



# Using a Social-Based Collaborative Filtering with Classification Techniques

Lamia Berkani 

Laboratory for Research in Artificial Intelligence (LRIA),  
Department of Computer Science,  
USTHB University, Bab Ezzouar, Algiers, Algeria  
lberkani@usthb.dz

**Abstract.** In this paper, a social-based collaborative filtering model named SBCF is proposed to make personalized recommendations of friends in a social networking context. The social information is formalized and combined with the collaborative filtering algorithm. Furthermore, in order to optimize the performance of the recommendation process, two classification techniques are used: an unsupervised technique applied initially to all users using the Incremental K-means algorithm and a supervised technique applied to newly added users using the K-Nearest Neighbors algorithm (K-NN). Based on the proposed approach, a prototype of a recommender system is developed and a set of experiments has been conducted using the Yelp database.

**Keywords:** Recommendation of users · Collaborative filtering  
Social filtering · Classification · K-means · Incremental K-means  
K-nearest neighbors

## 1 Introduction

Due to the powerful ability of solving information overload, recommender systems have attracted a lot of attention in the last decade [1]. Recommender systems automatically predict the preferences of users for some given items, in order to provide them with the useful recommendations. There are three main methods to generate recommendations [2]: content-based filtering, collaborative filtering (CF) and a hybrid approaches combining in different way the two aforementioned algorithms [3]. Content-based filtering algorithms identify features that appear in item contents that users have appreciated before, thereafter suggest more items which contain these relevant features to users. CF is a widely-exploited technique in recommender systems to provide users with items that well suit their preferences [2]. The basic idea is that a prediction for a given item can be generated by aggregating the ratings of users with similar interest. However, one of the shortcomings of this algorithm is the sparsity and cold start problems due to insufficient rating [4].

Nowadays, interactions and sharing knowledge over the Social Web have become the main way of communication between people. The study of social-based recommender systems has emerged as users are unable to reach the relevant information due to the exponential growth of data.

In this paper, we focus on the friends' recommendation in social networks. Our approach first enhanced the CF recommendations with social information. The social dimension is characterized by some social-behavior metrics such as friendship, commitment and trust degrees between users, to cope with problems of rating diversity and sparsity. Then, in order to optimize the performance of the recommendation process, we have used two classification techniques along with collaborative and social similarity measures: (1) an unsupervised technique using the Incremental K-means algorithm, applied initially to all users of the social network; and (2) a supervised technique using the K-Nearest Neighbors algorithm, applied to new users.

The remainder of the paper is organized as follows: Sect. 2 gives an overview on some related work about social recommendation. Section 3 presents our recommendation approach. In Sect. 4, we give an overview on the experimentation we carried out. Finally, we give the conclusion with some perspectives in Sect. 5.

## 2 Related Work

The review of the literature about social recommendation shows that much work proposes various approaches to provide personalized recommendations to users. For instance, Liu and Lee [5] developed a way to increase recommendation effectiveness by incorporating social network information into CF. Chang and Chu [6] proposed a recommendation approach which calculates similarity among users and users' trust-ability and information collected from the social networks. Banati et al. [7] explored the role of explicit social relationship by presenting two novel similarity metrics. The first metric is based on the social behavior that measures similarity between two users on the basis of "how similar they are in their social relationship". The second metric integrates the social similarity with the interest similarity between two users. Su et al. [8] proposed a music recommender system integrating social information (to cope with problems of rating diversity and sparsity) and collaborative information (to cope with problem of lack of rating information) to predict users' preferences.

In order to improve the performance of social recommendation, some studies have used classification techniques, for instance: Guo et al. [9] developed a multi-view clustering method through which users are iteratively clustered on the basis of rating patterns in one view and social trust relationships in the other. Najafabadi et al. [10] tried to improve the accuracy of CF recommendations using clustering and association rules mining on implicit massive data.

## 3 A Social-Based CF Approach

We propose in this section an enhanced-based CF algorithm by combining collaborative and social information in the recommendation process.

### 3.1 The User-User Based Collaborative Filtering (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 neighbors for a given user, using the usage matrix. The matrix can be constructed using the ratings of users on items. We used the *Pearson Correlation function* to calculate the similarity between users. The collaborative similarities between users allow the identification of neighborhoods and therefore building communities of users who evaluate in the same way, according to a given threshold.

### 3.2 The Social Filtering (SocF)

The social information of the user's profile is based on two metrics: (1) *the friendship*; and (2) *the credibility degree*. Two parameters are considered to identify the credibility of an active user  $u$ : Commitment and trust degrees.

#### 3.2.1 Friendship Metric

This metric computes the similarity weight between two users  $u_1$  and  $u_2$ , based on their social relationships which is defined as the size of the intersection divide by the size of the union of friend sets:

$$Sim_{soc}(u_1, u_2) = \frac{F(u_1) \cap F(u_2)}{F(u_1) \cup F(u_2)} \quad (1)$$

where:

$F(u_1)$ , respectively  $F(u_2)$ : represent the number of friends of  $u_1$ , respectively  $u_2$ .

#### 3.2.2 Commitment Degree

Two parameters are considered: (1) *the participation degree* of an active user  $u$ , including the degree or rate of evaluations he carried out; and (2) *the sociability degree* represents his friendship rate in the social network.

$$Commitment(u) = \alpha_1.Participation(u) + \beta_1.Sociability(u) \quad (2)$$

where:  $\alpha_1$  and  $\beta_1$  are weights that express a priority level, with  $\alpha_1 + \beta_1 = 1$

- **The participation degree of  $u$ :** concerns mainly the degree of performed evaluations by  $u$ . This degree is calculated based on the number of evaluations performed by  $u$ ,  $NbEval(u)$ , according to all the evaluations carried out in the system,  $NbTotalEval$ .

$$Participation(u) = \frac{NbEval(u)}{NbTotalEval} \quad (3)$$

- **The sociability degree of  $u$ :** this degree is calculated based on the number of friends of  $u$  according to all the registered users of the social network.

$$Sociability(u) = \frac{NbFriends(u)}{NbUsers - 1} \quad (4)$$

### 3.2.3 Trust Degree

This metric takes into account the seniority level of  $u$  and his degree of competence in the social network using the following formula:

$$Trust(u) = \alpha_2.Seniority(u) + \beta_2.Competency(u) \quad (5)$$

where:  $\alpha_2, \beta_2$  are weights that express a priority level, with  $\alpha_2 + \beta_2 = 1$

- **The seniority level of  $u$ :** is calculated based on the date of his registration in the social network.
- **The Competency degree of  $u$ :** is calculated on two steps, based the following assumption [11]: “A friend is competent if he has evaluated correctly the resources compared to their average evaluations in the social network”:
- **Step 1:** Calculate the competency degree of a friend  $F$  regarding a given item  $R_j$ . We start by calculating the average of ratings for each item. Then, we compare the rating given by  $F$  for the same item with the average value.

$$Competency(F, R_j) = \begin{cases} \frac{avg(R_j)}{v_{ij}} & \text{if } Avg(R_j) \leq v_{ij} \\ \frac{v_{ij}}{avg(R_j)} & \text{if } v_{ij} \leq Avg(R_j) \end{cases} \quad (6)$$

where:

$avg(R_j)$ : is the average evaluation of the item regarding all the users.

$v_{ij}$ : is the evaluation of the friend  $F$  for  $R_j$

- **Step 2:** Calculate the global trust degree of the friend, using this formula:

$$Trust(F) = \frac{1}{n} \sum_{j=1}^n Degree_{comp}(F, R_j) \quad (7)$$

where:  $n$  represents the number of items evaluated by the friend.

## 4 Combining SBCF Approach and Classification Techniques

The main objective of the classification in our recommendation process is to group similar users according to the collaborative/social dimensions.

This will reduce the search space for the identification of neighbors and allow us to cluster the different groups. So, each user of the social network will have both a collaborative class and a social one. Moreover, if any user has friends but has not yet made enough evaluations, the system can recommend him other friends based only on

the social dimension. Similarly, if a user has performed enough evaluations but has not yet added any friend, the system can recommend him new friends based on the collaborative dimension.

#### 4.1 Unsupervised Classification Using Incremental K-means

We have first used the K-means classical algorithm, which is considered to be the most popular algorithm because of its simplicity and its ability to handle large data sets. However, the disadvantage of this algorithm is that each initialization corresponds to a different solution (local optimum) that can be far from the optimal solution (global optimum). A naive solution to this problem is to run the algorithm several times with different initializations and to retain the best grouping found. The use of this solution is limited because of its cost and that in some cases we cannot explain the partitioning of the clusters where we can find an optimal partition in a single execution [12].

Then, we applied the incremental K-means proposed in [12] which is a variant of the K-means algorithm. This algorithm eliminates the initialization problem of centroids. It is based on the principle of global K-means, which aims to achieve an optimal solution, i.e. instead of having a single center of the whole population (global K-means). The algorithm chooses two objects, where each being the center of a cluster so that the two latter are the more distant. The next step is to choose the next center. A simple function allows calculating the distance between the center of the cluster and its neighbors. The farthest element of the center is the elected candidate to be the new centroid. After this operation the clusters are reconstructed by affecting the set of objects where the distance between the object and the center is minimal. This action is repeated until  $K$  clusters are obtained.

##### 4.1.1 Collaborative Incremental K-means (Col-Inc-K-means)

The set of users to be classified includes all the social network users. The application of col-Inc-K-means implies the application of the incremental K-means algorithm using the Pearson similarity functions for assigning objects to the corresponding clusters.

##### 4.1.2 Social Incremental K-means (Soc-Inc-K-means)

The set of users to be classified includes all the social network users. The application of soc-Inc-K-means implies the application of the incremental K-means algorithm using the social similarity measure already presented in the previous section.

#### 4.2 Supervised Classification Using K-NN Algorithm

We have chosen the k-nearest neighbors algorithm (K-NN), an instance-based classification method. The complexity of this algorithm is equal to  $O(n)$ , where  $n$  being the total number of users of the training set. This algorithm determines for each newly added user, the list of nearest neighbors among those already classified in the clustering step (unsupervised classification). The newly issued ratings cannot be quickly used for updating the classification obtained by Incremental K-Means as this operation is costly in terms of computation time. To overcome this obstacle, we classify newer users using both collaborative and social K-NN algorithms, adapted respectively to the collaborative

and social classification. The collaborative-based K-NN algorithm (resp. social-based K-NN algorithm) uses the same collaborative distance measure already used with the Col-Inc-K-means (resp. Soc-Inc-K-means).

## 5 Recommendation Algorithm

We present in this section our algorithms for friends' recommendation. Collaborative and social clusters have already been calculated, before running this algorithm.

### 5.1 SBCF Algorithm

The following is the proposed algorithm of recommendation combining the SocF and the CF without using the classification techniques.

---

**Algorithm 1:** SBCF Recommendation Algorithm

---

**Input:** User's profile (*U'Profile*), Usage matrix

**Output:** Rec-list /\* A list of recommended friends for a given user *u* \*/

**Begin**

**Rec-list:** =  $\emptyset$ ;

**If** *u* is a new user **Then**

    Rec-list: = Rec-list  $\cup$  {list of leaders}

**Else**

**If** not enough ratings **Then**

      SocF

**Else**

      SBCF/\* combines the CF with the SocF\*/

**EndIf**

**EndIf**

  Sort Rec-list (*u*);

  Display (*u*, Rec-list)

**End.**

---

Leaders are, for example, users who are very active in the social network, i.e. evaluate items correctly (based on the average item evaluations given by other users) and/or those with a significant number of friends, having strong/the most important social affinities in the social network.

*Sort Rec-list (u)*: is the function that sorts in descending order the list of recommended friends for *u* (from the most similar friend to the less similar one).

*Display (u, Rec-list)*: is the function that displays the list of recommended friends *Rec-list* for the user *u*.

The SBCF combination algorithm considers the recommendations made using the CF and then applies the SocF (i.e. the SoF is applied using the generated recommendations of users suggested by the CF).

## 5.2 Recommendation Algorithm Using SBCF and Classification

C-SBCF is the SBCF recommendation of users using the social and collaborative classification techniques.

---

### Algorithm 2: C-SBCF Recommendation Algorithm

---

**Input:**

UserTable /\*contains the collaborative classes (ColClass) and social classes (SocClass) of users\*/

**Output:**

Rec-list /\* A list of recommended friends for a given user  $u$  \*/

**For** each user  $u$  from UserTable **Do**

Rec-list: =  $\emptyset$ ;

$\alpha = 0, \beta = 0$  /\*  $\alpha$  and  $\beta$  represent a weighted levels of the collaborative and social classes\*/

**If**  $u$  has ColClass and SocClass **Then**

$\alpha = 0.5, \beta = 0.5$

**Else IF**  $u$  has ColClass and has no ColClass **Then**

$\alpha = 1, \beta = 0,$

**Else if**  $u$  has SocClass and has no SemClass **Then**

$\alpha = 0, \beta = 1,$

**Else** /\*  $u$  has no ColClass and has no SocClass\*/

Rec-list: = Rec-list  $\cup$  {list of leaders};

Display ( $u$ , Rec-list)

Exit

**End If.**

**For** the rest of users  $u'$  not friends with  $u$  and having same ColClass or SocClass as  $u$  **Do**

$\text{Sim}(u', u) = (\alpha * \text{Sim}_{\text{Col}}(u', u)) + (\beta * \text{Sim}_{\text{Soc}}(u', u))$

$\text{Credibility}(u') = (0.7 * \text{Trust}(u')) + 0.3 * \text{Commitement}(u')$

$\text{Rec-Val}(u', u) = (0.8 * \text{Sim}(u', u)) + 0.2 * \text{Credibility}(u')$

**IF** Rec-Val ( $u', u$ )  $\geq$  Recommendation-Threshold **Then**

Rec-list: = Rec-list  $\cup$  { $u'$ } /\*Add  $u'$  to the recommended list of  $u$ \*/.

**End If.**

**Done.**

Sort Rec-list ( $u$ )

Display ( $u$ , Rec-list)

**Done.**

**End.**

---

## 6 Evaluation

In this section, we conduct experiments on a real dataset to validate the effectiveness of our approach. We used the Yelp social network (<http://www.yelp.com/>), which aims to connect people with local businesses and we chose the “Restaurant” category because of its frequency use in this social network. The experiments carried out have two main objectives:

1. Show the contribution of combining social information with the user-based CF recommendations;
2. Compare the use of K-means and incremental K-means in the recommendation process and show the added value of the K-NN algorithm.

### 6.1 Dataset

Before using the Yelp Database, we have performed some pretreatment operations for the inclusion of implicit data (i.e. interests, preferences, commitment and trust degrees, average rating and number of assessments per restaurant, etc.). The resulting database includes 4823 restaurants, 65 categories and 5436 users who have performed 118,709 assessments on these restaurants (only users who have evaluated more than 9 restaurants have been considered).

### 6.2 Evaluation Metrics

We considered the following evaluation metrics:

- Precision (P): measures the relevance of the recommendations:

$$P = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \quad (8)$$

- Recall (R): measures the ability to make relevant recommendations:

$$R = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \quad (9)$$

- F-measure (F): combines P and R metrics:

$$F = \frac{2 * P * R}{P + R} \quad (10)$$

- Accuracy (A): measures the performance of the system:

$$A = \frac{\text{TruePositives} + \text{FalseNegatives}}{\text{TruePositives} + \text{TrueNegatives} + \text{FalsePositives} + \text{FalseNegatives}} \quad (11)$$



where:

- True Positives (TP): represents the number of recommendations made to users who were originally friends,
- True Negatives (TN): represents the number of recommendations that are not made to users who were not initially friends,
- False Positives (FP): represents the number of recommendations made to users who were not initially friends,
- False Negatives (FN): represents the number of recommendations not made to users who were initially friends.

### 6.3 Evaluation Results

We started first by evaluating the CF using the Pearson similarity function and we have varied the values of similarity rate from 0.1 to 0.9 (Sim-Threshold).

Then we evaluated the SocF by varying the weights of the three parameters: the friendship as well as the commitment and trust degrees. The results of the tests carried out revealed that the combination  $\alpha_1 = 0.1$ ;  $\beta_1 = 0.6$ ;  $\gamma_1 = 0.3$  gives the best results in terms of precision and F-measure than the two other combinations of  $\alpha_1$ ;  $\beta_1$  and  $\gamma_1$  (see Fig. 1), where:  $\alpha_1$ ,  $\beta_1$  and  $\gamma_1$  represent respectively friendship, commitment and trust degrees parameters.

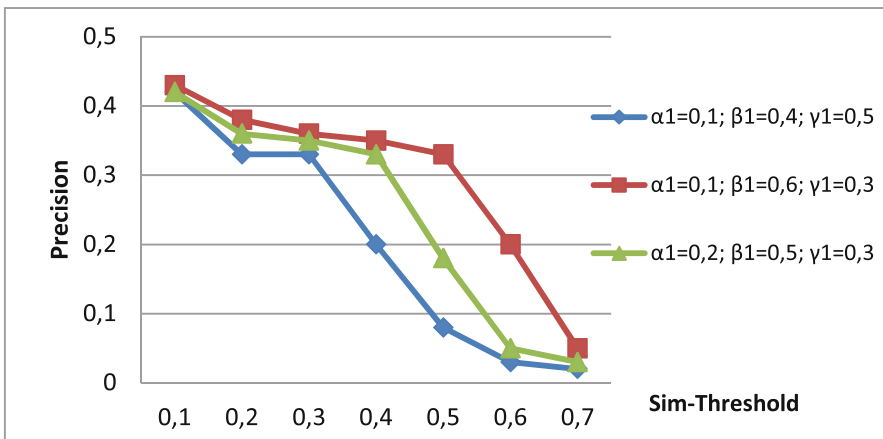


Fig. 1. Identification of the social parameters' weights

Finally, we have evaluated the SBCF using the best parameters for each algorithm. Figure 2 shows that the integration of social information enhanced the CF recommendation accuracy. We have obtained a better precision and F-measure using the SBCF compared to the CF.

In order to see the evolution of the two algorithms K-means and incremental K-means, we have considered in this experimentation only the social classification and

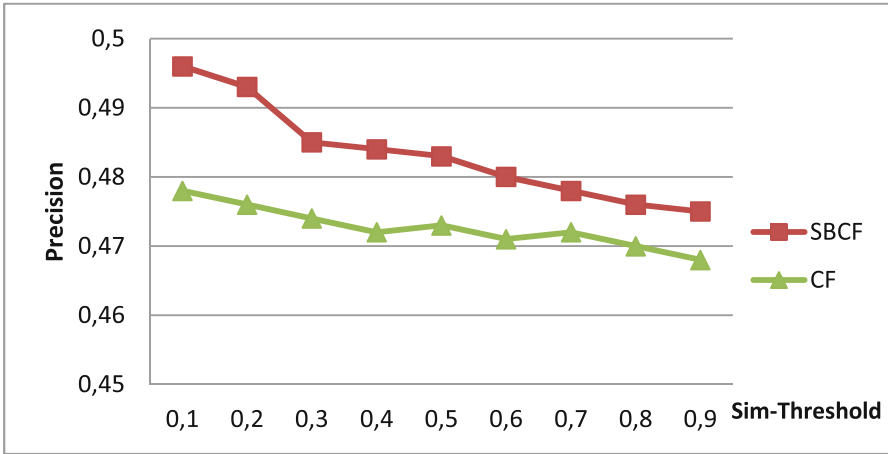


Fig. 2. Contribution of social information on CF recommendations

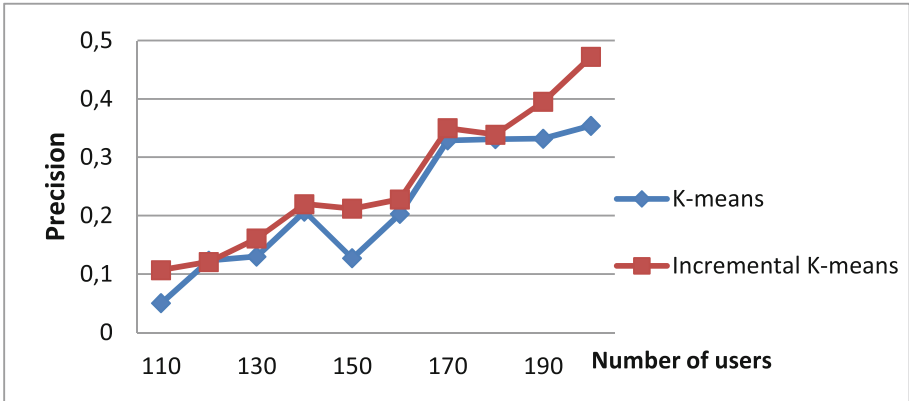
we have simulated the evolution of the social network using a partition of the database including: 200 users, 351 restaurants, 4852 ratings. We fixed the number of social classes ( $K = 3$ ) and the recommendation threshold value to 0.3 and we have varied the number of evaluations (NbE), number of users (NbU) and number of deleted friendships (NbDF) to check whether the system can recommend them again or not. The results obtained are shown in Table 1.

Table 1. Results of the evolution of social K-means and Soc-Inc-K-means

I	NbU	NbE	NbDF	social K-means			Soc-Inc-K-means		
				P	R	F	P	R	F
1	110	2950	775	0,050	0,087	0,064	0,107	0,043	0,061
2	120	3163	826	0,123	0,040	0,060	0,121	0,046	0,067
3	130	3356	873	0,130	0,064	0,086	0,161	0,056	0,083
4	140	3563	934	0,207	0,040	0,067	0,220	0,062	0,097
5	150	3773	1036	0,127	0,030	0,049	0,212	0,055	0,087
6	160	3955	1080	0,203	0,060	0,093	0,228	0,06	0,095
7	170	4202	1149	0,329	0,079	0,127	0,350	0,073	0,121
8	180	4415	1237	0,331	0,073	0,120	0,339	0,074	0,121
9	190	4591	1277	0,332	0,086	0,137	0,395	0,081	0,134
10	200	4852	1358	0,354	0,055	0,095	0,472	0,032	0,060

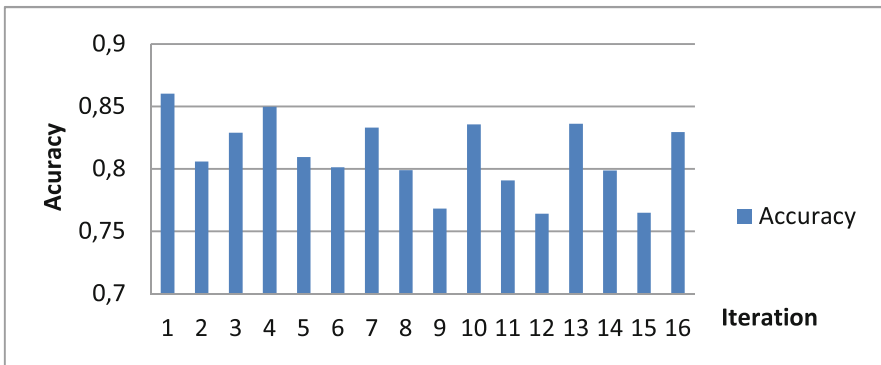
We have presented the difference between the evolution of social K-means and social Incremental K-means in Fig. 3.

To simulate the evolution of the social network, we have sorted the evaluations and users (by registration dates), from the least recent to the most recent. Then, we gradually injected this data into our system.



**Fig. 3.** Evolution of social K-means and incremental social K-means

We applied the Soc-K-Means for every 900 new users and col-K-means for every 20000 evaluations. The results presented in Fig. 4 were obtained with the following parameters for the different iterations (number of added users/new evaluations): col-K-NN = 6, social K-NN = 6, collaborative threshold = 20000, social threshold = 900, recommendation threshold value = 0.3. Between two applications of K-means, the K-NN algorithm is applied (Soc-K-NN is applied for each 300 new added users and Col-K-NN is applied for each 7000 evaluations). These results show the performance of the recommendation algorithm (accuracy value between 0.76 and 0.86) and confirm the contribution of the K-NN algorithm given that the system recommends friends between two applications of Incremental K-means.



**Fig. 4.** The accuracy of our system with the evolution of the social network

## 7 Conclusion

We presented in this article an enhanced collaborative filtering approach for the recommendation of friends in social networks. Our approach combines the CF recommendation with social information. Furthermore, in order to optimize the performance of the recommendation process, two classification techniques have been used, the incremental K-means and the K-NN algorithms. The results of the evaluations we carried out using the Yelp Dataset show the effectiveness of our system compared to the CF in terms of precision and F-measure. This combination alleviates the cold start problem as the system may suggest to a given user  $u$  a list of other appropriate users by using the social aspect. Moreover, the evaluation shows the contribution of using the incremental K-means and the added value of the K-NN algorithm (the obtained accuracy is approximately between 0.76 and 0.86). In our future work, we envisage to make further experiments using other datasets and to enrich our approach with semantic information in order to take into account the user's preferences and to benefit from the advantage of the semantic representation of the user's profile.

## References

1. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. *Knowl.-Based Syst.* **46**, 109–132 (2013)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng. (TKDE)* **17**, 734–749 (2005)
3. Burke, R.: Hybrid recommender systems: survey and experiments. *User Model. User Adapted Interact.* **12**(4), 331–370 (2002)
4. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) *The Adaptive Web*. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-72079-9\\_10](https://doi.org/10.1007/978-3-540-72079-9_10)
5. Liu, F., Lee, H.J.: Use of social network to enhance collaborative filtering performance. *Expert Syst. Appl.* **37**(7), 4772–4778 (2010)
6. Chang, C.-C., Chu, K.-H.: A recommender system combining social networks for tourist attractions. In: *Proceedings of the 5th International Conference on Computational Intelligence, Communication Systems and Networks*, Madrid, Spain, pp. 42–47 (2013)
7. Banati, H., Mehta, S., Bajaj, M.: Social behaviour based metrics to enhance collaborative filtering. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **6**, 217–226 (2014)
8. Su, J.-H., Chang, W.-Y., Tseng, V.S.: Effective social content-based collaborative filtering for music recommendation. *Intell. Data Anal.* **21**(1), 195–216 (2017)
9. Guo, G., Zhang, J., Yorke-Smith, N.: Leveraging multi-views of trust and similarity to enhance clustering-based recommender systems. *KBS* **74**, 14–27 (2015)
10. Najafabadi, M.K., Mahrin, M.N., Chuprat, S., Sarkan, H.M.: Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Comput. Hum. Behav.* **67**, 113–128 (2017)
11. Berkani, L.: SSCF: a semantic and social-based collaborative filtering approach. In: *Proceedings of the 12th ACS/IEEE International Conference AICCSA*, pp. 1–4 (2015)
12. Guellil, Z., Zaoui, L.: Proposition d'une solution au problème d'initialisation cas du K-means. Technical report (2012)