Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 54 (2015) 440 – 448

Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

A Particle Swarm Approach to Collaborative Filtering based Recommender Systems through Fuzzy Features

Mohammed Wasid and Vibhor Kant*

Department of Computer Science & Engineering, The LNM Institute of Information Technology, Jaipur 302 031, India

Abstract

Collaborative filtering (CF) either memory based or model based, has been emerged as an information filtering tool that provides effective recommendations to users utilizing the experiences and opinions of their similar neighbors when they interact with large information spaces. Memory based CF is more accurate than model based CF but it is less scalable. Our work in this paper is an attempt towards introducing a recommendation strategy (FPSO-CF) based on user hybrid features that retains the accuracy of memory – based CF as well as the scalability of model-based CF in an efficient manner. Since most user features are imprecise in nature, therefore these can be represented more naturally by using fuzzy sets. In this work, we employ particle swarm optimization algorithm (PSO) to learn user weights on various features and use fuzzy sets for representing user features efficiently. Effectiveness of our proposed RS (FPSO-CF) is demonstrated through experimental results in terms of various performance measures using the MovieLens dataset.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: Collaborative filtering; Fuzzy sets; Genetic algorithm; Particle swarm optimization; Recommender systems.

1. Introduction

In the age of information overload on the web, users employ many techniques to take decisions about how to utilize their free time, what to purchase, and even whom to date. Recommender system (RS) automate some of these techniques to provide effective recommendations to users while interacting with large information spaces[1]. RS recommends everything from movies, news, books, songs and Web sites to more complex suggestions for electronic gadgets, matrimonial matches, financial services, etc. In mid-90s, researchers started working on the development of recommendation algorithms by retaining a variety of filtering techniques namely collaborative filtering and content based filtering[2]. During the last decade, a lot of research has been carried out in the field of RS to design new algorithms for enhancing the recommendation accuracy.

Generally traditional RS generates suggestions to users through four filtering techniques content based filtering[3] (CBF), collaborative filtering[4] (CF), demographic filtering[5] (DMF) and hybrid filtering[6] techniques. CBF recommends items similar to those the user preferred in the past while DMF utilizes user attributes, classified as demographic data, for generating recommendations. Among these techniques, collaborative filtering (CF) has been established as

*Corresponding author. Tel.: +91 9899934190.
E-mail address: vibhor.kant@gmail.com.

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.051

the most successful and widely implemented technique in the area of RS. This interest produced a number of RS for various domains, such as Ringo[7] for music, the BellCore[8] Video Recommender for movies, and Jester for jokes. The basic idea of CF algorithm is to capture a user's preferences for building a user profile and then search for similar ones. These profiles are used to generate new suggestions to users. CF algorithm is broadly classified into memory-based and model-based systems[9]. Memory based systems are more accurate but they suffer from scalability problem whereas model based approaches are more scalable but less accurate. Al-Shamri and Kamal[10] developed a model that retains the accuracy of memory based CF and scalability of model based CF.

Although the incorporation of fuzziness in RS has been successfully introduced by many researchers, several directions are yet to be explored. Fuzzy logic is a multi-value logic that allows a better understanding of the result of a statement that is more approximate than accurate in real life. In contrast with "sharp logic", where results of a statement are binary ("true or false", "one or zero"), fuzzy logic admits a set of truth values in the interval [0, 1]. Based on fuzzy set theory introduced by Lotfi A Zadeh[11], a fuzzy set is determined by a membership function with a range of values between 0 and 1. Shahabi and Cyrus[6] proposed a Yoda RS that classified active user based on typical patterns of users and then made soft recommendations for her. Suryavanshi *et al.*[12] used relational fuzzy subtractive clustering while Nasraoui *et al.*[13] used fuzzy approximate reasoning to develop a general framework for the recommendation. Al-Shamri and Kamal[10] developed a fuzzified hybrid model in which a fuzzy distance measure is introduced for computing similarity between user profiles.

In real life every user places different priorities on various features. For example, some users give more prominence on particular features, while others do not show any interest in some features. Several efforts have been made in the past for incorporating various evolutionary approaches[14] into RS to learn optimal weights on many features[15]. Al-Shamri and Kamal[10] developed a hybrid fuzzy-genetic RS by employing genetic algorithm (GA) to evolve appropriate weights for each feature of the user. Similarly Ujjin and Peter[16] employed a genetic algorithm to learn personal preferences of users and further they extended it through Particle swarm optimization (PSO) in order to learn those preferences and the results were compared to those obtained from the GA recommender system. Finally they concluded that PSO worked significantly faster than GA.

In this work we develop a fuzzy-PSO CF (FPSO-CF) by employing particle swarm optimization algorithm to find optimal individual priorities to different features such as age, gender, rating etc. After finding suitable weights for different features, we compute effective similarities among users and generate appropriate recommendations to users.

The rest of this paper is organized as follows: Section 2 describes an overview of recent studies related to collaborative filtering and provides the basic introduction of PSO. In Section 3, we describe our proposed fuzzy-PSO CF and provide a detailed description of how the system employs the PSO for CF. Section 4 describes the data set, evaluation settings, evaluation metrics, and the results of the evaluation. Finally, Section 5 provides concluding remarks and suggest some future research direction.

2. Background

2.1 Collaborative filtering RS

Collaborative filtering (CF) exploits the preferences of users who have liked similar items in the past. The success of CF algorithms, however, is massively dependent on the technique designed to determine the set of similar users to the active user. Traditionally user similarity is evaluated by matching their preferences on a set of common items.

Formally, in CF we have a set of users $U = \{u_1, u_2, \ldots, u_p\}$ and a set of items $I = \{i_1, i_2, \ldots, i_q\}$ such as songs, books, news articles, or movies. Ratings are stored in a $p \times q$ user-item rating matrix. Explicit ratings from users follow a specified numerical scale indicating the degree of preferences (e.g. 1-bad to 5-excellent). Three major steps are needed to accomplish the recommendation task in CF.

- Collection of data for creation of a user profile
- Neighborhood set generation
- Predictions and recommendations

2.1.1 Collection of data for creation of a user profile

Usually, building a user model depends only on explicit user ratings. However, we can't say that two users are similar on the basis of their ratings on common items e.g., movies, but it also depends on other factors such as their background and personal details say age, gender, and user priorities to movie genres[16]. We collected the three types of data in our system namely demographic data (age, gender, and occupation), explicit rating about the items (movies) and implicit data about user behavior that is collected during registration. These movies are also represented in terms of genre in the used MovieLens dataset.

2.1.2 Neighborhood set generation

In this step, neighbors are simply a group of likeminded users of active user. The size of the neighborhood set could be fixed by choosing the top K users or could be flexible by choosing the users whose similarity value is above a certain threshold[7]. A variety of similarity methods have been researched, such as cosine similarity, the Pearson correlation[4], weight amplification, inverse user frequency and default rating[7], including probability-based approaches. Pearson correlation coefficient is the most popular method for memory-based CF which is defined by

$$sim(u, v) = \frac{\sum_{i \in C} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in C} (r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in C} (r_{v,i} - \bar{r}_v)^2}}, \tag{1}$$

where C is the set of items rated by both users u and v. Since formula (1), only captures the information of common items for both users, it is not suitable if other mentioned features are also included in the model. So, another way to compute similarity is the modified Euclidean distance function (2), which takes into account multiple features

$$d(U, V) = \frac{1}{x} \sum_{i=1}^{x} \sqrt{\sum_{j=1}^{m} (u_{i,j} - v_{i,j})^2}, \tag{2}$$

Here $u_{i,j}$ is the jth feature for the common item C_i, m is the number of features, and $x = |C|$ is the cardinality of C.

2.1.3 Predictions and recommendations

Once k neighbors are found, several approaches can be used to combine the ratings of neighbors to compute a prediction value on unseen items for the active user. After predicting how an active user will like specific items which have not been rated yet by the active user, the top-N item set, a set of ordered items with a higher predicted value, is identified and recommended. The predicted rating, $pre_{u,i}$, of item i of a user u is computed by the following formula[4]

$$pre_{u,i} = \bar{r}_u + k \sum_{u' \in C} d(u, u') \times (r_{u',i} - \bar{r}'_u), \tag{3}$$

where C denotes the set of neighbors who have rated item. The multiplier k is a normalizing factor and is usually selected as $k = 1/\sum_{u' \in C} |d(u, u')|$, and $\overline{r_{u'}}$ is the average rating of user u'.

2.2 Particle swarm optimization

Particle swarm optimization[17] (PSO) is a population based evolutionary technique like genetic algorithms. The only difference is that each particle or solution contains a position, velocity and acceleration. The velocity and acceleration change the position of the particle to explore the space of all possible solutions, instead of using crossover and mutation to generate new offspring. As particles move around the space, they sample onto different locations. Each location has a fitness value according to how good it is at satisfying the objective. Because of the rules governing the swarming process, particles will eventually swarm around the area in the space where fittest solutions are[18]. Every particle possesses its respective velocity and position and initial population of particle initialized randomly. The velocity and particle position is updated by following the two best values in the problem space. The first best value is the best solution (fitness) of the particle achieved so far which is called particle best denoted by "pbest". The other best value is the best value obtained so far by any particle. This best value is called global best denoted by "gbest".

Rating	Age	Gender	Occupation	18 genre frequencies
4	25	1	15	000010000000101000

Fig. 1. Profile(u, i) – typical user profile.

3. Fuzzy-PSO Approach to CF

In our proposed model there are three phases needed to accomplish the recommendation task by using the PSO.

- User profile formation
- Neighborhood set generation
- Predictions and recommendations

3.1 Phase I (user profile formation)

Generally a user profile contains rating information of experienced items. It can be extended by adding the demographic features for capturing the user behavior efficiently. Besides, an item is also represented by some features (movies by genre). Figure 1 depicts a user profile having rating on a particular item (movie), demographic feature (age, gender, occupation) and corresponding movie representation. In this representation users rating on a movie is an implicit measure of her likeness in various genre. To compute the interestingness measure of a genre G_j (GIM) for a user u_i we have used the following formula[10].

$$GIM(a, j) = \frac{2 \times N \times RGR(a, j) \times MRGF(a, j)}{RGR(a, j) + MRGF(a, j)}, \tag{4}$$

where MRGF is modified relative genre frequency of genre Gj for user u_a which is expressed as

$$MRGF(a, j) = \frac{\sum_{g \in G_j \subset C_i} \delta_3(r_{a,g}) + 2 \times \delta_4(r_{a,g}) + 3 \times \delta_4(r_{a,g})}{3 \times TF(a)}, \tag{5}$$

RGR (relative genre rating) is the ratio of u_i's ratings for high rated items of G_j to her total ratings, which is computed by formula (6)

$$RGR(a, j) = \frac{\sum_{g \in G_j \subset C_i \geq 3} r_{a,g}}{TR(a)}, \tag{6}$$

N is the normalization factor for a given system. TF and TR are the total frequency and total rating respectively. After computing GIM a user model will be created from demographic feature and corresponding GIM. But the crisp description of the age and genre interestingness measure does not reflect the actual case for human perceptions, because most human perceptions are fuzzy in nature therefore age is fuzzified into three fuzzy sets[19] young, middle-aged and old as shown in Fig. 2.

In similar manner, a GIM can be represented more naturally by linguistic variables using six fuzzy sets namely very bad (VB), bad (B), average (AV), good (G), very good (VG), and excellent (E) as shown in Fig. 3.

Based on the above discussion a user profile is represented more naturally in our proposed CF (FPSO-CF) for example that demographic features of a user are age is 45, a female user (0 used for female and 1 for male), occupation id is 23, and computed GIM for a particular genre is 0.510, then user profile can be represented by in the following manner as shown in Fig. 4.

3.2 Phase II (neighborhood set generation)

For performing CF it is required to compute similarity between users, as our profile is based on the fuzzy sets so we have used the following formula which is proposed by[10], the local fuzzy distance between u and v is defined as

$$Lfd(u_i, v_i) = d(u_i, v_i) \times d(u_i, v_i), \tag{7}$$

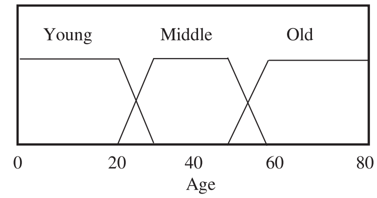

Fig. 2. Membership function for age feature.

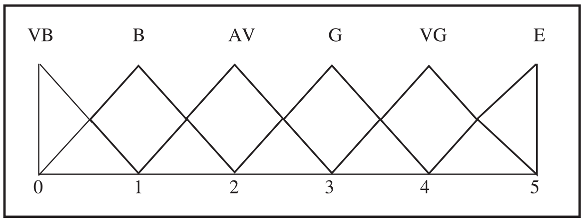

Fig. 3. Membership functions for genre interestingness measure.

0	1	0	0	23	.490	.510	0	0	0	0

Fig. 4. Representation of a user profile in our proposed CF.

where $d(u_i, v_i)$ is simply the difference operator, u and v are vectors of size m, and $d(u_i, v_i)$ is any vector distance metric. In all of our experiments, the Euclidean distance functions used for $d(u_i, v_i)$

$$d(u_i, v_i) = \sqrt{\sum_{i=1}^{m} (u_{i,j} - v_{i,j})^2}, \tag{8}$$

where m is the total number of fuzzy sets for the ith feature, and $u_{i,j}$ is the membership value of the ith feature in the jth fuzzy set. The global fuzzy distance is computed by the Euclidean distance formula (9)

$$Gfd(U, V) = \sqrt{\sum_{i=1}^{21} (Lfd(u_i, v_i))^2}, \tag{9}$$

In this formula features are equally weighted but it does not reflect the real life scenario where every user places different priorities on different features. The modified formula to compute the similarity between two users is as follows

$$Gfd(U, V) = \sqrt{\sum_{f=1}^{21} w_f \times (Lfd(x_f, y_f))^2}, \tag{10}$$

where w_f represents the weight for the fth feature.

We have used PSO to learn these weights in our proposed approach FPSO-CF. The necessary components of a PSO algorithm incorporated in our approach are as follows

3.2.1 Particle representation and initial population

A conventional PSO algorithm is used to learn the 21 feature weights, each feature weight is represented by 8-bit binary digits and the value of weight is ranging from 0 to 255. In PSO every particle is associated with its position and velocity therefore the particle is represented by 168 bit binary digits. To find the optimal weight of each feature after termination, we divide each weight by the total weight. Each weight can be obtained by converting binary digit into its corresponding decimal value. We have kept 10 as a population size containing 10 particles.

3.2.2 The fitness function

Determining an appropriate fitness function for a specific problem is a crucial task. In order to find the fitness score for each particle, the predicted ratings for each movie in the training set are computed. The average of the differences between the actual and predicted ratings of all movies in the training set is used as the fitness score for that set of weights

$$\text{fitness} = \frac{1}{t_R} \sum_{j=0}^{t_R} |r_j - pre_j|, \tag{11}$$

where t_R is the cardinality of the training set of an active user and pre_j is the predicted rating of item j of a user in the training set.

3.2.3 Particle dynamics

Particles move all over the search space by a simple set of equations. The algorithm updates the entire swarm at each time stamp by updating the velocity and position of each particle in every dimension by the following rules

$$\text{vel}_i = \omega \times \text{vel}_i + c_1 r_1 (p_{\text{pbest},i} - p_i) + c_2 r_2 (p_{\text{gbest}} - p_i), \tag{12}$$

$$\text{if}(|\text{vel}_i| > \upsilon_{\max}) \quad \text{vel}_i = (\upsilon_{\max}/|\text{vel}_i|)\text{vel}_i \tag{13}$$

$$p_i = p_i + \text{vel}_i \tag{14}$$

where p_i is the current position of particle i
p_{pbest} is the best position attained by particle i
p_{gbest} is the swarm's global best position
vel_i is the velocity of particle i
ω is a random inertia weight between 0.5 and 1
c_1 and c_2 are spring constants whose values are set to 1.494
r_1 and r_2 are random numbers between 0 and 1
$\upsilon_{\min}$ and $\upsilon_{\max}$ are – $(ub - lb)/2$ and $(ub - lb)/2$ respectively

3.2.4 Termination condition

We select predefined number of iterations as termination criteria. When the number of iterations reach the threshold value, algorithm terminates. Hence we get the optimal weights of every feature which leads to produce similar users to the active user.

After computing the similarity/distances among users effectively through PSO we select top 30 users for the neighborhood generation.

3.3 Phase III (predictions and recommendations)

We have predicted unseen items for every user with the help of formula (3).

4. Experiments and Results

To demonstrate the effectiveness of our proposed approach FPSO-CF we have conducted several experiments on the most popular MovieLens dataset.

Table 1. Parameter values for PSO algorithms.

Parameter name	Parameter	Description
Swarm size	10	The number of particles in the swarm at each generation.
Maximum number of iteration for each run	30	If the number of iterations reaches this value and the solution has not been found, the best solution for that iteration is used as the final result.
Number of runs	10	The number of times the system was run for each active user.

4.1 Design of experiments

We have selected only those users who have rated at least 60 movies from MovieLens dataset. Out of 943 users only 497 users rated at least 60 movies and contributed 84,596 ratings out of 100,000. We have used a 5 fold cross validation approach to reduce the biasness of the system, these folds will be referred to as fold-1, fold-2, ..., fold-5. For each fold, only 50 users are selected randomly as active user and remaining 447 users are used to form neighborhood set for active user. Each active user's ratings are divided randomly into training set (66%) and test set (34%). The ratings of training set is used for neighborhood generation whereas the ratings in the test set are considered as unseen items by the active user. We have conducted several experiment to demonstrate the relative performances of the following scheme.

- Pearson CF (PCF)
- Fuzzy CF (FCF)[10]
- Fuzzy genetic CF (FG-CF)[10]
- Proposed particle swarm approach to CF (FPSO-CF)

4.2 Performance measures

We measure system accuracy using the MAE, coverage of the system and number of correct predictions. The MAE measures the deviation of predictions generated by the proposed scheme from the true ratings specified by the user. The MAE for an active user u_i is given by following formula

$$MAE(i) = \frac{1}{t_i}\sum_{j=1}^{t_i} |pre_{i,j} - r_{i,j}|, \tag{15}$$

Coverage is defined as the percentage of items over all users for which a prediction was requested and the system was able to produce a prediction. Low coverage value indicates that the RS will not be able to avail the user with sufficient amount of the new items. The coverage is computed by the following equation

$$\text{Coverage} = \frac{\sum_{i=1}^{T_n} q_i}{\sum_{i=1}^{T_n} t_i} \tag{16}$$

where q_i is the total number of predicted items and t_i is the cardinality of the test ratings set of user u_i.

Table 1 shows the appropriate parameters for employing PSO in our approach.

4.3 Results

To demonstrate the ability of the proposed approach FPSO-CF to offer better recommendation quality as compared to PCF, FCF, and FG-CF, we analyzed the results for the MAE, coverage and number of correct predictions. Figure 5 and 6 show the correct prediction percentages obtained from these approaches for 50 users of the best fold and worst fold respectively. Results presented in Table 2 gives the relative performances of these schemes for each fold.

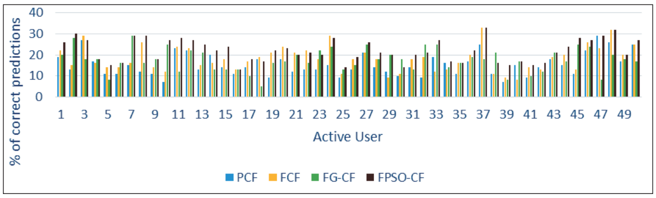

Fig. 5. Correct predictions percentage for active users of fold-2 (best fold).

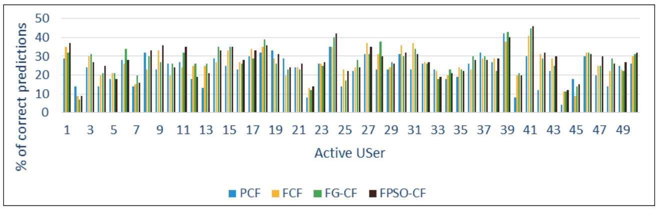

Fig. 6. Correct predictions percentage for active users of fold-4 (worst fold).

Table 2. Comparison of FPSO-CF with PCF, FCF and FG-CF.

	FPSO-CF with PCF			FPSO-CF with FCF			FPSO-CF with FG-CF		
Fold	Greater	Equal	Smaller	Greater	Equal	Smaller	Greater	Equal	Smaller
1	38	3	9	33	8	9	28	11	11
2	45	2	3	34	8	8	35	10	5
3	39	4	7	34	5	11	30	4	16
4	40	2	8	31	5	14	28	1	21
5	44	1	5	34	4	12	37	2	11

Table 3. Total MAE and coverage.

	MAE				Coverage			
Fold	PCF	FCF	FG-CF	FPSO-CF	PCF	FCF	FG-CF	FPSO-CF
1	0.8642	0.8196	0.8095	0.8033	0.8921	0.9556	0.9561	0.9654
2	0.8974	0.8310	0.8252	0.7954	0.8367	0.9490	0.9522	0.9546
3	0.8924	0.8157	0.7964	0.7892	0.8181	0.9351	0.9370	0.9564
4	0.8861	0.8566	0.8382	0.8321	0.8534	0.9383	0.9421	0.9424
5	0.8564	0.8082	0.7883	0.7833	0.8556	0.9478	0.9499	0.9579

Correct predictions generated by FPSO-CF are better than that of PCF, FCF, and FG-CF on 94%, 84%, and 90% cases respectively for the fold 2, whereas these are 84%, 72%, and 58% for the worst fold 4 respectively as depicted in Table 2. Table 3 shows the relative MAE and coverage for 50 users of each fold evaluated by these approaches. The MAE of FPSO-CF is superior to other approaches because of lower MAE similarly coverage of the proposed approach is also higher than other remaining approaches.

5. Conclusions and Future Work

We have proposed a particle swarm optimization approach to collaborative filtering based on fuzzy features (FPSO-CF). Focusing on the accuracy of CF, some major challenges such as imprecision in user features and computational time have been addressed in our proposed approach. To deal with imprecise nature of user features, we have designed fuzzy sets for representing user features efficiently. We proposed the use of the particle swarm optimization (PSO) in our approach to find the optimal priorities for individual features of different users and thereby generating more personalized and accurate recommendations to them. A comparison of experimental results against those obtained using correlation-based CF (PCF), fuzzy CF (FCF) and hybrid fuzzy-genetic CF (FG-CF) clearly indicates superiority of the proposed approach (FPSO-CF) in terms of MAE, coverage and correct predictions. Further, experimental results also establish that our proposed approach (FPSO-CF) achieved the final solution significantly faster in comparison to PCF, FCF and hybrid fuzzy-genetic CF (FG-CF).

Since contexts play a major role in the field of RS, it is to be seen how the notion of contexts can be incorporated in the proposed approach for further improvement.

References

[1] V. Kant and KK. Bharadwaj, Fuzzy Computational Models of Trust and Distrust for Enhanced Recommendations, *International Journal of Intelligent Systems*, vol. 28, pp. 332–365, (2013).

[2] G. Adomavicius and A. Tuzhilin, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Trans. on Knowledge and Data Eng.*, vol. 17, pp. 734–749, (2005).

[3] K. Lang, Newsweeder: Learning to Filter Netnews, In *Proc. of the 12th International Conference on Machine Learning*, pp. 331–339, (1995).

[4] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *Proc. 1994 ACM Conference on Computer Supported Cooperative Work*, New York: ACM; pp. 175–186, (1994).

[5] B. Krulwich, Lifestyle Finder: Intelligent user Profiling using Large-Scale Demographic Data, *AI Magazine*, vol. 18, pp. 37, (1997).

[6] C. Shahabi, F. Banaei-Kashani, YS. Chen and D. McLeod, Yoda: An Accurate and Scalable Web-Based Recommendation System, In *Cooperative Information Systems*, Springer Berlin Heidelberg, pp. 418–432, (2001).

[7] U. Shardanand and P.Maes, Social Information Filtering: Algorithms for Automating Word of Mouth, In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems, ACM*, Press/Addison-Wesley Publishing Co, pp. 210–217, (1995).

[8] W. Hill, L. Stead, M. Rosenstein and G. Furnas, Recommending and Evaluating Choices in a Virtual Community of Use, In *Proc. of the SIGCHI conference on Human Factors in Computing Systems, ACM*, Press/Addison-Wesley Publishing Co; pp. 194–201, (1995).

[9] JS. Breese, D. Heckerman and C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, In *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc, pp. 43–52, (1998).

[10] MYH. Al-Shamri and KK. Bharadwaj, Fuzzy-Genetic Approach to Recommender Systems Based on a Novel Hybrid User Model, *Expert Systems with Applications*, vol. 35, pp. 1386–1399, (2008).

[11] LA. Zadeh, Fuzzy Sets, *Information and Control*, vol. 8, pp. 338–353, (1965).

[12] B. Suryavanshi, N. Shiri and S. Mudur, A Fuzzy Hybrid Collaborative Filtering Technique for Web Personalization, In *Proc. of 3rd International Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pp. 1–8, (2005).

[13] O. Nasraoui and C. Petenes, An Intelligent Web Recommendation Engine Based on Fuzzy Approximate Reasoning, In *Proc. of the IEEE International Conference on Fuzzy Systems*, pp. 1116–1121, (2003).

[14] V. Kant and KK. Bharadwaj, Incorporating Fuzzy Trust in Collaborative Filtering Based Recommender Systems, In *Proc. of the 2nd International Conference of SEMCCO*, Springer Berlin Heidelberg, pp. 433–440, (2011).

[15] Tyagi Shweta and KK. Bharadwaj, Enhancing Collaborative Filtering Recommendations by Utilizing Multi-Objective Particle Swarm Optimization Embedded Association Rule Mining, *Swarm and Evolutionary Computation*, vol. 13, pp. 1–12, (2013).

[16] S. Ujjin and P. J. Bentley, Learning User Preferences Using Evolution, In *Proc. of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, Singapore, (2002).

[17] RC. Eberhart and J. Kennedy, A New Optimizer Using Particle Swarm Theory, In *Proc. of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, (1995).

[18] S. Ujjin and PJ. Bentley, Particle Swarm Optimization Recommender System, In *Swarm Intelligence Symposium, 2003. SIS'03. Proc. of the 2003 IEEE*, pp. 124–131, (2003).

[19] G. Klir and B. Yuan, Fuzzy Sets and Fuzzy Logic, New Jersey: Prentice Hall, (1995).