8020**	**0.8011**

Table 2. Performance comparisons for the Ciao dataset (80% training). The best results are presented in boldface

All users	Metrics	d=5	d=10
PMF	MAE	0.920	1.078
	RMSE	1.206	0.822
SVD	MAE	0.784	0.788
	RMSE	1.033	1.037
SVD++	MAE	0.752	0.748
	RMSE	1.013	1.001
SoReg	MAE	0.899	0.815
	RMSE	1.183	1.076
EISR	MAE	0.7288	0.7285
	RMSE	0.9612	0.9606
Proposed method	MAE	**0.7271**	**0.7267**
	RMSE	**0.9565**	**0.9557**

Table 3. Performance comparisons for the Last.Fm dataset (80% training). The best results are presented in boldface

All users	Metrics	d=5	d=10
PMF	MAE	0.427	0.426
	RMSE	0.537	0.535
SVD	MAE	0.414	0.414
	RMSE	0.534	0.533
SVD++	MAE	0.4111	0.4103
	RMSE	0.5307	0.5301
EISR	MAE	0.4065	0.4060
	RMSE	0.5263	0.5258
Proposed method	MAE	**0.4012**	**0.4006**
	RMSE	**0.5187**	**0.5177**

In Tables 4 and 5, multiple dimension values are applied to check the performance of the proposed method in different d values with the experiments being executed when d=10, 30, 50, 70, and 90. For each dimension value, MAE and RMSE are calculated. In Table 4, the results show that the accuracy is significantly improved when the dimension value increases, where increasing the dimension value means more information added to the vectors p and q to compute the prediction. The best results are achieved when d equals 70, where the result of the proposed method is 0.5166 for RMSE and 0.3989 for MAE. However, the accuracy slightly deteriorates to 0.5174 and 0.3994 for RMSE and MAE respectively when d is 90. The same scenario can be seen in other studies (SVD, SVD++, E-SVD++) so that for various dimension values, our proposed method outperforms other studies. Table 5 shows multiple values of d for the Ciao dataset. The same scheme can be seen as in Table 4, where RMSE and MAE are reduced (improving accuracy) when d increases. However, the difference here is in the peak values of RMSE and MAE when d=50. The results decline slightly after 50 (when d=70), RMSE and MAE are 0.9574 and 0.7271 respectively. When d=50, the results are 0.9542 for RMSE and 0.7262 for MAE. In summary, for all datasets and metrics that are used in this study, the proposed method accomplished the best results in terms of accuracy for all dimension values. Consequently, using social information for both sources (explicit and implicit) with implicit feedback has significant improvement, where the results reveal the outperformance of the proposed method in all dimension values.

Table 4. Performance comparisons for multidimensional values for Last.Fm (80% training)

	Matrices	Methods			
		SVD	SVD++	E-SVD++	Proposed method
d=10	RMSE	0.533	0.5321	0.5257	0.5177
	MAE	0.4140	0.4123	0.4033	0.4006
d=30	RMSE	0.5328	0.5316	0.5248	0.5175
	MAE	0.4138	0.4118	0.4025	0.3998
d=50	RMSE	0.5325	0.5293	0.5224	0.5169
	MAE	0.4134	0.4092	0.4012	0.3992
d=70	RMSE	0.5315	0.5289	0.5221	**0.5166**
	MAE	0.4128	0.4087	0.3994	**0.3989**
d=90	RMSE	0.5328	0.5295	0.5242	0.5174
	MAE	0.4135	0.4069	0.4012	0.3994

Table 5. Performance comparisons for multidimensional values for the Ciao dataset (80% training)

	Matrices	Methods			
		SVD	SVD++	E-SVD++	Proposed method
d=10	RMSE	1.037	1.001	0.9571	0.9557
	MAE	0.788	0.7488	0.7292	0.7267
d=30	RMSE	1.032	0.9972	0.9564	0.9551
	MAE	0.7874	0.7486	0.7288	0.7265
d=50	RMSE	1.021	0.996	0.9558	**0.9542**
	MAE	0.7869	0.7481	0.7286	**0.7262**
d=70	RMSE	1.016	0.9952	0.9577	0.9561
	MAE	0.784	0.7483	0.7289	0.7266
d=90	RMSE	1.024	0.996	0.9585	0.9574
	MAE	0.787	0.7485	0.7294	0.7271

3.3. Impact of changing beta values

Figures 2, 3 and 4 show the impact of the beta parameter in different values for FilmTrust, Ciao, and Last.Fm datasets. The beta parameter is used to control the degree of explicit and implicit social relations. The beta value is multiplied by the similarity value for each pair of explicit or implicit social relations. As seen in these figures, the RMSE value changes when the beta value does. For example, Figure 2 shows the RMSE results for different values. It can be seen that the RMSE achieved the worst result (0.8028) when beta equals 0.0001. However, the results improved when beta increased. The best result (0.8011) is registered when beta equals 0.05. Afterward, the RMSE result again starts to gradually increase (be less accurate) when the beta value increases. The same thing can be seen in Figures 3 and 4 where beta values affect the RMSE result. Hence, explicit and implicit social relations have a significant role to enhance the prediction. This role is shown by the beta value. Thus, the beta value can enhance prediction accuracy for a specific value. When the beta value is greater or smaller than a threshold, the accuracy is negatively affected.

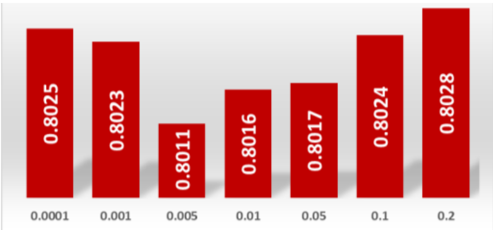

Figure 2. Impact of beta values on RMSE result for FilmTrust

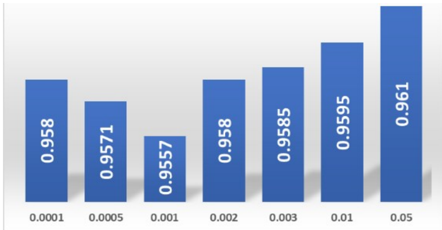

Figure 3. Impact of beta values on RMSE result for Ciao

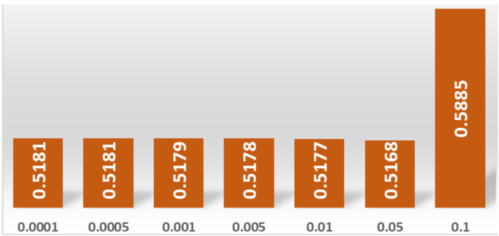

Figure 4. Impact of beta values on RMSE result for Last.Fm

4. CONCLUSION

In this paper, social relations were exploited in two channels (explicit and implicit relationships) to improve recommendation performance. Implicit relationships were extracted by applying the resource allocation index (RAI) which predicts hidden information in the social graph. Four sources of information, namely, explicit social relations, implicit social relations, explicit rating, and implicit rating feedback are incorporated into the SVD++ method to alleviate the sparsity problem. Social relations (implicit and explicit relationships) are utilized to help active users to find their preference items by computing similarities between active users and their social relations. Three real-world datasets were used, namely Last.Fm, FilmTrust and Ciao. The results of the experiment revealed that social relations have an obvious impact to boost the prediction accuracy. Moreover, the results revealed that our proposed method is superior to other studies such as SVD, SVD++, PMF, EU-SVD++, SocReg, and EISR in terms of accuracy. The proposed method significantly improved the prediction by exploiting users' relationships as it considers that all friends have similar tastes. Accordingly, classifying friends into various groups and computing the similarity of each group separately may enhance predictions and achieve more precise results. Eventually, the proposed method adopted the pearson correlation to compute the similarity. Different similarity measures can be employed to further improve the results.

REFERENCES

[1] S. Potluri and K. S. Rao, "Improved quality of service-based cloud service ranking and recommendation model," *TELKOMNIKA Telecommunication Computing Electronics and Control,* vol. 18, no. 3, pp. 1252-1258, Jun. 2020.

[2] M. Kunaver and T. Požrl, "Knowle dge-Base d Systems Diversity in recommender systems–A survey," vol. 123, pp. 154-162, 2017, doi: 10.1016/j.knosys.2017.02.009.

[3] I. Prasetyaningrum, *et al.*, "Application of recommendation system with AHP method and sentiment analysis," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 18, no. 3, 1343-1353, Jun. 2020.

[4] H.-F. Wang and C.-T. Wu, "A strategy-oriented operation module for recommender systems in E-commerce," *Comput. Oper. Res.*, vol. 39, no. 8, pp. 1837–1849, Aug. 2012.

[5] T. Jiule, *et al.*, "Collaborative filtering recommendation model considering integration of user rating and attribute similarity," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 14, no. 2, pp. 208-216, 2016.

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

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285-295.

[8] A. Mnih, *et al.*, "Probabilistic matrix factorization," *Advances in neural information processing systems*, 2008.

[9] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 263-272.

[10] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowledge-Based Syst.*, vol. 97, pp. 188-202, 2016.

[11] Y. Cao, W. Li, and D. Zheng, "An improved neighborhood-aware unified probabilistic matrix factorization recommendation," *Wirel. Pers. Commun.*, vol. 102, no. 4, pp. 3121-3140, Jan. 2018.

[12] H. Parvin, *et al.*, "A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method," *Knowledge-Based Syst.*, vol. 166, pp. 92-107, 2019.

[13] K. Surendro, *et al.*, "Integrated Social Media Knowledge Capture in Medical Domain of Indonesia," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 4, pp. 1846-1856, 2018.

[14] R. Kumar, *et al.*, "Social Popularity based SVD++ Recommender System," *Int. J. Comput. Appl.*, vol. 87, no. 14, 2014.

[15] A. M. A. Al-Sabaawi, *et al.*, "Exploiting implicit social relationships via dimension reduction to improve recommendation system performance," *PLoS One*, vol. 15, no. 4, 2020, [Online]. Available: https://doi.org/10.1371/journal.pone.0231457.

[16] A. Anandhan, L. Shuib, M. A. Ismail, and G. Mujtaba, "Social media recommender systems: review and open research issues," *IEEE Access*, vol. 6, pp. 15608-15628, Feb. 2018, doi: 10.1109/ACCESS.2018.2810062.

[17] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 287-296.

[18] M. Hussein, H. Naji, and W. Bhaya, "Influential nodes based alleviation of user cold-start problem in recommendation system," *Res. J. Appl. Sci.*, vol. 11, pp. 1107-1114, Jan. 2016, doi: 10.3923/rjasci.2016.1107.1114.

[19] Y. Zhang, W. Chen, and Z. Yin, "Collaborative filtering with social regularization for TV program recommendation," *Knowledge-Based Syst.*, vol. 54, pp. 310-317, 2013.

[20] W. Reafee, N. Salim, and A. Khan, "The power of implicit social relation in rating prediction of social recommender systems," *PLoS One*, vol. 11, no. 5, 2016.

[21] H. Ma, "An experimental study on implicit social recommendation," *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013, pp. 73-82.

[22] W. Shi, L. Wang, and J. Qin, "User embedding for rating prediction in SVD++-based collaborative filtering," *Symmetry (Basel).*, vol. 12, no. 1, p. 121, 2020.

[23] I. Cantador, *et al.*, "Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011)," *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 387-388.

[24] J. Liu, C. Wu, and W. Liu, "Bayesian probabilistic matrix factorization with social relations and item contents for recommendation," *Decis. Support Syst.*, vol. 55, no. 3, pp. 838-850, Jun. 2013.

[25] J. Tang, H. Gao, H. Liu, and A. Das Sarma, "eTrust: Understanding trust evolution in an online world," *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 253-261.

[26] R. Chen, *et al.*, "A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, pp. 64301-64320, 2018.

[27] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," *Twenty-Third International Joint Conference on Artificial Intelligence*, Jan. 2013.

[28] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating collaborative filtering recommender algorithms: a survey," *IEEE access*, vol. 6, pp. 74003-74024, 2018.

[29] A. Da'u, N. Salim, I. Rabiu, and A. Osman, "Recommendation system exploiting aspect-based opinion mining with deep learning method," *Inf. Sci. (Ny).*, vol. 512, pp. 1279-1292, Feb. 2020.

[30] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," *Minnesota Univ Minneapolis Dept of Computer Science*, 2000.