Journal of Universal Computer Science, vol. 27, no. 7 (2021), 667-692
submitted: 20/5/2020, accepted: 26/5/2021, appeared: 28/7/2021

BSO-MV: An Optimized Multiview Clustering Approach for Items Recommendation in Social Networks

Lamia Berkani
(LRIA laboratory, Dep. of Computer Sc., USTHB University, Algiers, Algeria
https://orcid.org/0000-0003-3601-2698, lberkani@usthb.dz)

Lylia Betit
(Dep. of Computer Sc., USTHB University, Algiers, Algeria
lbetit@usthb.dz)

Louiza Belarif
(Dep. of Computer Sc., USTHB University, Algiers, Algeria
lbelarif@usthb.dz)

Abstract: Clustering-based approaches have been demonstrated to be efficient and scalable to large-scale data sets. However, clustering-based recommender systems suffer from relatively low accuracy and coverage. To address these issues, we propose in this article an optimized multiview clustering approach for the recommendation of items in social networks. First, the selection of the initial medoids is optimized using the Bees Swarm optimization algorithm (BSO) in order to generate better partitions (i.e. refining the quality of medoids according to the objective function). Then, the multiview clustering (MV) is applied, where users are iteratively clustered from the views of both rating patterns and social information (i.e. friendships and trust). Finally, a framework is proposed for testing the different alternatives, namely: (1) the standard recommendation algorithms; (2) the clustering-based and the optimized clustering-based recommendation algorithms using BSO; and (3) the MV and the optimized MV (BSO-MV) algorithms. Experimental results conducted on two real-world datasets demonstrate the effectiveness of the proposed BSO-MV algorithm in terms of improving accuracy, as it outperforms the existing related approaches and baselines.

Keywords: Social recommendation, Collaborative filtering, Hybrid filtering, Clustering, Multiview clustering, Optimized-based clustering, Bees Swarm optimization algorithm, Optimized multiview clustering
Categories: G.1.6, I.5.3, L.2.2
DOI: 10.3897/jucs.70341

1 Introduction

The world is witnessing a rapid revolution of information and communication technology. With the development and expansion of the Internet and social media technologies, online learning communities and scientific social networks (SSN) have become the most efficient way for exchange and knowledge sharing between people. Social media statistics show that huge numbers of data are shared every day, which make it difficult for users to find highly valuable items and relevant information. However, the rapid increases in the rate at which new items are published and the ease

of sharing them in these SSN platforms has led to information overload problem [Wang, 2018]. In this regard, building recommendation systems has become an effective and important way to deal with these issues.
Among different recommendation algorithms, Collaborative Filtering (CF) is the most widely-exploited technique in recommender systems to provide users with items that well suit their preferences [Adomavicius, 05]. The basic idea is that a prediction for a given item can be generated by aggregating the ratings of users with similar interests. However, despite the growing popularity of the CF, the time-consuming process of searching for similar users is considered as a big challenge when facing large-scale data sets, which characterizes the Web 2.0 and the social media contexts in particular. Moreover, one of the shortcomings of memory-based methods is the sparsity and cold start problems due to insufficient ratings [Pazzani, 07]. In contrast, model-based methods can address these issues by training a prediction model offline using all the rating data.
To cope with the sparsity and cold start problems, which may affect the prediction accuracy and recommendation quality, many approaches have been proposed in the literature. Most of these approaches are based on the idea of integrating additional information to overcome the lack of rating data. Trust-aware CF approaches [Guo, 14; Park, 16] and social-based CF approaches [Guo, 18; Lai, 19] have attracted much attention in recent years.

On the other hand, recommender systems using clustering-based approaches offer an alternative to model-based methods [Sarwar, 02]. Instead of decomposing the rating matrix into matrices with small ranks, these approaches reduce the search space by clustering similar users or items together. Most previous works focus on clustering users and / or items from the view of similarity [Guo, 15]. However, the state of the art shows that these approaches suffer from relatively low accuracy and coverage. To address these issues, [Guo, 15] developed a multiview clustering method through which users are iteratively clustered from the views of both rating patterns (user similarity) and social trust relationships (social similarity). In the same way, a multi-view clustering approach for the recommendation of items in social networks is proposed [Berkani, 20a]. This work considered the similarity and social information with different features: friendship, trust and influence. Besides, in previous work, we have developed an optimized-clustering based collaborative and social filtering algorithm using the Bees Swarm optimization (BSO) and Kmedoids algorithms [Berkani, 19]. The objective was to refine the quality of medoids according to the objective function and therefore recommend the most appropriate items to a given user. In order to take advantage of this approach, an extended and improved version is proposed in this article with further enrichments.

The main contributions of this article can be summarized as follows:

1. We develop various clustering-based CF algorithms using different clustering algorithms, such as Kmedoids and CLARANS and adapt the multiview (MV) method using PAM to the Kmedoids and CLARANS algorithms.
2. We propose a novel optimized MV clustering approach using BSO optimization algorithm, considering the improvement in recommendation accuracy achieved with the optimized classification [Berkani, 19], on one side, and the multiview clustering method in social context [Berkani, 20a], on the other side.

3. We conduct extensive experiments using different datasets in order to confirm the effectiveness of the proposed MV clustering method in comparison with other methods and demonstrate the added value of the optimization on both, clustering-based and MV method.

In our work, the selection of the initial medoids is optimized using BSO, in order to generate better partitions (i.e. refining the quality of medoids according to the objective function) and therefore increase the accuracy of the recommendations (i.e. recommending the most appropriate items to a given user). Then, users are iteratively clustered from the views of both user similarity (using users' assessments on items) and social information (friendships and trust). Finally for comparison purposes, prediction results are generated according to different hybridization.

The remainder of this paper is organized as follows: Section 2 presents some related work on social, trust-based and clustering-based recommender systems. Section 3 proposes a novel social recommender system of items using an optimized multiview clustering approach. Then in Section 4, experiments are conducted using two real-world datasets. Finally, Section 5 highlights the most important results and outlines some future research.

2 Related Work

In this section, we review the existing related work through two main research threads: (1) social and trust-based recommendation; and (2) clustering-based recommendation approaches.

2.1 Social and Trust-based Recommendation

According to [Ma, 11] social recommender systems use the social friends' network to enhance recommender systems, while trust-based recommender systems utilize the users' trust relations and consider that users have similar tastes to those of other users they trust.

Trust-based recommendations can improve the performance of traditional recommender systems as people trusting each other usually share similar preferences [Singla, 08]. Trust information has been widely used as an additional dimension to support model user preference in recommender systems. It has been combined in both types of recommendation: (1) in memory-based methods [Massa, 07; Guo, 14]; and (2) in model-based methods [Ma, 09; Jamali, 10]. Due to data sparsity of the input ratings matrix, the authors in [Massa, 04] replaced the step of finding similar users with the use of a trust metric. They proposed an algorithm that propagate trust over the trust network and estimate a trust weight that can replace the similarity weight. Massa and Avesani [Massa, 07] implemented a trust metric to detect trusted friends through trust propagation over the so called Web of trust. Other authors proposed a trust-based approach where the relationships between users are calculated by propagating trust and using traditional CF [Nazemian, 12]. In [Guo, 14], the authors incorporated trusted neighbours into the CF, by merging the ratings of trusted neighbours. The objective is to form a more complete rating profile for active users to solve the cold start and sparsity problems.

On the other hand, the use of social network information has been widely used. For instance, [Ma, 11] developed two methods: a matrix factorization framework with social regularization and a factor analysis approach based on probabilistic matrix factorization that exploits users' social network. Based on these two methods, Wang and Huang [Wang, 14] included the friendships as the regularization term to enhance the prediction accuracy. Recently, a social recommendation system based on users' attention and preferences was developed [Chen, 19].

2.2 Clustering-based Recommendation

Clustering-based approaches are being demonstrated to be efficient and scalable to large-scale data sets. As a dimension-reduction method, they are capable of alleviating the sparsity of rating data [Pham, 11]. Recent works reported that by applying more advanced clustering method, the accuracy can be further improved and even outperform the other CF approaches [Bellogín, 12]. An incremental CF system based on a weighted clustering approach is proposed in [Salah, 16]. This approach aims to provide high quality recommendations with a very low computation cost.

Few works have tried to integrate social relationships into clustering-based methods with the aim of improving the performance of CF. For instance, in [Sun, 15], the authors proposed a social regularization approach that incorporates social network information, namely the users' friendships and rating records (tags) for the prediction of the recommendations. They used a bi-clustering algorithm to identify the most suitable group of friends for generating different final recommendations. In [DuBois, 09], the authors combined a correlation clustering algorithm and trust models together to derive trust from the connection distance in a trust network. However, only limited improvement is observed.

According to [Guo, 15] previous clustering-based approaches suffer from relatively low accuracy and, especially, coverage. To alleviate these issues, they developed a multiview clustering method through which users are iteratively clustered on the basis of rating patterns, in one view, and social trust relationships, in the other. Sheugh and Alizadeh [Sheugh, 15] proposed a multiview clustering based on Euclidean distance, merging similarity and trust relationships including explicit and implicit trusts. A web items recommendation system based on a multi-content clustering CF model was proposed in [He, 14]. Different views such as user ratings and user comments have been considered and users' preferences were analysed by their historical interaction features and additional behaviour features for an appropriate recommendation. Recently, a multiview clustering recommendation algorithm with additional social information (friendship, trust and influence) is developed [Berkani, 20a]. This work demonstrated the importance of combining these features and its positive impact on hybrid recommendation.

On the other hand, to our best knowledge, few works used optimization techniques in clustering-based recommender systems. A model-based CF based on a fuzzy c-means clustering approach is considered in [Selvi, 17]. Because model-based CF suffers by higher error rate and takes more iteration for convergence, the authors proposed a modified cuckoo search algorithm to optimize the data points in each cluster in order to provide an effective recommendation. An optimized-clustering based collaborative and social filtering algorithm using BSO and Kmedoids algorithms, is proposed in [Berkani, 19].

2.3 Discussion and Research Issues

Our study of related work confirmed the contribution of integrating social information with CF to improve the recommendation accuracy. On the other hand, the multiview clustering has been adopted in some works, as clustering-based approaches suffer from relatively low accuracy and, especially, coverage [Guo, 15]. Finally, to our best knowledge optimized multiview-based clustering methods have not been exploited in recommender systems. This motivates us to develop an optimized method that is capable of alleviating these issues and generate better partitions.

3 An Optimized Multiview Clustering-based Recommendation

3.1 Overall Description of the Recommendation Framework

Figure1 gives an overview of our recommendation approach. Several algorithms have been implemented for CF, social filtering (SocF) and hybrid filtering (HybF). The UCF, KCF, BSO-KCF abbreviations correspond respectively to the standard user-based CF, the clustering-based CF and the optimized clustering-based CF using the Bees Swarm optimization algorithm (BSO). Similarly, SocF, KSocF, BSO-KSocF algorithms correspond to the standard SocF, the clustering-based SocF and the optimized clustering-based SocF using BSO.

Moreover, different hybridization have been considered: (1) the weighted hybrid algorithm with different variants of combination: (i) the standard CF and SocF algorithms (W-HybF); (ii) the clustering-based CF and SocF algorithms (KHybF); and (iii) the optimized clustering-based CF and SocF algorithms using BSO (BSO-KHybF); (2) the multiview clustering based recommendation (MV); and (3) the optimized MV using BSO (BSO-MV) recommendation algorithms.

3.2 User's Profile Modelling

We consider that each user is characterized by a set of information including the assessments made on resources, the list of friends and trust information. The similarity between two users is based on collaborative and social distance calculation:

- **Collaborative distance:** The similarity calculation between two users, *u* and *v*, is based on the evaluation history, and uses the Pearson correlation function [Adomavicius, 05]. The distance between two users, *u* and *v*, denoted $D_{Sim}(u, v)$ is calculated as follows:

$$D_{Sim}(u, v) = 1 - Sim_{Pearson}(u, v) \tag{1}$$

- **Social distance:** In order to determine the social relationship between users, two features have been used, namely: friendship and trust [Berkani, 20a]:

1) *Extraction of the degree of friendship:* The extraction of the degree of friendship between two users is calculated with the Jaccard formula:

$$Friendship\ (u, v) = \frac{|F_u \cap F_v|}{|F_u \cup F_v|} \tag{2}$$

where : F_u is the set of friends of u and F_v is the set of friends of v.

The distance $D_{Friendship}(u,v)$ is calculated as follows:

$$D_{Friendship}(u,v) = 1 - Friendship(u,v) \quad (3)$$

2) *Extraction of the degree of* $D_{Trust}(u,v)$ *:* There are several algorithms for the calculation of trust. We have chosen the six-level method [Guo, 15], which calculates how much two users u and v trust each other, considering a distance equal to six. The distance $D_{Trust}(u,v)$ is calculated as follows:

$$D_{Trust}(u,v) = 1 - Trust(u,v) \quad (4)$$

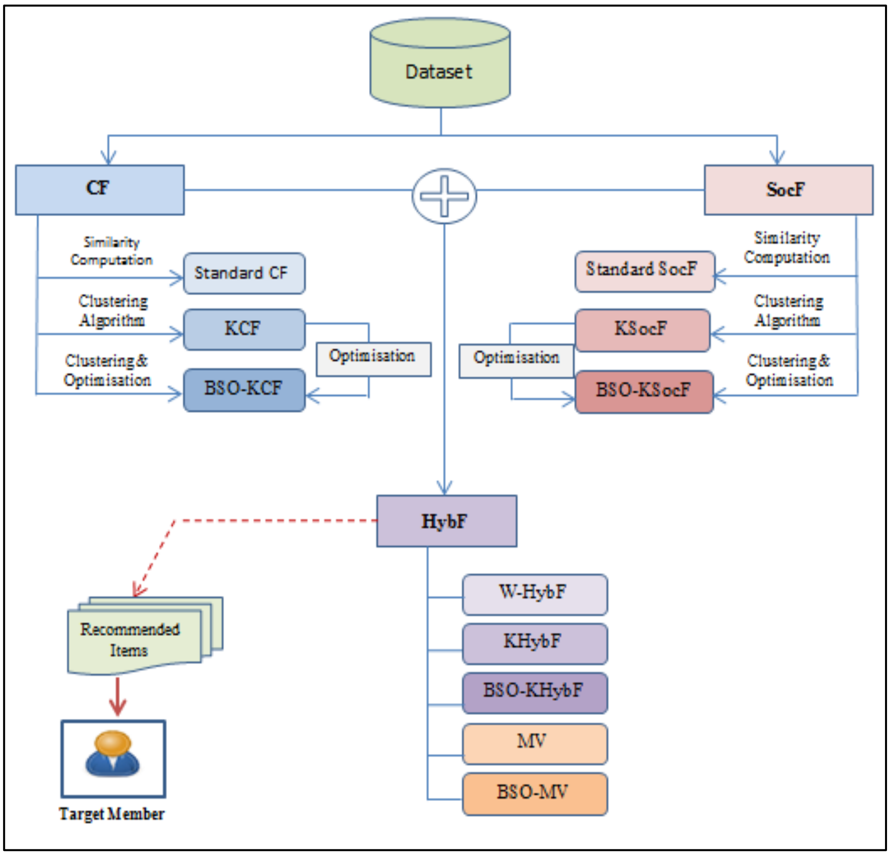


Figure 1: Overview of the BSO-MV clustering recommendation framework

3.3 Collaborative and Social Recommendation Algorithms

For the standard CF, we chose a memory-based CF approach and used the user-user based recommendation. In this approach, the system offers the possibility of identifying the best neighbours for a given user, using the ratings of users on items. As stated above, we adopted the Pearson correlation function to compute the similarity between users, i.e. $Sim_{Pearson}(u, v)$.

On the other hand, the standard SocF, considers trust and friendship features to calculate the social distance. A weighted formula was used to calculate the social distance:

$$D_{Soc} = \beta_1 * D_{Trust} + \beta_2 * D_{Friendship} \tag{5}$$

where: β_1, β_2 : represent the importance weights related, respectively, to trust and friendship, with: $\beta_1 + \beta_2 = 1$.

The pseudo algorithm below illustrates the standard SocF steps:

Algorithm 1 – SocF ( )
Input: Active user U_a, The distances: $D_{Friendship}, D_{Trust},\ Item_j: Item$,
k: number of nearest neighbours to consider for prediction.
Output: Prediction *(U_a , $Item_j$).*

BEGIN
1: Construct the D_{Soc} table using the weighted D_{Soc} formula (Formula 5);
2: Generate the clustering configuration based on D_{Soc};
3: Select the K closest neighbours in terms of D_{Soc};
4: Apply the prediction *(U_a,$Item_j$)* based on K nearest neighbours' assessments;
END.

3.4 Clustering-based Collaborative and Social Recommendation Algorithms

The clustering-based CF algorithm (**KCF)** detects the community of the active user U_a using a clustering algorithm. The prediction will be based on users' evaluations that belong to the same cluster. The Kmedoids algorithm has been chosen as this algorithm is more robust to noise and outliers as compared to K-means because it minimizes a sum of pairwise dissimilarities instead of generating a centroid by calculating the average of all the values of each attribute of the cluster.

We have implemented different variants of the K-medoids algorithm: (1) the Partitioning Around Medoïd (PAM) algorithm [Kaufman, 87], which is the most common realization of k-medoid clustering; and (2) the clustering large applications based upon randomized search (CLARANS) algorithm [Ng, 02]. The PAM-based CF allows predictions based on clusters that are generated by the application of the PAM algorithm. PAM uses a greedy search which may not find the optimum solution, but it is faster than exhaustive search. The following is the PAM algorithm:

Algorithm 2: PAM Algorithm ( )
Input: usage matrix, number of K clusters Output: sets of K cluster medoids.
Begin
  Initialize K cluster medoids: Randomly designate K users;
  Calculate the initial cost;
  For each medoid *M* Do
    For each non medoid *O* Do
      Swap (M, O);
      Distribution ();
      Calculate-Cost();
      If cost current iteration > cost previous iteration Then Undo (M, O)
      End If
      If no change Then exit algorithm;
      End If
    Done
  Done
End.
where:
  Swap (M, O): replaces the medoid *M* by the user *O*;
  Undo (M, O): cancel the swap of *M* by *O*;
  Distribution: Find the closest cluster for each user in the matrix by calculating the Pearson correlation between the user and the K Medoids; and
  Cost: is an objective function that is intended to be minimized, as follows:

$$Cost = min \sum_{c \in C} \sum_{u,v \in C} d(u, v) \quad (6)$$

where: C is the set of clusters (partitions) resulting from the K-medoïds algorithm; *u*, *v* are two users, where *v* belongs to the cluster and *u* is the medoid of the latter.

The following is the PAM-based CF algorithm:

Algorithm 3: PAM-based CF Algorithm ( )
Input D_{Sim}, Active user$U_a, Item_j$, *K*: number of medoids.
Output: Prediction for the active user U_a on $Item_j$.
Begin
  1- Random selection of initial K medoids set (Init_Medoids);
  2- Application of PAM with Init_Medoids using D_{Sim} distance;
  3- Identification of the cluster of U_a.
  4- Prediction ***(**U_a, $Item_j$**).***
End.

The clustering-based SocF algorithm (KSocF) is similar to the KCF algorithm, replacing the collaborative distance $\boldsymbol{D_{Sim}}$ by the social distance $\boldsymbol{D_{Soc}}$.

3.5 BSO Clustering-based CF and SocF algorithms

3.5.1 BSO clustering-based CF algorithm

Kmedoids clustering technique is based on a random selection of medoids, thus we are going to apply the BSO meta-heuristic for an optimized selection of initial medoids. We describe below the process of the BSO clustering-based CF algorithm:

1. Calculate the distances of U_a with the other users, based on the user-item rating matrix, then store the result in the D_{Sim} table.
2. Choose one of the partitioning clustering techniques (Kmedoids, PAM or CLARANS).
3. Apply the BSO for the medoids selection, which will return the best found reference solution.
4. Calculate the distances between the users and the medoids stored in the solution table.
5. Generate the final clustering configuration.
6. Identify the community of the active user U_a.
7. Apply the prediction formula.

3.5.2 BSO adaptation for clustering techniques

The BSO process adapted for our medoids selection problem is described as follows:

1- Solution space: is the set of identifiers of existing users in the database, where each solution is represented by a vector of size *k*, where *k* is the number of partitions (number of medoids).

2- Evaluation of a solution: the fitness of a solution is the sum of the distances between each non-medoid user *U* and the medoid *m* of the least distant solution to *U*, according to the following formula:

$$Fitness\ (S) = \sum_{\substack{U \in \omega \\ \omega \cap S = \emptyset}} \sum_{m \in S} \min\left(d^2(U, m)\right) \tag{7}$$

where:
U: is a non medoid user;
m: a medoid of a solution S; and
ω: is a set of all non-medoids users.

3- Bees exploration space: the search process is initiated by a reference solution called "*Sref*" which initially contains the vector of the initial medoids representing the scout bee. Once *Sref* is built, it will determine the *SearchAr*ea (i.e. the area to be exploited). This area consists of all the possible solutions. To build the bee exploration space, we used a method that generates new solutions dispersed in the search space from the *Sref* solution using the empirical parameter "*flip*", which will model a distance between the solutions.

4- Neighbourhood of a solution: after the generation of solutions, each artificial bee begins to explore the region which has been assigned to it to determine the best local neighbour solution. In order for the artificial bees to be able to carry out the local search, it would first be necessary to seek the neighbourhood of each solution.

For our case, we exploited non-medoid users who belong to clusters whose medoids of the solution in question are representatives of the latter (by ordering them from the closest to the most distant relative to the medoid of their cluster).
In order to avoid the worst case of complexity by exploring all the points of the clusters, we proposed the following formula which allows us to explore a large number of points which depends on the size of the cluster in question. The number of neighbours of a solution is given by the following formula:

$$Nb\_Neighbours = \sum_{m \in S} \frac{1}{Threshold} * |C_m| \quad (8)$$

where:
Threshold: represents the proportion of the size of the cluster to be explored;
m: is the medoid; C_m: is the cluster having *m* as representative medoid; and
S: represents a solution.

Once the neighbourhood of the solution has been identified and the local search has been carried out, each bee will store the solution in the Dance table, which is visible to all the other bees. Once all the solutions have been placed in the Dance table, the solution that minimizes the objective function is selected as the best solution, and will then become a reference solution for the next iteration. The reference solutions are inserted in the taboo list in order to avoid exploring regions already visited.
To avoid the stagnation problem, we adopted the principle of dispersion, by choosing the most distant solution from the explored one. Let's consider "S_1" be the stagnant solution, "S" the most distant solution from S_1 which is chosen from the *Dance* table and is selected using the following formula:

$$S = Max^{Dance\_Size}_{\substack{i=1, \\ r_t \in S_i,\ S_i \in Dance}} \sum^{K}_{\substack{j=1 \\ m_j \in S1}} \mathrm{Min}^{K}_{t=0}\, d(m_j, r_t) \quad (9)$$

where:
Dance_Size: is the size of the *Dance* table containing the solutions;
r_t: is the medoid of the S_i solution which belongs to the *Dance* table; and
m_j: is the medoid of the S_1 we want to get away from.

Once the max iteration is reached, we will keep the best configuration, which minimizes the objective function.

3.6 Hybrid Recommendation Algorithm

3.6.1 The weighted hybrid algorithm (W-HybF)

The W-HybF combines the interests based similarity of users with their social similarity weight to compute the overall similarity between two users as follows:

$$Sim_{Hyb}(u_1, u_2) = \alpha * Sim_{Pearson}(u_1, u_2) + \beta * Sim_{Soc}(u_1, u_2) \quad (10)$$

where: α and β are weights that express a priority level, with $\alpha + \beta = 1$.

The Hybrid Distance D_{Hyb} is calculated as follows:

$$D_{Hyb} = \alpha * D_{Sim} + (1-\alpha) * D_{Soc} \tag{11}$$

This algorithm starts by calculating the distances between the active user U_a and all the other users by applying the previous formula (Formula 11), in order to identify the k nearest neighbours to U_a. Then the prediction function will be applied.

3.6.2 The clustering-based hybrid algorithm (KHybF)

The KHybF algorithm combines the KCF and the KSocF algorithms in the same way as with the W-HybF. Two variants of this algorithm can be considered, combining the CF with trust and friendship social information (KCFSoc) or with trust information (KCFTrust).This algorithm proceeds according to the following steps:

- Calculate the distances between U_a and the other users using D_{Hyb};
- Generate a final configuration of clustering;
- Identify the cluster associated with the active user;
- Calculate the predictions using the harmonic average weighted prediction formula considering the users of the cluster which contains U_a.

3.6.3 The optimized clustering-based hybrid algorithm (BSO-KHybF)

The weighted optimised clustering-based hybrid algorithm combines the BSO-KCF and the BSO-KSocF algorithms in the same way as with the KHybF. The only difference is that in this case the clusters generated have been optimized with the BSO metaheuristic.

3.7 BSO Multiview Clustering-based Recommendation

The multiview clustering algorithm presented by [Guo, 15] combines the trust and similarity information, where two classification processes run in an alternative way and end with an integration phase to generate a final multiview classification. We have adapted this algorithm by considering similarity and social information (friendship and trust) and applying the BSO algorithm for an optimized selection of the initial medoids (BSO-Initial-Medoids-Optimization). The optimized clustering-based MV is executed according to the following steps (see Algorithm 4):

- *Step 1:* This step allows an optimized selection of the initial medoids. First a randomly selection of social medoids is done. Then, this selection is optimized using the BSO algorithm. Finally, a generation of social clustering C_{Soc} configuration is done based on D_{Soc}.

- *Step 2:* In this step, the classification will be done using a clustering algorithm (Kmedoids, PAM or CLARANS) once according to the social view, and another time according to the similarity view, while passing the resulting medoids from each classification step to the next classification that will follow. Thus two clustering configurations of different views C_{soc} and C_{sim} will be built.

- *Step 3:* Apply an integration algorithm [Guo, 15]. The social and similarity clusters are taken as input, and user clusters are obtained as output. The integration will be triggered by a criterion, i.e., the number of cluster members being less than a cluster threshold θ_c. For each cluster C^i_{Soc} in the clusters C_{Soc}, if the criterion is satisfied, the integration will proceed. If we find another cluster C^j_t that achieves the minimum average distance between each member *u* in C^i_{Soc} and the medoid centroid m^i_{Soc} of cluster C^j_{Soc}, all the members of cluster C^i_{Soc} will be merged into cluster C^j_{Soc}. The cluster C^i_{Soc} will be pruned regardless of whether it will be merged or not. After processing social clusters C_{Soc}, we repeat the procedure by replacing C_{Soc} with similarity clusters C_{Sim}. Finally, the clusters are combined in a pairwise manner and returned as output. The pairwise combination is due to the iterative procedure, where the cluster medoid is derived from the previous clusters from the other view.
- *Step 4*: With the adopted hybridization technique, a user can belong to at most two clusters, due to the integration step (step 3). Once the final configuration is obtained, we will only have to identify the community of the user U_a to be able to calculate its prediction on the item *j* to recommend. The latter can belong either to a single cluster *"C"* and thus the generation of prediction $P^C_{U_a,j}$ will be done with the formula of the harmonic mean, or it will belong to the intersection of two clusters, in this case the prediction generation will proceed in two ways [Berkani, 20a]:
  1. *The Harmonic average method:* let us start with the most trivial method that will be noted AVG, which consists in calculating the average of the two predictions: $P^{C_1}_{U_a,j}$ and $P^{C_2}_{U_a,j}$ applying the formula of the harmonic mean and considering the two clusters C_1 and C_2 separately where U_a appears, as follows:

$$AVG = \frac{1}{2}\left(P^{C_1}_{U_a,j} + P^{C_2}_{U_a,j} \right) \quad (12)$$

  2. *The SVR regression-based method:* in order to improve our results, we applied a second method for the generation of predictions for the active user, which belongs to the intersection of two clusters. This method involves using supervised classification for the prediction, using the SVR regression technique.

The algorithm 4 adapted from [Guo, 15] is described as follows:

Algorithm 4: BSO-MV ( )

Input: Distance matrix D_{Sim}, $D_{Friendship}$, D_{Trust}, cluster number k, β_1 *and* β_2 (importance weights of trust and friendship respectively used in Formula 5 to calculate the social distance D_{soc});
Output: user clusters C
P ← 0;
Randomly select k medoids m_{Soc} from social users $\theta^0_{soc} \leftarrow m_{Soc}$
$\theta^0_{soc} \leftarrow$ BSO-Initial-Medoids-Optimization (θ^0_{soc});
Calculate D_{soc} (v, m_{soc}) using Formula 5;
C^0_{Soc} ← v, given min (D_{soc} (v, m_{soc}));

While medoids changed **and <** max iterations **do**
p ← p +1 ;
θ_{sim}^{p} ← θ_{soc}^{p-1} ;
Swap(m_{sim} , u), u ∈ C_{soc}^{p-1} ; //Clustering Algorithm : K-medoids / CLARANS
Calculate sum_{sim} (u) = $\sum_v D_{sim}(u, v), v \in C_{soc}^{p-1}$;
If sum_{sim} (u) < sum_{sim} (m_{soc}) **then**
m_{sim} ← u ;
θ_{sim}^{p} ← m_{sim} ;
C_{sim}^{p} ← v, for ∀v, find m_{sim} s.t. min (D_{sim} (v, m_{sim})) ;
p ← p+1 ;
θ_{soc}^{p} ← θ_{sim}^{p-1} ;
swap(m_{soc} , u), u ∈ C_{soc}^{p-1} ; //Clustering Algorithm : K-medoids / CLARANS
Calculate sum_{soc} (u) = $\sum_v D_{soc}(u, v), v \in C_{sim}^{p-1}$;
If sum_{soc} (u) < sum_{soc} (m_{sim}) **then**
m_{Soc} ← u ;
θ_{soc}^{p} ← m_{Soc} ;
C_{soc}^{p} ← v, for ∀v, find m_{Soc} s.t. min(D_{Soc} (v, m_{Soc})) ;
Return C ← Integrate ($C_{soc}^{p}, C_{sim}^{p-1}$) ;

4 Experiments

We conducted empirical experiments in order to evaluate the proposed optimized MV clustering-based recommendation approach. Two main research questions have been studied: (1) demonstrate the added value of using BSO optimization on clustering for the different algorithms (KCF, KSocF, KHybF); and (2) compare the optimized MV approach with existing baselines and related work. In these evaluations, we will also show how social information, composed eventually of several features, can contribute to the improvement of the recommendation.

The experiments have been done using two real-world data sets, namely: Flixster1 and FilmTrust2. Flixster.com is a movie sharing and discovering website where users can report their movie ratings (in the range from 0.5 to 5.0 with step 0.5) and where the trust information is symmetric. FilmTrust allows users to share movie ratings and explicitly specify other users as trusted neighbours. Ratings are ranged from 0.5 to 4.0 with step 0.5. For comparison purposes, we used the same data sample sizes as those used in [Guo, 15]. The statistics of these sample data sets are shown in Table 1:

Dataset	# Users	#Items	#Ratings	#Trust	Density (%)
Flixster	5000	13527	264,540	2898	0.39
FilmTrust	1508	2071	35,497	2853	1.14

Table1: Statistics of the sample datasets

[1] Flixster.com
[2] http://www.librec.net/datasets.html.

4.1 Metrics

The performance evaluation of the different algorithms is measured in terms of recommendation accuracy. Mean absolute error (MAE) and root mean square error (RMSE) have been used as they are the most popular predictive metric to measure the closeness of predictions relative to real scores (smaller values indicate better accuracy).

$$MAE = \frac{\sum_{u,i \in \Omega} |r_{u,i} - p_{u,i}|}{|\Omega|} \tag{13}$$

$$RMSE = \sqrt{\frac{\sum_{u,i \in \Omega} (r_{u,i} - p_{u,i})^2}{|\Omega|}} \tag{14}$$

where:
Ω: is the set of test assessments and $|\Omega|$ indicates the cardinality of the set Ω;
$r_{u,i}$: is the rating given by the user *u* on the item *i;* and
$p_{u,i}$: is the rating prediction of the user *u* on the item *i.*

4.2 Baselines

To demonstrate the effectiveness of our optimized multiview clustering approach, we have implemented the following baselines and related work, using Kmedoids and CLARANS algorithms:

- **KCF:** is the clustering-based CF algorithm, where users are clustered according to the rating information by k-medoids or ClARANS algorithms, and item predictions are generated using similarity as user weights.
- **KNN-CF:** is the convolutional CF algorithm, where users are clustered according to the k nearest neighbor algorithm [Sarwar, 01].
- **KSocF:** is the clustering-based SocF algorithm, where users are clustered according to the social distances (trust and friendship) by k-medoids or ClARANS algorithms.
- **KTrust:** this baseline is a specific case of KSoF considering only the Trust information.
- **KCFTrust (KCFT):** is a variant of the KCF method that computes user weights by the harmonic mean of similarity and trust for rating prediction.
- **KCFSoc:** is a variant of the KCF method that computes user weights by the harmonic mean of similarity and social information (trust and friendship).
- **KNN-CFSoc:** is a variant of KNN-CF method that computes user weights by the harmonic mean of similarity and social information for rating prediction. This algorithm has been considered as a naive classification method, which is independent of the principle of multiview and partitioning of users by clustering algorithms.
- **MV-Trust:** is the multiview k-medoids method proposed by [Guo, 15] that clusters users using both ratings and trust information.

- **MV-Soc:** is the multiview method using both ratings and social information (trust and friendship).
- **BSO-MV-Soc:** is our approach, that optimizes the MV method using both ratings and social information.
- **BSO-MV-Trust:** is a variant of our approach, that optimizes the MV method using both ratings and trust information.

4.3 Evaluation results

4.3.1 Contribution of optimisation on clustering-based recommendation

We have studied the contribution of optimization on the different clustering-based recommendation algorithms. This evaluation was carried out using FilmTrust and Kmedoids algorithm, where the number of clusters K is varied from 10 to 100. Figure 2 shows that BSO-KCF gives better performance than KCF. The optimized selection of initial medoids allows an improvement of the clustering and therefore enhances the recommendation accuracy.

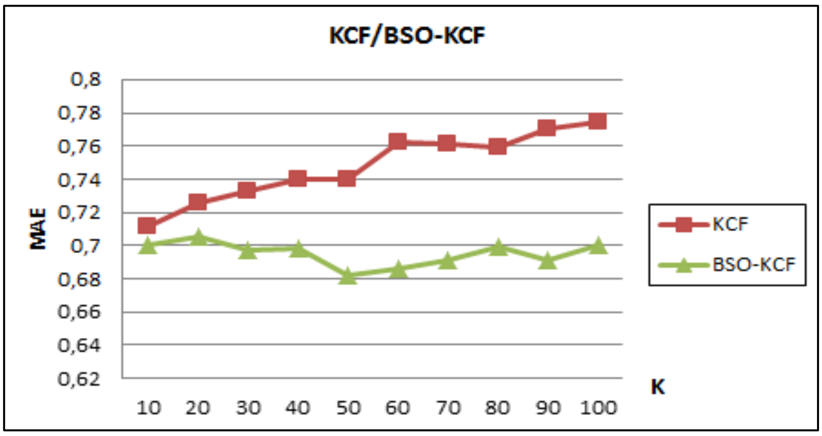


Figure 2: Evaluation of KCF and BSO-KCF

As illustrated in Figure 3, the optimization of KSocF gives better performance, considering the two variants of social information: (1) Trust; and (2) trust with friendship.

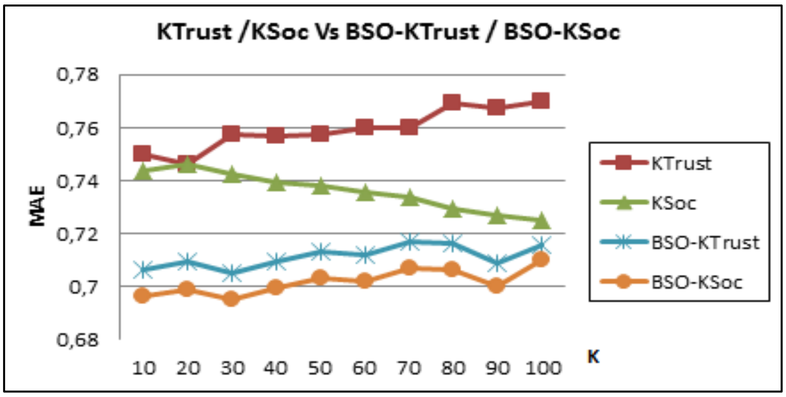


Figure 3: Evaluation of KTrust / KSoc and BSO-KTrust / BSO-KSoc

In order to evaluate the impact of optimization on clustering-based hybrid algorithms, namely: KCFT and KCFSoc, we compared them with the BSO-KCFT and BSO-KCFSoc algorithms respectively.

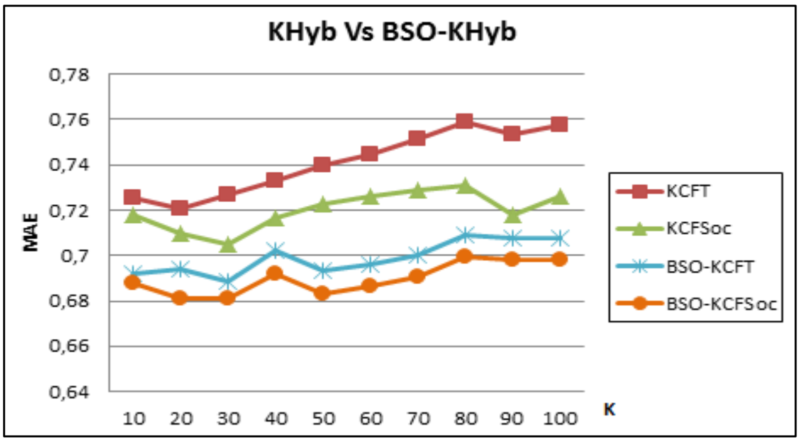


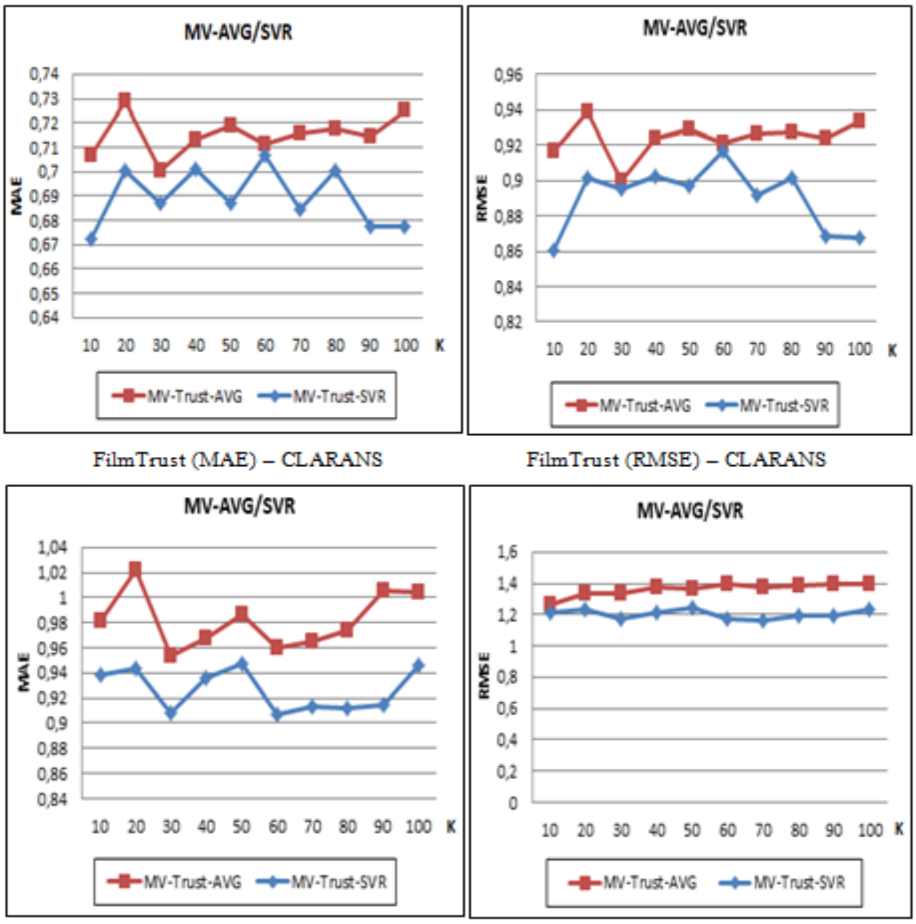
Figure 4: Evaluation of KCFT / KCFSoc and BSO-KCFT / BSO-KCFSoc

The results obtained demonstrate the advantage of using BSO on the clustering-based recommendation algorithms. Figure 4 shows that the better recommendation accuracy is obtained with BSO-KCFSoc.

4.3.2 Contribution of the SVR module on the MV clustering method

The authors in [Guo, 15] demonstrated the effect of the SVR module on the MV method using Kmedoids algorithm. This module significantly performs better than the harmonic average (AVG) method. In order to confirm this contribution, we have performed this same evaluation using CLARANS algorithm.

Figure 5 demonstrates that this module performs better than AVG, for the two datasets FilmTrust and Flixster, as it has achieved the highest recommendation accuracy (i.e. lowest MAE and RMSE) with the variation of the number of clusters.



FilmTrust (MAE) – CLARANS FilmTrust (RMSE) – CLARANS

Flixster (MAE) – CLARANS Flixster (RMSE) – CLARANS

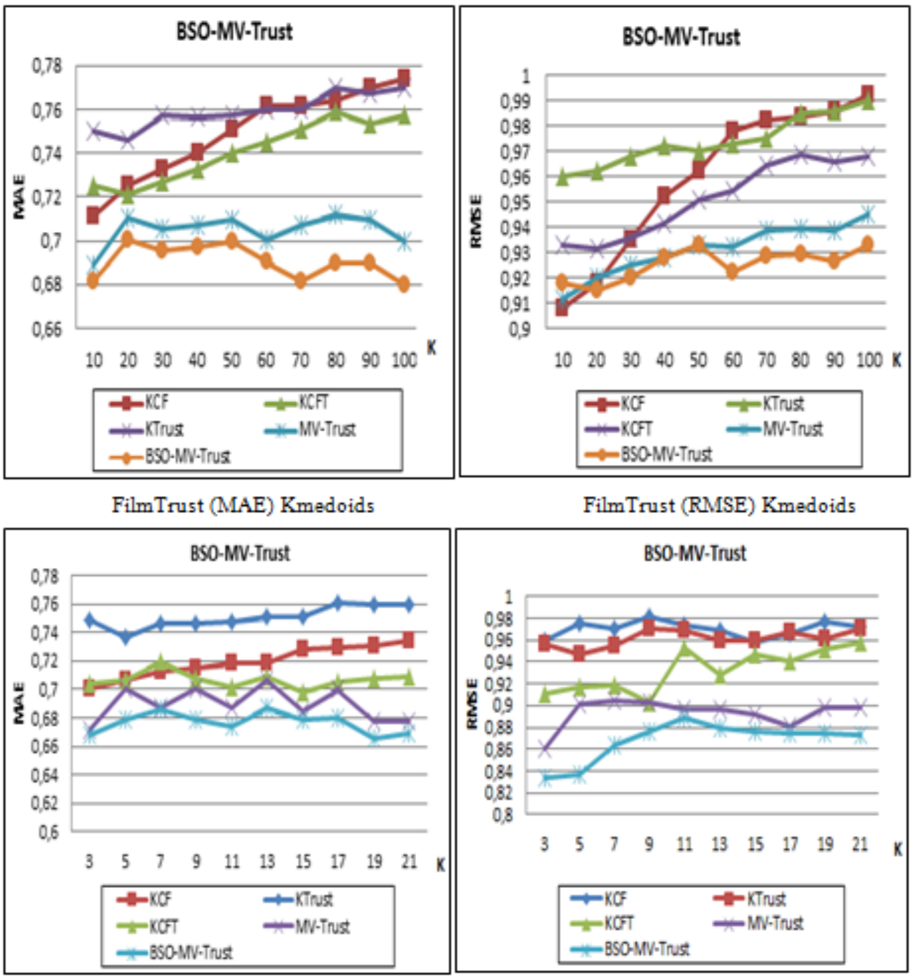
Figure 5: Evaluation of the SVR module using CLARANS

4.3.3 Contribution of the optimisation on the MV clustering method

We compared the BSO-MV algorithm with the baselines and related work, considering the two variants of social information (trust / trust and friendship). Figure 6 illustrates the results obtained with BSO-MV-Trust using FilmTrust with Kmedoids and CLARANS clustering algorithms.

This evaluation allowed us to show the performance of the BSO-MV algorithm, which reached a better MAE value equal to 0.6711 for K equal to 70 with Kmedoids and 0.6651 for K equal to 19 with CLARANS for the BSO-MV-Trust algorithm. For the same values of the number of clusters K, the values of MAE are equal to 0.6888 and 0.6774 respectively with the MV-Trust algorithm.

On the other hand, we can see that the KCFT algorithm gives better performance than the two algorithms KCF and KTrust, showing that the integration of the Trust information has improved the recommendation accuracy of the KCF. However, the MV remains better than KCFT hybridization, demonstrating the contribution of multiview clustering. The BSO-MV algorithm confirms its effectiveness by improving the performance of MV due to the optimized selection of the intial medoids before starting the multiview clustering.



FilmTrust (MAE) Kmedoids — FilmTrust (RMSE) Kmedoids

FilmTrust (MAE) CLARANS — FilmTrust (RMSE) CLARANS

Figure 6: Evaluation of the BSO-MV-Trust algorithm using Kmedoids and CLARANS

We carried out the same evaluation, considering social information (trust and friendship). Similarly, Figure 7 illustrates the results obtained with BSO-MV-Soc using FilmTrust with Kmedoids and CLARANS clustering algorithms. The results obtained confirm the previous evaluation for all the algorithms, which have progressed in the same way.

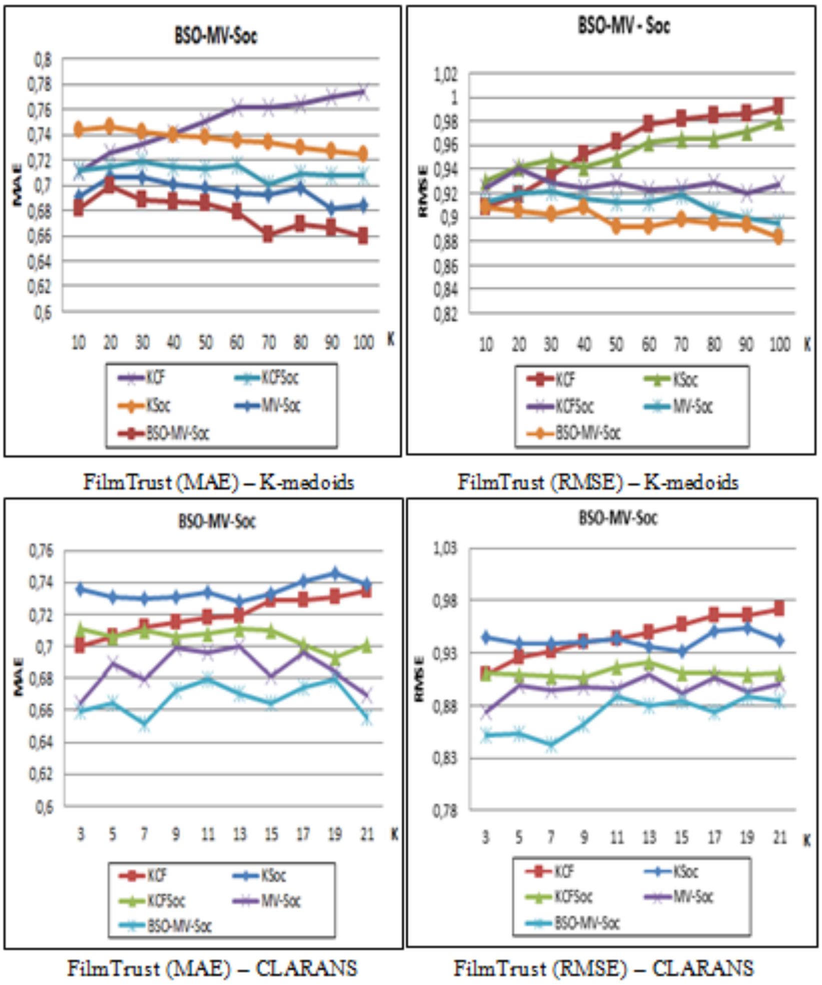


Figure 7: Evaluation of the BSO-MV-Soc algorithm using Kmedoids and CLARANS

Moreover, by analysing the results obtained from the two previous evaluations (Figures 6 and 7), we can deduce that by including friendship with trust information, the MV-Soc and BSO-MV-Soc algorithms gave better results than MV-Trust and BSO-MV-Trust respectively, for both Kmedoids and CLARANS clustering algorithms. With Kmedoids we obtained a better MAE value equal to 0.6561 with K equal to 70 for the BSO-MV-Soc algorithm, against an MAE value equal to 0.6825 with MV-Soc. Similarly, for CLARANS, we obtained a better MAE value equal to 0.6517 with K equal to 7 for the BSO-MV-Soc algorithm, against an MAE value equal to 0.6793 with MV-Soc.

4.3.4 Comparison between MV and classification-based approaches

In order to further show the contribution of our approach, we have compared the MV and the BSO-MV algorithms with two classification-based algorithms, namely: (1) KCFSoc, the hybrid-based clustering algorithm based on an unsupervised technique that uses the Kmedoids algorithm; and (2) KNN-CFSoc, the hybrid algorithm based on a supervised technique that uses the K-Nearest Neighbours algorithm.

Figure 8 demonstrates the performance of the BSO-MV which outperformed both supervised and unsupervised classification techniques in terms of the MAE and RMSE metrics.

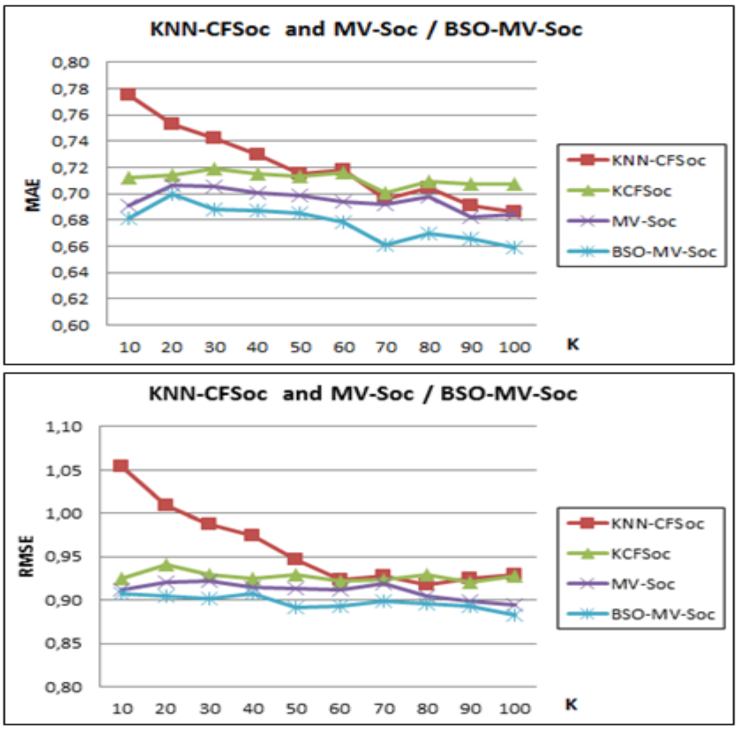


Figure 8: BSO-MV / MV and supervised / unsupervised classification based recommendation algorithms

The obtained results demonstrate the importance of improving the classification of users, which would make it possible to recommend the most appropriate items to them. We can notice that the MV method performs better than the classification-based algorithms. On the other hand, the use of the BSO algorithm allows us to generate better partitions (refining the quality according to the objective function), increasing the accuracy of the recommendations.

4.4 Discussion

4.4.1 Results interpretation

The results experiments demonstrate the contribution of optimisation on all the clustering-based recommendation algorithms. We have obtained the best recommendation accuracy with BSO-KCF, BSO-KTrust, BSO-KSocF, BSO-KCFT and BSO-KCFSoc, compared respectively to KCF, KTrust, KSocF, KCFT and KCFSoc.

On the other hand, the evaluation carried out demonstrates that the MV algorithm has given better performance than the other baselines and that BSO-MV outperformed all these algorithms. The optimized selection of intial medoids allowed an improvement of the MV algorithm, for both Kmedoids and CLARANS clustering algorithms. This result remains valid when using different variants of the MV algorithm (MV-Trust / BSO-MV-Trust and MV-Soc / BSO-MV-Soc).

Furthermore, all the previous evaluations shows that CLARANS algorithm performs better than K-medoids and that social information (trust and friendship) has significantly improved the recommendation accuracy of all the algorithms (MV and BSO-MV). Accordingly, we deduce that with more social features, the social distance will be more precise and expressive and therefore social clustering will be improved. This demonstrates the better results of MV-Soc / BSO-MV-Soc compared with MV-Trust / BSO-MV-Trust. Figure 9 summarizes these comparison results.

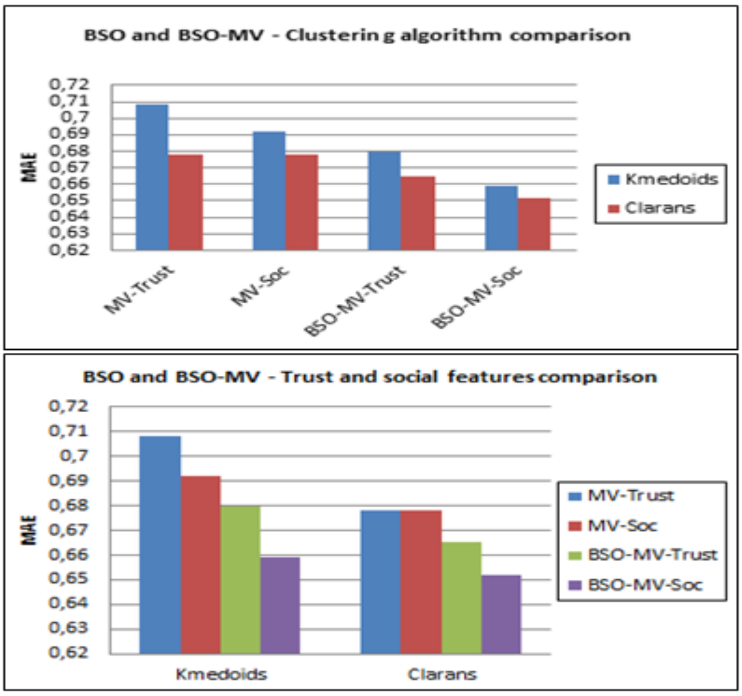


Figure 9: Comparison between MV and BSO-MV

Finally, the MV / BSO-MV algorithms have been compared with KCFSoc and KNN-CFSoc. The results obtained confirmed the performance of our proposal compared with both, supervised and unsupervised classification-based hybrid recommendation algorithms. The best partitioning offers the advantage of grouping users who are most similar to each other in terms of the collaborative and social dimensions, giving rise to the best predictions for these users. This hypothesis is demonstrated since we obtained the best evaluation values in terms of the MAE and RMSE metrics.

Table 2 gives an overview of the overall results in terms of the average and best MAE and RMSE values for the different algorithms. AVG-MAE (resp. AVG RMSE) is the average of the MAE (resp. RMSE) values obtained with the variation of the number of clusters "k", while Best-MAE (resp. Best-RMSE) is the minimum MAE (resp. RMSE) value obtained with the variation of the number of clusters. The kmedoids algorithm has been used for the clustering-based algorithms.

Algorithms	**MAE**		**RMSE**	
	AVG-MAE	**Best-MAE**	**AVG-RMSE**	**Best-RMSE**
KCF	0.7492	0.7110	0.9597	0.9080
KNN-CF[Sarwar, 01]	0.7718	0.7010	1.0238	0.9165
KTrust	0.7595	0.7460	0.9741	0.9600
KSoc	0.7360	0.7250	0.9557	0.9305
KNN-Soc	0.7273	0.7076	0.9099	0.9219
KCFT	0.7411	0.7207	0.9515	0.9318
KCFSoc	0.7115	0.7003	0.9273	0.9202
KNN-CFSoc	0.7213	0.7062	0.9595	0.9178
BSO-KCFT	0.6992	0.6987	0.9350	0.9255
BSO-KCFSoc	0.6897	0.6907	0.9210	0.9150
MV-Trust [Guo, 15]	0.6894	0.6888	0.9024	0.9125
MV-Soc	0.6852	0.6825	0.9013	0.8951
BSO-MV-Trust	0.6767	0.6711	0.8974	0.9150
BSO-MV-Soc	**0.6720**	**0.6561**	**0.8908**	**0.8831**

Table 2: Comparison between the different algorithms

4.4.2 Future extension and adaptation of our approach

The results obtained on the FilmTrust and Flixster databases are promising. However our approach could be extended in several ways:

1) ***Adaptation of our approach***: We are convinced that our approach will be useful for other domains as well. It would be possible to exploit the optimized MV method in other social networks such as CiteULike, for example, for the recommendation of articles in scientific social networks. Recently, [Berkani, 20b] proposed a novel hybrid algorithm for the scientific articles recommendation based on the MV clustering approach. The authors considered the improved collaborative and content-based filtering algorithms using respectively friendships and tags.
2) ***Improvement of the MV method:*** we believe that the results could be enhanced using other optimization algorithms such as the Genetic or BAT algorithms.

3) ***Enrichment of the social dimension:*** other features can be considered such the influence and credibility of users in the social network. Furthermore, it would be interesting to consider the use of implicit trust links to enrich user trust information.
4) ***Hybridization with other recommendation algorithms:*** our algorithm could be combined with other algorithms such as the semantic filtering and build new prediction models, using other Machine Learning algorithms that can be more efficient than the SVR module.
5) ***Evaluate the performance for cold users:*** it is important to investigate the effectiveness of our approach dealing with cold-start users.
6) ***Automatic assignment of weights:*** we have empirically tested several combinations of the weights of the parameters related to the social and hybrid algorithms. Performing an automatic optimization of the values of these weights should improve the recommendation accuracy, especially if we use several features to represent social information and other algorithms such as the semantic filtering.

5 Conclusions

We proposed in this article an optimized multiview clustering-based recommendation approach in social networks, where users are iteratively clustered from the views of both rating patterns and social information. We have considered trust and friendship information for the social filtering. The multiview clustering is optimized using the Bees Swarm optimization algorithm. Different clustering algorithms have been considered (K-medoids, PAM and CLARANS).

The experimental results showed that: (1) the optimization has significantly improved all the clustering-based algorithms; (2) the MV algorithms (MV-Trust / MV-Soc and BSO-MV-Trust / BSO-MV-Soc) have performed better than the supervised and unsupervised classification-based algorithms (KNN-CFSoc and KCFSoc); (3) the BSO-MV-Soc (resp. BSO-MV-Trust) has outperformed the MV-Soc (resp. MV-Trust) in terms of the recommendation accuracy, using different clustering algorithms. Furthermore, the evaluations have shown that the recommendation accuracy increases when using more features for social information and that the CLARANS algorithm has performed better than Kmedoids.

As perspectives to this work, it would be interesting to enrich the social information with other features (such as the influence and the credibility of users in the social network) and consider the use of implicit trust links. Moreover, we plan an in-depth evaluation of our approach by investigating the effectiveness of our approach dealing with cold-start users and developing other clustering and optimization algorithms. Finally, in order to further improve the recommendation accuracy, it would be interesting to integrate other dimensions in our MV clustering algorithm such as the semantic view for better representation of users' interests and preferences.

Acknowledgements

We express our deepest gratitude to the Editors of J.UCS' special issue on Advances and Challenges for Model and Data Engineering and the anonymous reviewers that have greatly helped us to improve our article from its original version. We note that the work proposed in this article is an extended and improved version of the paper previously presented at The Ninth International Conference on Model and Data Engineering (MEDI 2019).

We also express our deepest gratitude to the Algerian Directorate-General for Scientific Research and Technological Development (DGRSDT), for the support of this research under the grant number C0662300.

References

[Adomavicius, 05] Adomavicius, G., Tuzhilin, A.: "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions"; IEEE Transaction Knowledge Data Engineering (TKDE), 17 (2005), 734-749.

[Bellogín, 12] Bellogín, A., Parapar, J.: "Using graph partitioning techniques for neighbour selection in user-based collaborative filtering"; Proceedings of the 6th ACM Conference on Recommender Systems (RecSys), (2012), 213-216.

[Berkani, 19] Berkani, L.: "Social-Based Collaborative Recommendation: Bees Swarm Optimization Based Clustering Approach"; edited by Schewe, K-D Singh, N.K.: Model and Data Engineering - 9th International Conference, MEDI 2019, Toulouse, France, October 28-31, 2019, Proceedings. Lecture Notes in Computer Science 11815, Springer (2019), ISBN 978-3-030-32064-5, 156-171.

[Berkani, 20a] Berkani, L., Betit, L., Belarif, L.: "A Multiview Clustering Approach for the Recommendation of Items in Social Networking Context"; Proceedings of the 4th Conference on Computing Systems and Applications (CSA'20), December 14th, Algiers, Algeria, (2020).

[Berkani, 20b] Berkani, L., Hanifi, R., Dahmani, H.: "Hybrid Recommendation of Articles in Scientific Social Networks Using Optimization and Multiview Clustering"; Proceedings of the Third International Conference on Smart Applications and Data Analysis for Smart Cyber-Physical Systems (SADASC'20), Marrakech, Morocco, 25th - 26th June, (2020), 117-132.

[Chen, 19] Chen, J., Wang, C., Shi, Q., Feng, Y., Chen, C.: "Social recommendation based on users' attention and preference"; Neuro-Computing, 341 (2019), 1-9.

[DuBois, 09] DuBois, T., Golbeck, J., Kleint, J., Srinivasan, A.: "Improving recommendation accuracy by clustering social networks with trust"; Recommender Systems Social Web, (2009), 1-8.

[Guo, 15] Guo, G. Zhang, J. Yorke-Smith, N.: "Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems"; KBS Journal, 74 (2015), 14-27.

[Guo, 14] Guo, G., Zhang, J., Thalmann, D.: "Merging trust in collaborative filtering to alleviate data sparsity and cold start"; Knowledge-Based Systems 57 (2014), 57-68.

[Guo, 18] Guo, I., Luo, J., Dong, K. Ming Yang.: "Differentially private graph-link analysis based social recommendation", Sciences, 463-464 (2018), 214-226.

[He, 14] He, X., Kan, M.Y., Xie, P., Chen, X.: Comment-based multi-view clustering of web 2.0 items. In: International World Wide Web Conference WWW 2014, Seoul, Korea, 7–11 April 2014, ACM (2014) 771–781

[Jamali, 10] Jamali, M., Ester, M.: "A matrix factorization technique with trust propagation for recommendation in social networks"; Proceedings of the 4th ACM Conference on Recommender Systems (RecSys), (2010), 135–142.

[Kaufman, 87] Kaufman, L., Rousseeuw, P.J.: "Clustering by means of Medoids, in Statistical Data Analysis Based on the Norm and Related Methods"; edited by Y. Dodge, North-Holland (1987), 405-416.

[Lai, 19] Lai, C-C., Lee, S-J., Huang. H-L.: "A social recommendation method based on the integration of social relationship and product popularity"; International Journal of Human-Computer Studies, 121 (2019), 42-57.

[Ma, 09] Ma, H., King, I., Lyu, M.: "Learning to recommend with social trust ensemble"; Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), ACM (2009), 203-210.

[Ma, 11] Ma, H., Zhou, D., Liu, C., Lyu, M., King, I.: "Recommender systems with social regularization"; Proceedings of the fourth ACM international conference on Web search and data mining. ACM, New York, (2011), 287-296.

[Massa, 04] Massa, P., Avesani, P.: "Trust-aware collaborative filtering for recommender systems"; On the move to meaningful internet systems, Springer, Berlin, (2004), 492-508.

[Massa, 07] Massa, P., Avesani, P.: "Trust-aware recommender systems"; Proceedings of the 2007 ACM conference on recommender systems, (2007), 17-24.

[Nazemian, 12] Nazemian, A., Gholami,H., Taghiyareh, F.: "An Improved Model of Trust- aware Recommender Systems Using Distrust Metric"; Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Istanbul, Turkey, August 26-29, (2012),1079-1084.

[Ng, 02] Ng, R.T., Han, J.: "CLARANS: A method for clustering objects for spatial data mining"; IEEE Transactions on Knowledge and Data Engineering, (2002), http://dx.doi.org/10.1109/TKDE.2002.1033770.

[Park, 16] Park, C. Kim, D. Oh, J. Yu, H.: "Improving top-K recommendation with truster and trustee relationship in user trust network"; Inform. Sci. 374 (2016) 100-114.

[Pazzani, 07] Pazzani, M.J., Billsus, D.: "Content-based recommendation systems"; edited by Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.); The Adaptive Web; LNCS, 4321 (2007), 325-341, Springer, Heidelberg, https://doi.org/10.1007/978-3-540-72079-9_10.

[Pham, 11] Pham, M., Cao, Y., Klamma, R., Jarke, M.: "A clustering approach for collaborative filtering recommendation using social network analysis"; Journal for Universal Computer Science, 17(2011), 583–604.

[Salah, 16] Salah, A., Rogovschi, N., Nadif, M.: "A dynamic collaborative filtering system via a weighted clustering approach"; Neuro-Computing, 175 (2016), 206-215.

[Sarwar, 01] Sarwar, B., Karypis, G., Konstan, J., Reidl, J. : "Item-based collaborative filtering recommendation algorithms"; Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, (2001), 285-295.

[Sarwar, 02] Sarwar, B. Karypis, G. Konstan, Riedl, J.: "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering»; Proceedings of the 5th Inter. Conference on Computer and Information Technology (2002), 158-167.

[Selvi, 17] Selvi, C., Sivasankar, E.: "A Novel Optimization Algorithm for Recommender System using Modified Fuzzy C-means Clustering Approach" ; Soft Computing (2017), 1- 16.

[Sheugh, 15] Sheugh, L., Alizadeh, S.H.: "Merging similarity and trust based social networks to enhance the accuracy of trust-aware recommender systems"; Journal of Computer & Robotics, 8, 2 (2015), 43-51.

[Singla, 08] Singla, P., Richardson, M.: "Yes, there is a correlation: from social networks to personal behavior on the web"; Proceedings of the 17th International Conference on World Wide Web (WWW) (2008), 655–664.

[Sun, 15] Sun, Z. Han, L. Huang W.g, Wang, X. Zeng, X. Wang M., Yan, H.: "Recommender systems based on social networks"; The Journal of Systems and Software, 99 (2015), 109-119.

[Wang, 14] Wang, X., Huang, W.: "Research on Social Regularization-based Recommender Algorithm"; Mathematical and Computer Modelling, Elsevier, 1(2014), 77-80.