Hybrid Personalized Recommender System Using Fast K-medoids Clustering Algorithm

Subhash K. Shinde
Research Scholar, SRTMU, Nanded, India
skshinde@rediffmail.com

Uday V. Kulkarni
Professor, Computer Engineering, SGGSIET, Nanded, India
kulkarniuv@yahoo.com

***Abstract*— Recommender systems attempt to predict items in which a user might be interested, given some information about the user's and items' profiles. This paper proposes a fast k-medoids clustering algorithm which is used for Hybrid Personalized Recommender System (FKMHPRS). The proposed system works in two phases. In the first phase, opinions from the users are collected in the form of user-item rating matrix. They are clustered offline using fast k-medoids into predetermined number clusters and stored in a database for future recommendation. In the second phase, clusters are used as the neighborhoods, the prediction rating for the active users on items are computed by either weighted sum or simple weighted average. This helps to get more effective and quality recommendations for the active users. The experimental results using Iris dataset show that the proposed fast k-medoids performs better than k-medoids and k-mean algorithms. The performance of FKMHPRS is evaluated using Jester database available on website of California University, Berkeley and compared with web personalized recommender system (WPRS). The results obtained empirically demonstrate that the proposed FKMHPRS performs superiorly.**

***Index Terms*— Web Personalized Recommender System, Fast k-medoids, k-medoids.**

I. Introduction

A Recommender System arises from the need to be able to provide users with relevant and personalized information. They help the user make choices when there is not sufficient personal experience regarding the available options. These systems can assist the consumer in various ways. They can simplify the information search process and facilitate the comparison of products, report the reviews of other users, or exploit the consumer's history to suggest products similar to those purchased in the past or previously selected by users with a similar buying behaviour [1]. Nowadays, the advance of Internet and Web technologies has continuously boosted the prosperity of e-commerce. Through the Internet, different merchants and customers can now easily interact with each other, and then have their transactions within a specified time. However, the Internet infrastructure is not the only decisive factor to guarantee a successful business in the electronic market. With the continuous development of electronic commerce, it is not easy for customers to select merchants and find the most suitable products when they are confronted with the massive product information in Internet. In the whole shopping process, customers still spend much time to visit a flooding of retail shops on Web sites, and gather valuable information by themselves. This process is unnecessary time-consuming Even sometimes the contents of Web document that customers browse have nothing to do with that of they need indeed. This inevitably influences customers' confidence and interests for shopping on Internet. In order to provide decision support for customers, one way to overcome the above problem is to develop intelligent recommendation systems to provide personalized information services. A recommendation system is a valid mechanism to solve the problem of information overload in Internet shopping. On the shopping websites, the system can help customers find the most suitable products that they would like to buy by providing a list of recommended products. For those products that customers buy frequently, such as grocery, books and clothes, the system can be developed to reason the customers' personal preferences by analyzing their personal information and shopping records. This produces the sensible recommendations for them. Therefore, it is of important to develop the high efficient learning algorithm to capture what customers need to help them buy. The WPRS framework intended for hybrid recommender system which combines collaborative recommendation with other types of recommendation components [2]. It alleviates the challenges such as data sparsity and scalability.

The remainder of this paper is organized as follows. The section II summarizes the different strategies for recommender systems and their drawbacks. The proposed clustering based hybrid personalized recommender system is described in the section III. The section IV illustrates experimental setup of the proposed recommendation system. This section also gives performance evaluation with the existing algorithms. Finally, the section V concludes the paper.

Manuscript received June 03, 2011; revised June 30, 2011; accepted July 8, 2011. Corresponding author: Subhash K. Shinde.

doi:10.4304/jait.2.3.152-158

II. Recommender System Strategies

In the recent years web personalization has undergone through tremendous changes. The content [3, 4], collaborative [5, 6] and hybrid [7] based filtering are three basic approaches used to design recommendation systems.

The content based filtering [8] relies on the content of an item that user has experienced before. The content based information filtering has proven to be effective in locating text, items that are relevant to the topic using techniques such as Boolean queries, vector space queries etc. However, content based filtering has some limitations. It is difficult to provide appropriate recommendation because all the information is selected and recommended based on the content. Moreover, the content based filtering leads to overspecialization i.e. it recommends all the related items instead of the particular item liked by the user.

The collaborative-filtering [9] aims to identify users who have relevant interests and preferences by calculating similarities and dissimilarities between their profiles. The idea behind this method is that to one’s search the information collected by consulting the behavior of other users who shares similar interests and whose opinions can be trusted may be beneficial. The different techniques have been proposed for collaborative recommendation; such as correlation based method, semantic indexing etc. The collaborative filtering overcomes some of the limitations of the content based filtering. The system can suggest items to the user, based on the rating of items, instead of the content of the items which can improve the quality of recommendations. However, collaborative filtering has some drawbacks. The first drawback is that the coverage of rating could be very sparse thereby resulting in poor quality recommendation. In the case of the addition of new items into database, the system would not be able to recommend until that item is served to a substantial number of users known as cold-start. Secondly, when new users are added, the system must learn the user preferences from the rating of users, in order to make accurate recommendations. Moreover, these recommendation algorithms seem to be very extensive and grow non-linearly when the number of users and items in a database increase. The hybrid recommendation systems [10, 11] combine content and collaborative based filtering to overcome these limitations. As stated below, there are different ways of combining content and collaborative based filtering [12].

I. Implementing these approaches separately and combining them for prediction.
II. Incorporating some content based characteristics into collaborative approach and vice versa.
III. Constructing a general unified model that incorporates both content and collaborative based characteristics.

The hybrid approach proposed in this paper extracts user’s current browsing patterns using web usage mining, and forms a cluster of items with similar psychology to obtain implicit users rating for the recommended item.

III. Proposed FKMHPRS

We have developed and tested the FKMHPRS for Jester dataset available on website of California University, Berkeley. The system architecture has been partitioned into two main phases; offline and online. The Fig. 1 depicts the architecture of FKMHPRS with its essential components.

The phase I is offline. It preprocesses and forms clusters. In this phase background data in the form of user-item rating matrix is collected and clustered using the proposed approach which is described in section A. Once the clusters are obtained the cluster data along with their centroids is stored for future recommendations. The phase II is online where the recommendation takes place for the active user. Here, similarity between active users and clusters is calculated to choose the best clusters for making recommendations. The rating quality of each item unrated by active user is computed in the chosen clusters. To generate the recommendations, clusters are further selected based on rating quality of an item. The recommendations are then made by computing the weighted average of the rating of items in the selected clusters. The working of FKMHPRS is described below in detail with the Jester dataset.

A. Preprocessing phase

The Jester dataset is available online on the site www.ieor.berkeley.edu/~goldberg/jester-data. The Jester is a WWW Based Joke Recommender System, developed by University of California, Berkeley. This data has 73421 user entered numeric rating for 100 jokes, ranging on real value scale from -10 to 10. User-item rating taken from Jester dataset rated in the scale of -10 to +10 is normalized in the scale of 0 to 1, where 0 indicates that item is not rated by corresponding user. To facilitate the discussion, running example shown in the Table I is used, where U_1-U_{10} are the users and J_1-J_{10} are the items (jokes) rated or unrated by users. The last row of Table 2 gives ratings of the active user (U^1).

B. Proposed fast K-medoids clustering

K-medoids is a clustering algorithm that is related to the k-means algorithm [13]. The k-medoids is a partitioning algorithm that divides the data set up into separate clusters. The algorithm attempts to minimize the squared error, which is the distance between points in the cluster and a point that is designated as the center (medoid) of the cluster. A medoid is considered as an object of a cluster whose average dissimilarity to all the objects in a cluster is minimal [14].

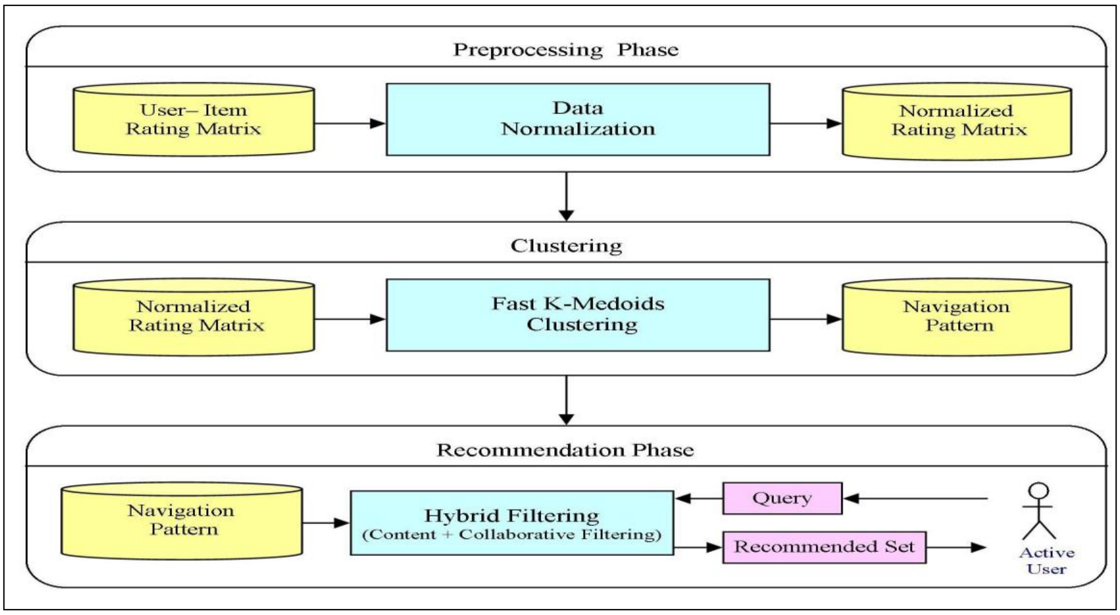

Figure 1: The architecture of FKMHPRS

TABLE I
RUNNING EXAMPLE OF RATING MATRIX FROM JESTER DATA SET AFTER NORMALIZATION IN THE RANGE OF 0 TO 1

Users	**J$_1$**	**J$_2$**	**J$_3$**	**J$_4$**	**J$_5$**	**J$_6$**	**J$_7$**	**J$_8$**	**J$_9$**	**J$_{10}$**
U$_1$	0.15	0.94	0.06	0.13	0.16	0.11	0.05	0.72	0.09	0.29
U$_2$	0.71	0.51	0.82	0.73	0.41	0.06	0.48	0.26	0.94	0.96
U$_3$	0.00	0.00	0.00	0.00	0.95	0.96	0.95	0.96	0.00	0.00
U$_4$	0.00	0.92	0.00	0.00	0.60	0.91	0.38	0.81	0.00	0.61
U$_5$	0.92	0.74	0.32	0.26	0.58	0.60	0.85	0.74	0.50	0.79
U$_6$	0.23	0.35	0.54	0.11	0.18	0.31	0.11	0.48	0.20	0.43
U$_7$	0.00	0.00	0.00	0.00	0.93	0.05	0.89	0.94	0.00	0.00
U$_8$	0.84	0.67	0.96	0.22	0.13	0.44	0.96	0.59	0.27	0.31
U$_9$	0.34	0.35	0.07	0.19	0.10	0.51	0.27	0.09	0.14	0.44
U$_{10}$	0.66	0.76	0.76	0.66	0.82	0.76	0.94	0.64	0.66	0.91
U^1	0.38	0.71	0.00	0.00	0.20	0.00	0.64	0.27	0.00	0.59

However K-medoids has some limitations. K-medoids clustering algorithm centroids are initially selected by the user. Therefore, performance of these algorithms depends on this manual selection of centroids. It works inefficiently for large data sets due to its complexity [15].

This is the major motivation behind the work presented in this paper. The proposed clustering algorithm initially calculates centroids appropriately; this results in the proper creation of the clusters

Let p patterns having n variable are to be grouped into $k(k\langle p)$ clusters, where k is assumed to be given and let j th variable of object i as $R_{ij} = (i = 1,\cdots,n;\ j = 1,\cdots,p)$.

The Euclidean distances will be used as a dissimilarity measure. The distances between i and j is given as

$$d_{ij} = \sqrt{\sum_{a=1}^{p} \left(R_{ia} - R_{ja}\right)^2}$$

where $i = 1,\cdots,n;\ j = 1,\cdots,n.$

The proposed algorithm is composed of the following steps.

Step 1: Initial selection of medoids:

To determine the medoids of the cluster, all the patterns are applied to each of the pattern and the patterns with Euclidian distance less than or equal to α, a user defined value, are counted for all the patterns, as described by following equation 1.

$$\text{If} \left(\left| \boldsymbol{R}_i \mathop{-}_{j=1}^{P} \boldsymbol{R}_j \right| \le \alpha \right) \quad \text{then} \quad D_i = D_i + 1 \qquad (1)$$

for $i = 1, 2, \cdots, p.$

The top *k* patterns given maximum count are selected as initial medoids.

The pattern with the maximum count is selected as the medoids of the cluster as illustrated in equation 2.

$$[D_{max} D_{ind}] = max[D] \qquad (2)$$

Where, D_{max}, is the maximum value in the row vector *D*, and, D_{ind}, is the index of maximum value. Therefore the most appropriate pattern, $C_1 = R_{ind}$ is chosen as the initial medoid.

Step 2: Finding Nearest Neighbour

I. If the distances between given pattern and the medoid are greater than β, user defined parameter, the pattern is assigned to outlier, otherwise kept near to the nearest medoids. The process of Nearest Neighbour is as follows:

Let d_{ij} is distance between pattern R_i and medoid C_j, where $i = 1,2\cdots,p$; $j = 1,2\cdots,m$ and $1 \subseteq m \subseteq k$.

If [($d_{ij} \langle \beta$) and ($min\cdot d_{ij} = d_{im}$)]

Then R_i is nearest neighbour of Cm

else R_i is outlier.

II. Calculate the sum of distances from all patterns to their mediods as described by equation 3.

$$\mathrm{sum}(p,c) = \sum_{i=1}^{d} |p-c| \qquad (3)$$

Step 3: Update mediods

I. To update mediods, select the pattern which count is less than previous mediods. Assign each object to nearest mediods using step 2 and obtain the cluster result.

II. Calculate the sum of distances from all patterns to their mediods using equation 3.

Step 4: If sum is garter than equal to the previous one then stop the algorithm, otherwise goes Step 2.

The proposed fast K-mediods is used for clustering of the Jester data set as shown in Table 1. The clustering is resulted in the three clusters $(k = 3)$ with $\alpha = 0.9$ and $\beta = 0.9$. The details of the clusters are created and users in each cluster are shown in the Table II. After clustering as stated in the fast K-mediods, knowing the members of each group, we have recomputed new centroids as the average of all corresponding coordinates of the members of each cluster.

TABLE II.
USERS IN EACH CLUSTER USING FAST K-MEDIODS

Cluster No.	Users	Centroid
1	U_4, U_6, U_8, U_9	C_1
2	U_1, U_2, U_5, U_{10}	C_2
3	U_3, U_7	C_3

C. Recommendation Phase

This phase consists of two steps: (I) Similarity Computation and (II) Generating Recommendation

.

I. Similarity Computation:

In order to find the nearest neighbors of the active user, it must measure the similarity of the users. Calculate similarity between clusters' centroids and active users. Select cluster that have the highest similarity. We use Pearson correlation similarity algorithm to measure the similarity between active user (u^1) and clusters (c_u). User rating can be treated as a vector on an n-dimensional item space. The similarity between u^1 and c_u is computed as follows

$$sim(u^1, c_u) = \frac{\sum_{i \in I} \left(R_{u^1,i} - \overline{R_{u^1}}\right)\left(R_{c_u,i} - \overline{R_{c_u}}\right)}{\sqrt{\sum_{i \in I} \left(R_{u^1,i} - \overline{R_{u^1}}\right)^2} \sqrt{\sum_{i \in I} \left(R_{u^1,i} - \overline{R_{u^1}}\right)^2}} \qquad (4)$$

For running example, the similarity value of active user of three clusters is shown in Table III. Choose the clusters with the highest similarity value.

TABLE III.
SIMILARITY COMPUTATION BETWEEN u^1 AND c_u

	Cluster 1	Cluster 2	Cluster 3
$sim(u^1, c_u)$	0.32	0.43	0.31

II. Generating Recommendation

When a cluster of nearest neighbors (neighborhood) of the active user is chosen, predictions are generated based on a weighted aggregate of their ratings. Most used aggregating functions are *similarity sum or simple similarity average*. The predication rating of item *i* by u^1 is $P_{u^1}(i)$ which is gained by the rating of nearest neighbors set c_u rated by active user u^1, the computation method is as the follows:

$$P_{u^1}(i) = \overline{R_{u^1}} + \frac{\sum_{u^1 \in c_u} sim(u^1, c_u) * \left(\overline{R}_u(i) - R_{u^1}(i)\right)}{\sum_{u^1, c_u} \left(\left\| sim(u^1, c_u) \right\|\right)} \quad (5)$$

where $\overline{R_{u^1}}$ and $\overline{R}_u(i)$ are average rating of items rated by active user u^1 and average rating of item i rated by all user. $sim(u^1, c_u)$ is similarity between active user and users clusters.

TABLE IV.
PREDICATION RATING FOR ACTIVE USER

	Items(Jokes)	Predicting Rating
Active User (u^1)	J_3	0.63
	J_4	0.59
	J_6	0.53
	J_9	0.43

The predication rating of active user for running example is shown in Table IV. Once the predicting rating of each item is calculated, the recommendation to the active user is provided, e.g., joke 3 predication rating up to 0.63 will be recommended and so on.

IV. Experiments

We have conducted a set of experiments to examine the effectiveness of our proposed recommender system in terms of accuracy of neighbor-selection and recommendation quality.

The proposed FKMHPRS is implemented in MATLAB version 7.2. The experiments are conducted on a 2.0 GHz, Intel Pentium-IV PC with 512 MB memory, running Microsoft Windows XP Professional.

A. Performance evaluation of clustering

In order to check the performance of the proposed clustering algorithm, we first applied the algorithm to real data set, 'Iris' data, whose true classes are known. The Iris data set is available in UCI repository (ftp://www.ics.uci.edu/ pub/machinelearningdatabases/), which includes 150 objects (50 in each of three classes – 'Setosa', 'Versicolor', and 'Virginica') with four variables ('sepal length', 'sepal width', 'petal length', and 'petal width').

The performance was measured by the accuracy, which is the proportion of objects that are correctly grouped together against the true classes. To investigate the performance more objectively, a simulation study was carried out by generating artificial data sets repetitively and calculating the average performance of the method.

We have applied the proposed fast K-medoids, K-medoid, K-means [16], and fuzzy c-means clustering (FCM) [17] to create three clusters using this data without the class information. The table V shows the results obtained using existing and proposed clustering method.

TABLE V.
CLUSTER RESULTS OF IRIS DATA BY THE PROPOSED AND TRADITIONAL METHODS

Algorithms	Setosa	Versicolor	Virginica	Computational Time
K-means	50	39	61	13.9790 seconds
K-medoids	50	40	60	13.9330 seconds
FCM	50	40	59	13.5790 seconds
Proposed Fast K-medoids	50	43	57	11.0390 seconds

The table V shows that the proposed fast K-Medoids clustering algorithm works superior than the traditional algorithms because the algorithm calculates centroids properly instead of selecting randomly.

B. Performance evaluation of recommender system

The experiments are performed on the small Jester dataset consisting of user-item rating matrix of size 10 (users) × 10 (jokes) as shown in the Table 1. The measurement method of evaluating the recommendation quality of recommendation system mainly includes statistical precision measurement method and decision supporting precision measurement method [18, 19]. Statistical precision measurement method adopts MAE (Mean Absolute Error) to measure the recommendation quality [20]. MAE is a commonly used recommendation quality measurement method. So we use MAE as the measurement criteria. MAE calculates the irrelevance between the recommendation value predicted by the system and the actual evaluation value rated by the user. We represent each pair of interest predicted rank as <p_i, q_i>, p_i is the system predicted value, q_i is the user evaluation value. Basing on the entire <p_i, q_i> pairs, MAE calculates the absolute error value $|p_i$-$q_i|$ and the sum of all the absolute error value, and then calculates their average value. If the MAE value is small, it indicates good recommendation quality.

The predicted user rating set can be represented as $\{p_1, p_2, \cdots, p_N\}$, its corresponding actual user rating set can be represented as $\{q_1, q_2, \cdots, q_N\}$, the MAE can be defined as the following[21]:

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \quad (6)$$

The performance of proposed FHMHPRS is compared with WPRS [10]. The WPRS proposed an overall framework for hybrid recommender system which combines collaborative recommendation with other types of recommendation components. Let us examine the influence of various nearest neighbor set on predictive validity. We gradually increase the number of neighbors; the experiment results are shown in Table VI:

Table VI.
INFLUENCE OF VARIOUS SIZE OF NEAREST NEIGHBOR SET ON PREDICTIVE VALIDITY

Size of Neighbor Set	*MAE*	
	WPRS	FHMHPRS
04	1.3122	1.0213
08	1.3651	1.0438
12	1.4663	1.2332
16	1.4893	1.4332
20	1.5173	1.4733

As Fig. 2 shows, the FKMHPRS has smaller MAE value than WPRS in most cases, which means that the sparseness has the less impact on our proposed algorithm.

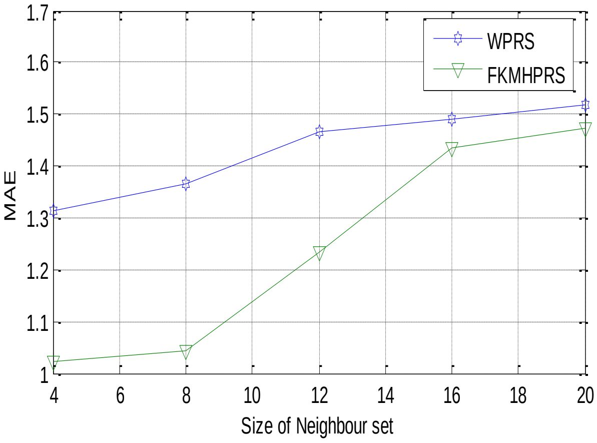

Figure: 2 MAE on each algorithm. (A small value means a better performance)

IV. CONCULSIONS

This paper describes a novel hybrid personalized recommender system that utilizes clustering of user-item rating matrix through proposed fast K-medoids and provides the recommendations for the active user with good quality rating using similarity measures. The results from various simulations using Iris data set shows that the proposed fast K-medoids clustering algorithm performs better than K-medoid, K-means, and FCM which helps to improve the quality of rating. Through the experiment analysis, it is found that the proposed FKMHPRS performs better than WPRS and the sparseness has less impact on the proposed system.

ACKNOWLEDGMENT

We would like to thank management of Bharati Vidyapeeth College of Engineering, Navi Mumbai, India and the director, SGGS Institute of Engineering and Technology, Nanded, India, for providing the infrastructure to carry out the proposed research work

REFERENCES

[1] Ricci and F. Del Missier, “Supporting Travel Decision making Through Personalized Recommendation,” *Design Personalized User Experience for e-commerce*, pp. 221-251, 2004.

[2] S. K. Shinde and U. V. Kulkarni, “The hybrid web personalized recommendation based on web usage mining,” *International Journal of Data Mining, Modeling and Management*, vol. 2, no. 4, pp.315-333, 2010.

[3] R. B. Allen, “User models: theory, method and Practice,” *International Journal of Man–Machine Studies*, vol. 43. no.11, pp. 27-52, 1990.

[4] D. Kalles, A. Papagelis and C. Zaliagis, “Algorithmic aspects of web intelligent systems,” *Web Intelligence*, Springer, Berlin, pp. 323–345, 2003.

[5] J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl, “Evaluating collaborative filtering recommender systems”, *ACM Trans. on Information Systems (TOIS)*, vol. 22, no.1, pp. 5-53, 2005.

[6] T. Hofmann, “Collaborative filtering via Gaussian probabilistic latent semantic analysis,” *Proceedings of the 26th Annual Int’l ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 145-153, 2003.

[7] M. Balabanovic and Y. Sholam, “Combining content-based and collaborative recommendation”, *Comm. ACM*, vol. 40, no.3, pp. 23-43, 1997.

[8] Chun Zeng et al. “Personalized Services for Digital Library,” *In Proceeding of 5th International Conference on Asian Digital Libraries*, pp. 252-253, 2002.

[9] G. Ulrike and F. R. Daniel, “Persuasion in recommender systems”, *Int’l Journal of Electronic Commerce*, vol. 11, no. 2, pp. 81-100, 2006.

[10] S. K. Shinde and U. V. Kulkarni, “A New Approach for on Line Recommender System in Web Usage Mining,” *Proceeding Inte’l Conference on Advanced Computer Theory and Engineering*, pp. 312-317, 2008.

[11] K. Yu, A. Schwaighofer and H. P. Kriegel, “Probabilistic Memory-Based Collaborative Filtering,” *IEEE Trans.*

Knowledge and Data Engineering, vol. 16, no.1, pp. 56-69, 2004.

[12] K.W. Cheung and Ch. Tsui, "Extended latent class models for collaborative recommendation," *IEEE Trans. on Systems, Man and Cybernetics-Part A: Systems and Humans,* Vol. 34, No.1, pp. 143-148, 2004.

[13] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," Wiley, New York, December 1990.

[14] A. P. Reynolds, G. Richards and V. J. Rayward, "The application of k-medoids and PAM to the clustering of rules," *Intelligent Data Engineering and Automated Learning,* Lecture Notes in Computer Science, Springer, pp. 173–178, 2004.

[15] J. Han, M. Kamber and A. Tun, "Spatial clustering methods in data mining: A survey," In H. J. Miller & J. Han (Eds.), *Geographic data mining and knowledge discovery.* Taylor and Francis, 2001.

[16] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, 1967.

[17] J. Bezdek. "Pattern Recognition with Fuzzy Objective Function Algorithms". Plenum Press, USA, 1981.

[18] D. Billsus and M. "Learning collaborative information filter." *Proceeding 5th International conference on Machine Learning,* pp. 46-54, 1998.

[19] C. H. Basu et al., "Recommendation as classification: Using social and content–based information in recommendation," *Proceeding 15th International conference on Artificial Intelligence*, pp. 714-720, 1998.

[20] Huihong Zhou, Yijun Liu, Weiqing Zhang, and Junyuan Xie, "A Survey of Recommender System Applied in E-commerce" *Computer Application Research*, vol. 1, no. 1, pp. 8-12, 2004.

[21] Z. Zhao and S. Bing, "An Adaptive Algorithm for Personal Recommendation," *Journal of Changchun University,* vol. 1, no. 1, pp. 22-29, 2005.

Subhash K. Shinde has submitted his Ph.D. (Computer Engineering) in March-2011 at Swami Ramanand Teertha Marathwada University, India. Currently he is working in the field of Web Mining, Frequent Pattern Discovery and Integration of domain knowledge such as ontologies in the mining process and benefits of patterns in web personalized recommendations for e-commerce. He received B. E. (Computer Engineering) in 1999 and M. E. (Information Technology) from Mumbai University, India in 2006.

Uday V. Kulkarni is working as a Professor in the Department of Computer Science and Engineering at SGGS Institute of Engineering and Technology, India. He received doctoral degree from Swami Ramanand Teertha Marathwada University, India in 2002. He is a recipient of a national level gold medal in the Computer Engineering Division for his research paper 'Fuzzy Hypersphere Neural Network Classifier' published in the journal of Institution of Engineers in 2004. He has published more than forty research papers in the field of Neural Networks, Fuzzy Logic and hybrid computing systems in the reputed journals and conferences.