International Journal of Innovative Computing, Information and Control
Volume **17**, Number **2**, April **2021**

ICIC International ©2021 ISSN 1349-4198
pp. **383–396**

SVD++ AND CLUSTERING APPROACHES TO ALLEVIATING THE COLD-START PROBLEM FOR RECOMMENDATION SYSTEMS

Ali Mohsin Ahmed Al-Sabaawi[1,3], Hacer Karacan[2] and Yusuf Erkan Yenice[1]

[1]Department of Electrical, Electronic and Computer Engineering
Aksaray University
Bahçe Saray, Aksaray University, Campus Road, Merkez/Aksaray 68100, Turkey
alisabaawi5@gmail.com

[2]Department of Computer Engineering
Gazi University
Emniyet Mah, Bandırma Cd No:6/1, Yenimahalle/Ankara 06560, Turkey
hacerkaracan@gmail.com

[3]Department of Computer Science
Mosul University
Al Majmoaa Street, Mosul 41002, Iraq

Received September 2020; revised January 2021

Abstract. *Recommendation systems provide a solution to tackle information overload problem. These systems have several limitations, one of which is cold-start users. In this article, a new method is proposed to overcome the cold-start user problem. The main idea of this study is to apply a clustering technique using trust relations and rating information to compute the weights. First, the implicit relations are determined, and then the similarity is computed for each pair of explicit and implicit relations. Second, confidence values are determined through an information rating by dividing the number of common items for each pair of users by the number of items that have been rated by the first user of this pair. Furthermore, the similarity and confidence values are integrated to produce weight values, and then the distance values are inferred. Additionally, the partitioning around medoids clustering algorithm is adopted to cluster the users into groups according to their computed distances. Moreover, the Singular Value Decomposition Plus (SVD++) method is employed for each cluster to predict the items for cold-start users. Eventually, the proposed method is evaluated with two real-world datasets. The results reveal that the proposed method outperforms the state-of-the-art trust methods in terms of prediction accuracy.*

Keywords: Recommendation systems, Cold-start users, Clustering, SVD++, PAM, Social relations

1. **Introduction.** With the growing utilization of E-commerce, huge amounts of information are available online, which leads to information overload. Therefore, finding preference items among massive choices is a really hard job. Recommendation Systems (RSs) are a successful solution to overcome this problem. RSs provide sets of helpful items that are similar to users' tastes. Basically, RSs face some challenges that affect prediction accuracy. One of the most common challenges is the cold-start problem which includes cold-start users and cold-start items. A cold-start item is a new item that has been introduced to a system. It is hard to predict the customer preference for this item, or the item will be at the end of the prediction list due to the lack of feedback given on this item

DOI: 10.24507/ijicic.17.02.383

(still new). Likewise, the cold-start user problem occurs when a new user registers in the system or when a user has few ratings (less than a threshold) [1]. The main factor of the prediction process is users' ratings and an RS utilizes this factor to compute the similarity among users and produce a set of items for the targeted user. However, when the rating history is poor, it is exhausting to compute the similarity, and thus the performance of the prediction is negatively affected. Recently, the cold-start user problem has attracted many researchers to alleviate such limitations.

Collaborative Filtering (CF) is one of the most common types of RS. It has two versions: memory-based and model-based [2]. In the memory-based method, the RS exploits users' feedback ratings to compute the similarity among users/items. Whereas in the model-based method, the users' behavioral patterns are adopted to predict a list of items for them using data mining methods and machine learning techniques such as Matrix Factorization (MF) methods and clustering [3,4]. MF is a prevalent technique that has been used in many studies. The size of the rating matrix can be reduced to a small dimension by finding the best latent feature of the user-item matrix [5]. Several methods have been applied in MF techniques such as Singular Value Decomposition (SVD) [6], Probabilistic MF [7], and SVD++ [8]. In addition, clustering techniques are utilized also to enhance the performance of CF model-based approach [3]. Clustering means distributing users into groups/clusters according to the distances among them. An RS can exploit this technique by recommending items that were rated by users of a particular cluster to which the targeted user belongs. Various clustering approaches have been adopted in the literature such as the k-means [9,10], Partitioning Around Medoids (PAM) [11], the Fuzzy C-means [12], and hierarchical clustering [13].

All studies aim to improve the prediction accuracy. To achieve this goal, many types of information have been exploited to boost predictions such as demographic information and social relations. Demographic information refers to the information of users that is collected during their registration. By exploiting this information, the system can conclude the user's tastes even for a cold-start user. Although demographic information provides a good solution to alleviate the cold-start problem, it is not always available in the data. Another aspect is social relations, which provide substantial information. They include explicit and implicit relations.

Many studies have employed this source to mitigate RSs problems [2,11-16]. Some studies used rating information to create user communities, whereas others used social information to establish the communities. The problem is determining the influence node (user) to be the center of a cluster that has a greater number of connections with other users in a community. In [11], they adopted explicit and implicit relations and used a clustering technique to find the influential nodes in each cluster. The top-n influential nodes were then ranked to serve cold-start users. The authors included implicit relations by using a transitive technique. However, they proposed that all relations between users have similar weights, and they applied the PAM clustering technique based on social relations only. Moreover, in that study [11] along with the previous limitation, the authors followed a traditional technique to find clusters' centers that depends on the closest distance between the nodes only. In [16], the authors integrated the MF technique with social trust information to mitigate the cold-start user problem.

Although the previous studies [11] and [16] alleviated the cold-start user problem by exploiting social relations along with rating values, they neglected confidence values which are an essential source of information that can reduce the intensity of the cold-start user problem. Some datasets like FilmTrust have a towering deficiency in terms of social relations; thus, applying a clustering process using such relations only will not be sufficient and it will disregard many users as outliers. Accordingly, in this article, a hybrid method

that integrates the SVD++ method and PAM clustering technique is proposed to overcome the above problems and alleviate the impact of the cold-start user problem. The clustering is done not only by using social relations, but also by employing confidence values that can be extracted via users' ratings. These values can provide two advantages. First, social relations sparseness limitation is overcome by integrating social relations and confidence values. Second, confidence values are utilized to create weight values between each pair of users so the limitation of considering all users have the same weights is surmounted. The PAM algorithm is employed for the intent to exploit users' attributes that can be withdrawn through users' ratings and social information to group them afterward into several clusters; subsequently, cold-start users will be determined in each cluster. Finally, the SVD++ method is applied to predict items. The contributions of this paper can be summarized as follows.

I. The weight of each node is computed by incorporating social relations and confidence values.
II. The selection of the medoid node in each cluster is determined based on the node density in terms of relationships and distance from the center.
III. The PAM and SVD++ methods are integrated into one model.
IV. The proposed model is evaluated using two real-world standard datasets and the results are compared with those of the state-of-the-art approaches.

The rest of this article includes the following. The SVD++ method, PAM algorithm, extraction of social relations, confidence values, and the proposed method are demonstrated in Section 2. Section 3 is dedicated to exhibiting the preparing for evaluation. The results and discussions are illustrated in Section 4. Finally, the concluding points are given in the last section of this study.

2. **Methods.** In this study, two main sources are employed to produce the proposed model: social relations and users' ratings. First, the clustering technique is adopted to group users according to their relations and ratings. Second, the SVD++ method utilized users' rating explicit and implicit feedback to compute the missing predictions for cold-start users in each cluster. The following subsection demonstrates the methods of SVD++ and PAM in Sections 2.1 and 2.2 respectively. Moreover, the social relations that involve explicit and implicit relations are explained in Section 2.3. Extraction confidence values are illustrated in Section 2.4. Ultimately, the proposed method that employs all data sources is well-illustrated in Section 2.5.

2.1. **Singular Value Decomposition Plus (SVD++).** SVD++ was produced by [8]. It is an expansion of the SVD method. The implicit rating information is the data that is considered in SVD++. Recently, many studies have employed this approach [17,18] due to its ability to enhance accuracy. The main idea of this method is reducing the amount of rating information (user-item matrix), where the new user matrix and the new item matrix are derived by detecting the latent features from the user-item matrix. Then the prediction value is computed as follows:

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

$$b_i = \frac{\left(\sum_{u \in R(i)} r_{u,i} - \mu \right)}{\lambda_1 + |R_i|} \tag{2}$$

$$b_u = \frac{\left(\sum_{u \in R(u)} r_{u,i} - \mu - b_i\right)}{\lambda_2 + |R_u|} \tag{3}$$

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

where μ is the mean value of all the entire ratings; and b_u and b_i are the observed deviations from the μ of user u and item i respectively. p_u and q_i indicate latent factor vectors of the user u and the item i respectively. $|R_u|$ is the number of users who rated a specific item, and $|R_i|$ is the number of items that were rated by a set of users, while y_i is the latent influence vector of users' ratings; also, it is called an implicit feedback set. If $r_{u,i} > 0$, $I(r_{u,j} > 0)$ is one, otherwise it is 0. λ_1 and λ_2 are two regularization parameters. In this study, the SVD++ method is adopted in each cluster to predict items for cold-start users.

2.2. **Partitioning Around Medoids (PAM) algorithm.** The clustering technique in a recommendation system is a process that distributes the users of a given dataset into many groups according to the distances between them. In other words, users similar in terms of ratings, relations or any other features are gathered together in one community. Subsequently, items that interest a particular cluster are recommended to a user who belongs to that cluster. Many methods such as the k-means, fuzzy clustering, and PAM [12,19] have been used to implement clustering. In this study, the PAM algorithm is adopted. PAM is a conventional partitioning clustering algorithm based on the k-medoids technique. A medoid or a center is an object of a particular cluster where the distance between each object and the selected medoid is the minimum. The k-medoid is more powerfully-built than the k-means as the former is less sensitive to outliers and noise than the latter.

Algorithm 1: PAM

1- Select k medoids randomly, where $k < n$, and n is the number of nodes in the whole dataset. Each medoid indicates the center of a cluster.
2- Apply the distance measure (the Euclidean distance, the Manhattan distance, or any other distance method) to finding the distance (cost) between each point and the selected medoids.
3- Assign each node to the closest medoid.
4- Compute the total cost of all nodes.
5- Select a new medoid from the non-medoid objects randomly.
6- Compute the new cost (cost_2) using the similarity measure.
7- Difference = $\text{cost}_2 - \text{cost}_1$
8- If the difference < 0, swap the new medoid with the old one.
9- Repeat steps 2-8 until the medoids do not change anymore.

2.3. **Extraction of social relations.** Social relations are usually used in RS to alleviate the sparsity and cold-start problems and improve the accuracy of similarities between users [20]. In this study, the similarity values are utilized to cluster users into a set of communities on the basis of social relations between them. In other words, if two users (trustors) have a similar set of trustees, there is a high likelihood that both users have similar tastes, and they will be consequently in the same community. Additionally, social relations involve two types: explicit and implicit. Therefore, in this study, both types are adopted to group users on the basis of their relations, and the following subsections demonstrate the extraction process of these types.

2.3.1. *Extraction of explicit social relations.* Explicit relations are obtained directly from a dataset. Usually, these data include two columns: trustor and trustee. Based on these columns, a binary user-user matrix is created: trustor users are the rows, and trustee users are the columns. The one values are placed when there are interactions between trustor and trustee users; otherwise place zero. However, explicit relations are few and counting on these relations to compute the common features among users will lead to the exclusion of the majority of them from the process. Subsequently, the clustering process gives inaccurate results, since most users remain out of clusters as outliers. Therefore, implicit relations are extracted and added to explicit ones to increase the relations among users.

2.3.2. *Extraction of implicit social relations.* Implicit social relations are a substantial source used to enrich social networks with more information. These relations can be concluded based on explicit relations. Several properties are dedicated to developing implicit trust relations such as asymmetry, transitivity, dynamicity, and context-dependence. In this study, the transitivity technique is followed to extract implicit trust relations. For example, assume that user a trusts user b, and user b trusts user c, then undoubtedly there is a high likelihood of establishing a new relationship (implicit relation) between a and c.

Implicit social relations are added to the binary user-user matrix that was created when we collected explicit social relations. Like explicit relations, if there is an implicit relationship between a particular pair of users the one value is placed, or zero otherwise. After implicit and explicit trust relations are attained, the similarity value between each pair of the explicit and/or implicit user is computed. The Jaccard binary measure is employed to compute the similarity by using Equation (5).

$$sim_{u,v} = \frac{x}{x + y + z} \tag{5}$$

where x is the number of common trusted users between users u and v. Whereas y indicates the number of trusted users for user u only, and z refers to the number of trusted users for user v only.

2.4. **Confidence values.** Although much information can be extracted from the social network by applying the transitive technique, some datasets like FilmTrust have poor data in terms of social relations, namely: 1,358 explicit relations and 73,057 implicit ones. Clustering users on the basis of explicit and implicit social relations leads to ignoring many users as outliers. Thus, more sources are required to include more users in the clustering technique. So, rating information is employed to compute confidence values or trust statements between each pair of users as follows:

$$T_{u,v} = A_{u,v}/A_u \tag{6}$$

$A_{u,v}$ is the number of common items that were rated by users u and v, and A_u denotes the number of items rated by user u.

Then, the adjusted weight values can be computed by integrating the similarity and confidence values as follows:

$$w_{u,v} = \begin{cases} \dfrac{sim_{u,v} + T_{u,v}}{2}, & sim_{u,v} \neq 0, \text{ and } T_{u,v} \neq 0 \\ sim_{u,v}, & sim_{u,v} \neq 0, \text{ and } T_{u,v} = 0 \\ T_{u,v}, & T_{u,v} \neq 0, \text{ and } sim_{u,v} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

As presented, $sim_{u,v}$ denotes the similarity between users u and v, which is computed using Equation (5). $T_{u,v}$ is the confidence value between users u and v and it is computed by Equation (6). Weight values are calculated using Equation (7). In this equation, social relations and confidence values are integrated for each pair of users. If a pair of users has a similarity value of social relations and confidence value, the first option of Equation (7) implements. While if it has a similarity value only, the weight value assigns the similarity value. Whereas if the pair has trust value only, the weight assigns the trust value. Finally, if there is neither a trust value nor a similarity value, the weight will be zero. After obtaining weight values, the clustering technique can be implemented to distribute users into several clusters according to the computed weight values.

2.5. **Implementation of the proposed method.** In this study, the datasets are divided into two parts: training and testing according to a threshold that determines cold-start users for the testing phase and non-cold-start users for the training phase. Two main sources are employed in the proposed model: social relations and users' ratings. First, the social relations and ratings of non-cold-start users are exploited to compute the similarity and confidence values respectively. Then, the PAM clustering method is adopted to group users into several communities. Second, again users' ratings are utilized, which include explicit and implicit feedback, to compute the missing predictions for cold-start users in each cluster. Figure 1 shows the details of the proposed method.

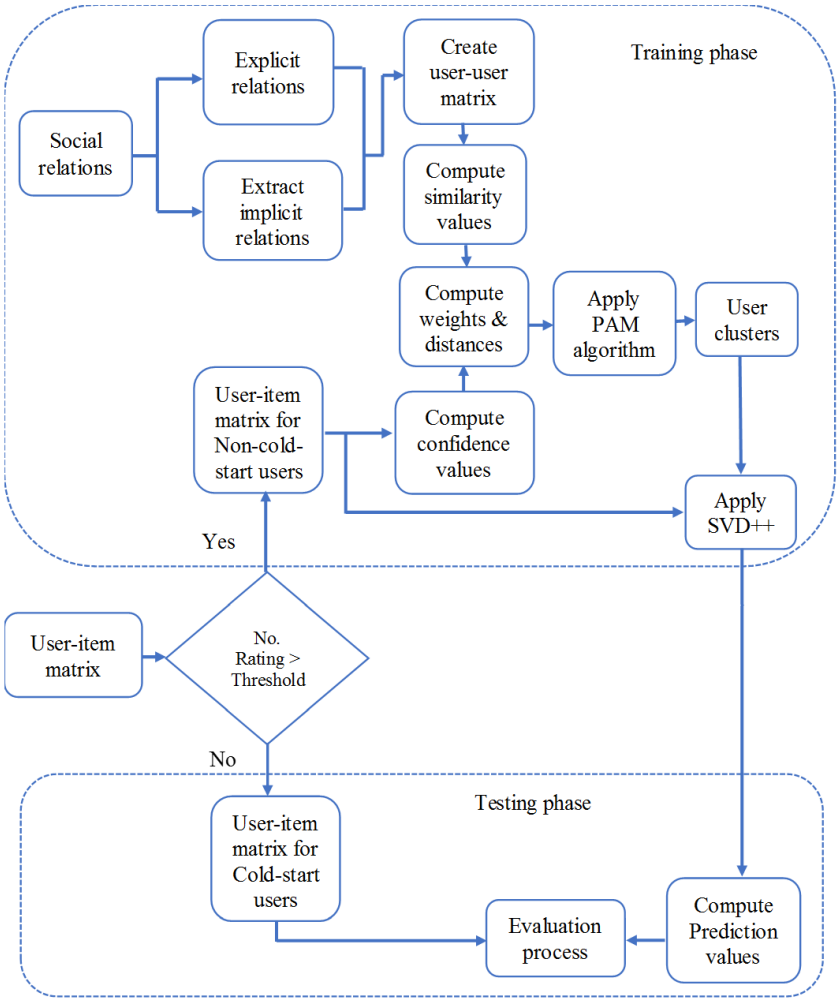

FIGURE 1. Overview of the proposed model

The following steps explain the proposed method.

Step 1: The dataset is divided based on the rating value of each user into two phases. If the rating value exceeds a threshold, the user is considered a non-cold-start user, and his rating is used in the training phase; otherwise, he is assigned as a cold-start user in the testing phase.

Step 2: In the training phase, a binary user-user matrix is created to represent explicit and implicit social relations. If there is a relation between two users, one value is placed; otherwise it is zero.

Step 3: The similarity of explicit and implicit social relations is computed using Equation (5).

Step 4: To decrease clusters' outliers, rating information is employed to compute confidence values among each pair of users by using Equation (6). A new user-user matrix is created for the confidence values.

Step 5: The confidence values and similarity values are incorporated to compute weight values by using Equation (7).

Step 6: After acquiring the weights, the distances are calculated by subtracting the weight values from one. Then, the PAM clustering algorithm is adopted to cluster the users according to the computed distances. The value of k (the number of clusters) is determined by trial and error. The MAE and RMSE metrics are computed for different k values, then k value is selected depending on the best result among these metrics. Moreover, two conditions are dedicated to determining the medoids in the PAM algorithm. First, the condition uses the greatest number of relations of the object to be a medoid. While the second condition finds the minimum distance between the medoid and the objects on a particular cluster. The output of this step is users' clusters.

Step 7: The stochastic gradient descent method is utilized to optimize the result of the SVD++ method by using the following Equations (9)-(13), which are applied to tune the values of p_u, q_i, y_j, b_u, and b_i respectively.

Step 8: In the testing phase, the SVD++ method is implemented in each cluster to compute the predictions for cold-start users through Equation (1). After applying SVD++ on each cluster, an average is calculated to attain the final outcome.

Step 9: The evaluation process is done by comparing the computed prediction values with the actual values in the dataset.

In the training phase, the prediction value is computed by utilizing the values of p_u, q_i, y_j, b_u, and b_i factors, then it is compared with the actual rating value to compute the difference (regularization error) between them. Additionally, stochastic gradient descent is a well-known optimization method, which is applied for several loops to reduce the regularization error and update the values of the aforementioned factors. Consequently, the prediction value will be too close to the actual rating. After tuning the values of these factors, they can be utilized in the testing phase to predict items for cold-start users.

According to the stochastic gradient descent method, there is a rule that should be followed to update the factors. These rules depend on some parameters, such as learning rate value (γ), the regularization error (E), constant value to avoid overfitting (λ), and the old values of the factors. In Equation (8), the regularization error (E) is computed in each loop, then it is used in Equations (9)-(13) to update and tune the values of p_u, q_i, y_j, b_u, and b_i respectively. Equations (9) and (10) are used to update p_u and q_i latent factor vectors respectively. Equation (11) is used to update the implicit feedback set, which is represented by y_j. Finally, Equations (12) and (13) are adopted to update the observed deviations values of user u and item i respectively.

$$E = \sum_{(u,i)\in m} \left(r_{ui} - \mu - b_u - b_i - {q_i}^t \left(p_u + \frac{1}{\sqrt{|R_u|}} \sum_{j\in R_u} y_i \right) \right)^2 + \lambda \left({b_i}^2 + {b_u}^2 + \|p_u\|^2 + \|q_i\|^2 + \|y_j\|^2 \right) \quad (8)$$

$$p_u = p_u + \gamma \left(E q_i - \lambda p_u \right) \quad (9)$$

$$q_i = q_i + \gamma \left(E \left(p_u + \frac{1}{\sqrt{|R_u|}} \sum_{j\in R_u} y_i \right) - \lambda q_i \right) \quad (10)$$

$$y_j = y_j + \gamma \left(E \left(\frac{q_i}{\sqrt{|R_u|}} \right) - \lambda y_j \right) \quad (11)$$

$$b_u = b_u + \gamma (E - \lambda b_u) \quad (12)$$

$$b_i = b_i + \gamma (E - \lambda b_i) \quad (13)$$

E is the regularization error value, $\lambda > 0$ is utilized to set the degree of the constraint used to avoid over-fitting, γ is the learning rate, and $\|x\|$ denotes the Frobenius norm.

3. **Experimental Setting.** The proposed model is applied to real-world datasets that are widely used in recommendation systems, namely, Ciao and FilmTrust. Both datasets have rating and trust relationship files. Ciao is a general item dataset that is comprised of 7,375 users and 99,746 items, and the rating scale is 1-5 [21]. Due to the memory consumption, each item with less than 2 ratings is removed, and thus there are 22,229 remaining items. FilmTrust is a film website, it involves 1,508 users and 2,071 items. The rating scale of this dataset includes 8 levels from 0.5-4 with a step size of 0.5 [22], where 0.5 indicates the lowest preference, 4 implies the highest preference for an item rated by a particular user. Table 1 shows the details of the datasets.

Table 1. Statistics of the FilmTrust and Ciao

Dataset	**FilmTrust**	**Ciao**
# users	1,508	6,767
# items	2,071	22,229
# ratings	35,494	185,759
%Density	1.14	0.0012
# explicit relations	1,853	111,780
# implicit relations	73,057	54,056,070

Regarding the evaluation metrics we evaluated our experimental results by using statistical accuracy metrics, and we assessed the performance by comparing the predicted value with the actual user rating in a dataset. The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are widely used metrics to compute the prediction accuracy. The ideal result is zero which presents high accuracy for both metrics.

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |r_{u,i} - \bar{r}_{u,i}| \quad (14)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(r_{u,i} - \bar{r}_{u,i} \right)^2} \quad (15)$$

where, N is the total number of predictions, $r_{u,i}$ is the actual value of the dataset for item i that was given by user u, and $\bar{r}_{u,i}$ is the result of the prediction system.

Additionally, the implementation of this study depends on setting some parameters to achieve the best results. The parameters values vary from dataset to another according to the density ratio, the distribution of ratings among users, and the applied method. These parameters are determined by trial and error for all of them except d value, which is selected by state-of-the-art studies. These parameters are shown in Table 2.

Table 2. Parameter setup

Dataset	**Ciao**	**FilmTrust**
γ	0.01	0.07
λ	0.01	0.2
λ_1 and λ_2	4	20, 15
d	5, 10	5, 10
Number of iterations	5	5

4. **Results and Discussions.** The proposed method is evaluated through four implementations that are shown in the following subsections. The next subsection is dedicated to finding the best cluster of users for both datasets. The second subsection includes the comparison between the results of the proposed method with the results of the state-of-the-art methods in terms of accuracy. While the third implementation demonstrates the impact of utilizing implicit relations. Finally, the last subsection shows the impact of using confidence values. Additionally, the proposed method is evaluated by comparing its results with those of several state-of-the-art approaches, including RSTE [4], socialMF [23], SocialRec [24], and TrustSVD [15]. Moreover, the proposed method is compared with two recent studies including Trust ANLF [16] and RTARS [25] in terms of MAE and RMSE. The benchmarks were achieved with the aforementioned studies under the same conditions, which include the number of users, latent dimension values, and the number of iterations. The test was conducted for cold-start users. In this study, users were considered to be cold-start ones if they have less than 5 item ratings.

4.1. **Determining the best cluster.** Tables 3 and 4 show the results of the MAE and RMSE for different numbers of clusters. For example, 7 clusters mean that the users were distributed into 7 groups according to the distance between a user and the center of these clusters. As shown in Table 3, for the Ciao dataset, the results of the MAE and RMSE vary depending on the number of clusters, the number of normal users (non-cold-start users), and the number of cold-start users in each cluster. The best result of the MAE is 0.7126, and that of the RMSE is 0.9221 when the number of clusters is 15. The same scenario can be seen in Table 4, but the best results are produced when the number of clusters is 7 for both metrics.

Table 3. Accuracy results of various numbers of clusters for Ciao

No. of clusters	**MAE**	**RMSE**
5	0.7268	0.9333
7	0.7200	0.9259
9	0.7186	0.9346
11	0.7213	0.9312
13	0.7278	0.9258
15	**0.7126**	**0.9221**
17	0.7475	0.9614

TABLE 4. Accuracy results of various numbers of clusters for FilmTrust

No. of clusters	**MAE**	**RMSE**
5	0.6273	0.7797
7	**0.5960**	**0.7706**
9	0.6055	0.7915
11	0.6151	0.8107
13	0.6188	0.8126
15	0.6358	0.8206
17	0.6574	0.8383

4.2. **Benchmarking with previous studies.** Using social information only does not achieve good results, especially for datasets that have few social relations such as FilmTrust. Accordingly, this study overcame this limitation by utilizing confidence values along with social relations to reduce the sparsity to a reasonable level and decrease the outliers (users) during the clustering process by assigning them to clusters based on their social relations and confidence values. Additionally, to conduct an efficient evaluation, the comparison used trust-based methods that also exploited the social relations and some of them used rating information as well. The proposed method is also compared with the SVD++ method. Tables 5 and 6 report the results of the MAE and RMSE for two latent factor values of (d) 5 and 10. The evaluation results are given for the Ciao and FilmTrust datasets. The best results are displayed by using Boldface and (*) indicates the second-best results. The results revealed that extra sources, such as social relations and confidence values have an obvious role to improve prediction accuracy. With the exception of RSTE, all studies that used social relations outperformed the SVD++ method in terms of MAE and RMSE. This fact demonstrates that social relations and users' ratings provide substantial sources to boost prediction accuracy especially with cold-start users who have poor rating history. It is better than working with users' ratings as the sole source of information.

TABLE 5. Performance comparison for cold-start users for Ciao. Boldface implies best results and (*) indicates the second-best results.

	Metrics	SVD++	RSTE	SocialRec	Trust SVD	Trust ANLF	Proposed model
$d = 5$	MAE	0.759	0.957	0.789	0.729*	–	**0.7144**
	RMSE	1.039	1.113	0.998	0.953*	–	**0.9226**
$d = 10$	MAE	0.749	0.803	0.730	0.721	0.716*	**0.7126**
	RMSE	1.020	1.014	1.031	0.962	0.928*	**0.9221**

TABLE 6. Performance comparison for cold-start users for FilmTrust. Boldface implies best results and (*) indicates the second-best results.

	Metrics	SVD++	RSTE	SocialRec	Trust SVD	Trust ANLF	RTARS	Proposed model
$d = 5$	MAE	0.677	0.680	0.670	0.661	–	0.599*	**0.5958**
	RMSE	0.897	0.884	0.857	0.853	–	0.774*	**0.7703**
$d = 10$	MAE	0.680	0.674	0.668	0.663	0.607	0.599*	**0.596**
	RMSE	0.905	0.900	0.897	0.853	0.784	0.774*	**0.7706**

Equally important, as exhibited in Table 5, our method achieved the best results. The MAE is improved by approximately 0.47% and the RMSE is improved by approximately 0.64%, when (d) is 10 on the Ciao dataset compared with the second-best results (TrustANLF). Likewise, in Table 6, the proposed method outperforms the other studies in all cases. For instance, the closest results to our model are given by RTARS, the proposed method is obviously superior. The results are approximately 0.50% and 0.44% in terms of the MAE and RMSE respectively when d is 10 in FilmTrust. A similar improvement can be seen when (d) equals 5 on the same dataset. The improvement is 53% for MAE and 48% for RMSE. Hence, for users who have few ratings, extra sources of information are needed to predict their preferences. Therefore, social relations with confidence values provide a perfect choice for recommendation systems to serve this kind of users.

4.3. **The impact of implicit social relations.** Figure 2 shows the impact of utilizing implicit relationships. This figure displays 10% of the relations using the Gephi software. As shown in this figure, the number of users increases when including implicit relations. For example, Figure 2(a) indicates the explicit relations only for the Ciao dataset, which has 111,780 relations. Figure 2(b), on the other hand, depicts the explicit-implicit relations for the same dataset, which has 54,056,070 relations. As seen in both figures, there are significantly more relations among users in Figure 2(b). This fact demonstrates that the information that can be extracted using explicit-implicit relations is more than the information extracted via explicit relations only. This information can include more users, especially cold-start ones since the opportunity to serve them by exploiting explicit-implicit relations is higher than when using explicit relations only. Furthermore, if the clusters are established using explicit relations only, most users will be considered outliers

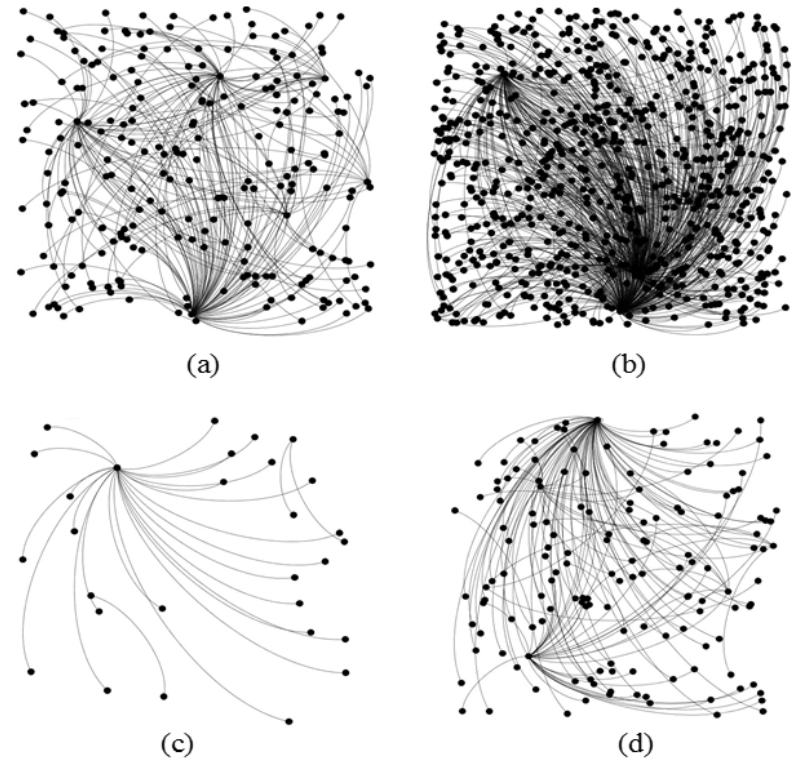

FIGURE 2. The impact of adding implicit relations. Figures (a) and (c) are explicit relations only for Ciao and FilmTrust respectively. Figures (b) and (d) are explicit-implicit relations for the same dataset.

and are not assigned to any cluster. Therefore, many cold-start users are not serviced. Consequently, implicit relations provide an excellent option to enrich the relations and can contribute to clustering the users more efficiently. Similarly, Figures 2(c) and 2(d) show the explicit and explicit-implicit relations respectively for the FilmTrust dataset. In these figures, the relations are also increased by adding implicit relations. Whereas the number of explicit relations is 1,853, the number of explicit-implicit relations is 73,057.

4.4. **The impact of using confidence values.** Tables 7 and 8 display the results of MAE and RMSE for different cluster numbers. Two views are displayed in these tables with and without confidence values. In other words, the first view reports the results by combining social relations with confidence values. The second view depicts the results via social relations only. As seen in both tables, the clustering with confidence values outperformed that without confidence values. Two criteria are utilized to check the impact of using confidence values. They are the accuracy metrics (MAE and RMSE) and the number of users in each cluster. Regarding the number of users, it is obvious that the clustering with confidence values includes more users than that without confidence values. For example, in Table 8, there is a big gap between clustering with confidence and without confidence. The maximum number of users in the clustering without confidence values is 62 when the number of the clusters is 15. On the other hand, in the clustering with confidence values, the number of users is 330 with the same number of clusters. Thus, 268 users remain out of the clustering process as outliers. Table 8 displays the results of the FilmTrust dataset. In this dataset, there are few social relations among users, namely: 1,853 explicit relations and 73,057 implicit ones. So, in the clustering process, relying

Table 7. The clustering results with and without confidence values, for Ciao dataset

	With confidence values			**Without confidence values**		
No. of clusters	**MAE**	**RMSE**	**No. cold-start users**	**MAE**	**RMSE**	**No. cold-start users**
5	0.7268	0.9333	724	0.7197	0.9345	626
7	0.7200	0.9259	740	0.7255	0.9407	590
9	0.7186	0.9346	742	0.7559	0.9589	739
11	0.7213	0.9312	744	0.7194	0.9561	741
13	0.7278	0.9258	745	0.7238	0.957	741
15	0.7126	0.9221	749	0.7202	0.9417	732
17	0.7475	0.9614	746	0.7142	0.9324	651

Table 8. The clustering results with and without confidence values, for FilmTrust dataset

	With confidence values			**Without confidence values**		
No. of clusters	**MAE**	**RMSE**	**No. cold-start users**	**MAE**	**RMSE**	**No. cold-start users**
5	0.6273	0.7797	**285**	0.6736	0.8634	**53**
7	0.5960	0.7706	**302**	0.6364	0.7870	**55**
9	0.6055	0.7915	**321**	0.6641	0.8819	**59**
11	0.6151	0.8107	**327**	0.5986	0.7798	**58**
13	0.6188	0.8126	**328**	0.7175	0.9136	**56**
15	0.6358	0.8206	**330**	0.6941	0.8905	**62**

on social relations (without confidence values) leaves most users out of the clustering as outliers. In contrast, Table 7 shows the results of the Ciao dataset. A small difference in the number of users can be seen between clustering with confidence and clustering without confidence values. As the Ciao dataset has a huge number of social relations among users, namely: 11,780 for explicit relations and 54,056,070 for implicit once; thus, confidence adds a small number of relations. Consequently, the difference between confidence and without confidence is small. For example, the difference is 17 users when the number of clusters is 15. Regarding the second criterion, the accuracy of the results is affected positively by adding more users. Thus, the results with confidence outperformed the results that do not use confidence values.

In summary, the relations among users can be represented using social relations or confidence values. Integrating both types leads to increasing relations between users. Hence more users can be involved in the clustering process. Thus, the clustering is implemented more accurately and more users will be served.

5. **Conclusions.** The cold-start problem is very common in recommendation systems. It is considered a special case of the sparsity problem, where cold-start users are those who have few ratings (less than a threshold). In this paper, a new hybrid method was proposed to alleviate this problem. The proposed method exploited social trust relations by computing the similarity between each pair of explicit and implicit trust relations users. It also employed the users' ratings by calculating the confidence values between each pair of users. Then, the users are clustered into groups according to their distances from the medoid of the cluster. Moreover, the SVD++ method is applied for each cluster to predict items for cold-start users. In addition, the results of this study revealed that grouping users with similar attributes into one cluster rather than distributing them over the whole dataset is a suitable option to reduce the cold-start user problem and increase the prediction accuracy. Furthermore, the results showed that trust relations provide vital information that can be utilized to cluster users on the basis of their social relations, especially when adding implicit relations as an extra source. However, some users have few social relations. Thus, if the clustering is done using social relations only, it leads to disregarding many users, who will remain out of clusters. Therefore, confidence values provide an additional source of information to boost the clustering process. They allow more users to be included in the clusters, which will subsequently reduce the number of outliers, thereby helping the cold-start users. The experimental results demonstrated that the proposed method enhanced the prediction accuracy and surpassed the state-of-the-art methods in all cases. In a further study, a different similarity measure can be utilized to promote the clustering accuracy. A further issue is that, during the clustering process, determining the number of cold-start users in each cluster can enhance the results, in cases where the number of cold-start users is less than that of non-cold-start users. Eventually, the proposed model can be developed by applying it to the cold-start item problem.

REFERENCES

[1] L. H. Son, Dealing with the new user cold-start problem in recommender systems: A comparative review, *Information Systems*, vol.58, pp.87-104, 2016.

[2] A. M. A. Al-Sabaawi, H. Karacan and Y. E. Yenice, Exploiting implicit social relationships via dimension reduction to improve recommendation system performance, *PLOS ONE*, vol.15, no.4, DOI: 10.1371/journal.pone.0231457, 2020.

[3] C.-F. Tsai and C. Hung, Cluster ensembles in collaborative filtering recommendation, *Applied Soft Computing*, vol.12, no.4, pp.1417-1425, 2012.

[4] H. Ma, I. King and M. R. Lyu, Learning to recommend with social trust ensemble, *Proc. of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.203-210, 2009.
[5] J. Yu, M. Gao, W. Rong, Y. Song and Q. Xiong, A social recommender based on factorization and distance metric learning, *IEEE Access*, vol.5, pp.21557-21566, 2017.
[6] B. Sarwar, J. Konstan, A. Borchers and J. T. Riedl, *Applying Knowledge from KDD to Recommender Systems*, Technical Report, University of Minnesota Digital Conservancy, http://hdl.handle.net/11299/215369, 1999.
[7] A. Mnih and R. R. Salakhutdinov, Probabilistic matrix factorization, *Advances in Neural Information Processing Systems*, pp.1257-1264, 2008.
[8] Y. Koren, P. Ave, F. Park, H. D. Management and D. Applications, Factorization meets the neighborhood: A multifaceted collaborative filtering model, *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.426-434, DOI: 10.1145/1401890.1401944, 2008.
[9] M. Wasid and R. Ali, An improved recommender system based on multi-criteria clustering approach, *Procedia Computer Science*, vol.131, pp.93-101, 2018.
[10] M. Medhat, Y. F. Hassan and A. Elsayed, Humans and bots web session identification using K-means clustering, *ICIC Express Letters*, vol.13, no.12, pp.1149-1156, 2019.
[11] M. Hussein, H. Naji and W. Bhaya, Influential nodes based alleviation of user cold-start problem in recommendation system, *Research Journal of Applied Sciences*, vol.11, pp.1107-1114, DOI: 10.3923/rjasci.2016.1107.1114, 2016.
[12] H. Koohi and K. Kiani, User based collaborative filtering using fuzzy C-means, *Measurement*, vol.91, pp.134-139, 2016.
[13] Z. Nazari, M. Nazari and D. Kang, A bottom-up hierarchical clustering algorithm with intersection points, *International Journal of Innovative Computing, Information and Control*, vol.15, no.1, pp.291-304, 2019.
[14] B. Yang, Y. Lei, J. Liu and W. Li, Social collaborative filtering by trust, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.39, no.8, pp.1633-1647, 2016.
[15] G. Guo, J. Zhang and N. Yorke-Smith, Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, *The 29th AAAI Conference on Artificial Intelligence*, 2015.
[16] H. Parvin, P. Moradi, S. Esmaeili and N. N. Qader, A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method, *Knowledge-Based Systems*, vol.166, pp.92-107, 2019.
[17] J. Jiao, X. Zhang, F. Li and Y. Wang, A novel learning rate function and its application on the SVD++ recommendation algorithm, *IEEE Access*, vol.8, pp.14112-14122, 2019.
[18] W. Shi, L. Wang and J. Qin, User embedding for rating prediction in SVD++-based collaborative filtering, *Symmetry*, vol.12, no.1, p.121, 2020.
[19] M. H. Dunham, *Data Mining: Introductory and Advanced Topics*, Pearson Education, India, 2006.
[20] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang and X. Kong, 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.
[21] J. Tang, H. Gao, H. Liu and A. Das Sarma, eTrust: Understanding trust evolution in an online world, *Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.253-261, 2012.
[22] G. Guo, J. Zhang and N. Yorke-Smith, A novel bayesian similarity measure for recommender systems, *The 23rd International Joint Conference on Artificial Intelligence*, 2013.
[23] M. Jamali and M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, *Proc. of the 4th ACM Conference on Recommender Systems*, pp.135-142, 2010.
[24] H. Ma, D. Zhou, C. Liu, M. R. Lyu and I. King, Recommender systems with social regularization, *Proc. of the 4th ACM International Conference on Web Search and Data Mining*, pp.287-296, 2011.
[25] S. Ahmadian, M. Afsharchi and M. Meghdadi, An effective social recommendation method based on user reputation model and rating profile enhancement, *Journal of Information Science*, vol.45, no.5, pp.607-642, 2019.