Enhancing User-Centric Clustering (UCC) model with Weighted KNN for Music Recommendation System

M.Sunitha[1], Dr.T.Adilakshmi [2], Dr. M.Sreenivasa Rao [3]

[1,2] Vasavi College of Engineering, Hyderabad, India,

[3] School of Information and Technology, JNTU, Hyderabad, India

Abstract: Music is Omnipresent, the growth of smartphones and internet led to the generation of large volumes of music in digital platforms. Music recommendation systems (MRS) serve as data filtering tools and handles data overloading problem. Collaborative Filtering (CF) is the most popularly used method in MRS. This paper is based on the model-based CF approach. Clustering is used to build User-Centric Clustering (UCC) model. UCC model is enhanced by combining with KNN (UCC-KNN). KNN finds nearest neighbours of a test user from the mapped user cluster in UCC. UCC-KNN is further enhanced by assigning weights to the neighbours. Based on the Weights of the neighbours, proportionate number of songs are considered for recommendation. Two weight assignment functions are explored in the paper. Proposed method UCC-W-KNN is experimentally evaluated on the benchmark dataset obtained from Last.fm. The results attained shows a considerable improvement over UCC and UCC-KNN models.

Keywords: Music recommendation system, User-Centric Clustering model, Weighted KNN

1. Introduction

The explosive growth of the Internet and the revolution in smartphones has generated huge volumes of data leading to information overloading problem. Users are provided with multiple choices in case of music, books, movies etc [6]. Recommendation system (RS) addresses information overloading problems. They have become an integral part of many online sites and are used to filter and prioritize the information according to the relevance and interests of the users. Both service providers and users are benefited with it. RS reduces searching cost, helps in decision making process, and increases user retention thus enhances total revenue [6]. Some of the e-commerce websites using RS are Amazon, MovieLens, eBay, CDNow etc.

Music recommendation system (MRS) is an information filtering tool which predicts the users interesting music tracks based on user's listening profile, thus enabling the music industry to handle huge volumes of data and satisfy the needs of users.

A music recommendation system has been studied in Collaborative Filtering (CF) and Content Centric approaches (CC) [6]. Collaborative filtering system recommends new items to the user by analysing items purchased by similar users, which is used in Amazon. Content Centric method recommends items with similar content as the items preferred by the target users, which is adopted by Pandora Radio. Hybrid method is obtained by combining the above two approaches. Netflix follows a hybrid approach combining both CF and CC. Collaborative filtering is proved as an effective method in recommendation systems because of its simplicity in understanding and implementation [6].

Model-based CF approach with weighted KNN is proposed in this paper. Rest of the paper is organized as, Section 2 shows the related work in the field of music recommendation, Section 3 describes proposed model, Experimentation results of the proposed method are shown in Section 4 and Section 5 explains the conclusion and future scope.

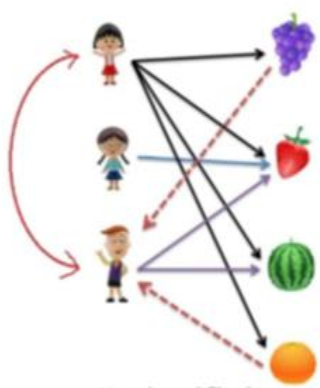

Figure 2.1 User-Centric CF method

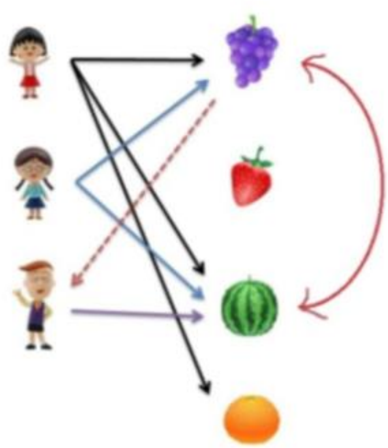

Figure 2.2 Item-Centric CF method

2. Related Work

Objective of a music recommendation system is to suggest music interesting to the users and help users in discovering new artists, items based on their interest [1]. Many music service providers such as Allmusic, Pandora, Audiobaba, Mog, Musicovery, Spotify and Apple Genius aggregated millions of users listening history and suggest music based on their interest. Most of the research in this field focused at suggesting a list of artists and a sequence of songs (playlist addressing personal interest of user) [1]. Research performed in music recommendation system can be categorized as Demographic based model, Collaborative filtering model and Content based model. Context based recommendation systems are also proposed in recent times [1]. Kuo, F.F., Shan, M.K in [2] proposed very basic type of music recommendation system based on the genre or tag of the song.

Demographic-based model

It is the easiest model in music recommendation system and based on the data related to songs such as song title, name of the artist, lyrics of a song to find the target songs. Even though it is easy and fast, this model requires user to know about the metadata of the songs and difficult to maintain metadata for huge number of songs/items. The important limitation of this method is users will never get a chance to explore new and novel songs [3].

Cheng Z, Shen J in [4] proposed a venue-aware music recommendation system. Schedl M in [5] proposed a country specific music recommendation system based on the data obtained from Last.fm.

Collaborative Filtering model

This is the most fundamental and most popularly used model for music recommendation system. The underlying principle of CF based method is if two users have similar taste in the past i.e., they liked few songs in the past, then they may also like similar kind of songs in the future. Collaborative filtering is further classified into three subcategories given as memory-based, model-based, and hybrid collaborative filtering. Memory-based collaborative filtering provides suggestions based on the nearest neighbours of users/items by using entire collection of previous user-item ratings. Memory-based CF is implemented in two different ways, one is user-based CF, and another is item- based CF. CF most commonly uses the approach to find nearest neighbours for any given user or item and provide recommendations from nearest neighbours. Former is known as user-based CF and later is known as item-based CF as shown in Figure 2.1 and Figure 2.2 respectively. Active users are mapped with nearest users based on similar interests, so that new items are recommended [6].

Elahi M, Ricci F, Rubens N in [7] explored active learning in CF models. Active learning assigns different importance to ratings based on their influence on the recommendations.

Hu R, Pu P in [8] proposed a method to enhance CF by including user personal information.

Personal information is also combined with user ratings to enhance CF method-based on only user ratings. Rashid AM, Karypis G, Riedl J in [9] combined offline process to gather information about users with online recommendations in CF method.

This paper is based on model-based CF method. Clustering is applied to generate user clusters based on their listening profile. Model-based CF addresses the problem of scalability and sparsity. User clusters thus generated are used for recommendations.

3. Proposed System

This section describes the proposed algorithm to enhance user-centric clustering (UCC) model proposed in [10]. User implicit feedback is used to construct user-song matrix. Based on the users listening profile, users are grouped into clusters. Optimal value for the number of clusters is obtained by using Elbow method. Once the user clusters are generated, they are used to provide recommendations to the test users. As the user-song matrix is very sparse, identifying the accurate recommendations from user clusters may not be possible. So, to improve the performance of UCC model it is combined with KNN. UCC with KNN is explained in section 3.1. It is further enhanced by a proposed method which assigns weights to the neighbors. UCC-W-KNN is described in section 3.2

3.1 UCC-KNN

UCC model [10] is applied to generated user clusters. KNN is combined with UCC to find the nearest neighbors of a test user from the mapped user cluster. The nearest neighbors are used to generate the recommendation vector as shown in the Algorithm 3.1. Consider the sample user-song matrix in Table 3.1 to show the working of UCC-KNN.

Table 3.1 Sample User-Song Matrix

Song/User	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}
S_1	2	0	0	0	0	3	1	2	0	2
S_2	0	4	0	1	0	0	0	0	3	5
S_3	1	3	0	0	0	3	0	3	1	0
S_4	0	0	0	1	1	4	3	0	4	0
S_5	0	1	0	0	4	0	1	1	0	2
S_6	1	0	3	4	0	0	2	0	2	0
S_7	0	1	0	3	4	0	0	2	0	2
S_8	0	1	0	4	0	2	3	0	1	0
S_9	1	0	0	1	0	0	0	4	3	2
S_{10}	4	0	4	0	4	0	2	2	0	0
S_{11}	0	2	3	0	1	0	3	1	0	0
S_{12}	3	2	2	2	0	3	1	0	3	0
S_{13}	2	0	1	3	0	2	0	0	4	0
S_{14}	0	2	0	0	0	1	0	2	1	2
S_{15}	0	0	4	0	2	4	2	3	0	0

User clusters generated with UCC model for K=3 is indicated in the Figure 3.1.

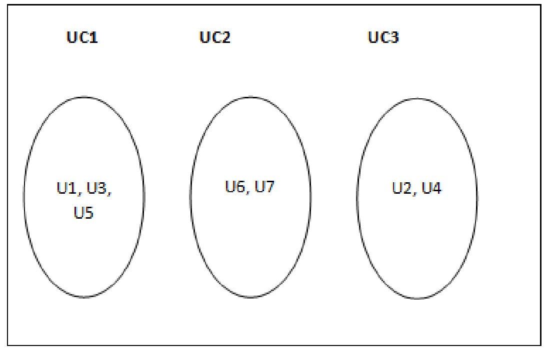

Figure 3.1 Showing User clusters generated for sample user-song matrix with K=3

Users U_8 to U_{10} are considered as test users from the sample user-song matrix. Each test user is mapped to the nearest user cluster. From the mapped cluster first, nearest neighbor is obtained for recommendations as shown in Table 3.2.

Table 3.2 Recommendation vector generated for sample test users based on UCC-KNN

	Mapped User Cluster	**Nearest neighbor with K=1**	**Songs Recommended**	**Precision**
Test_User$_1$ (U_8)	UC_1	U_1	S_1, S_3, S_6, S_{10}, S_{12}, S_{13}	0.16
Test_User$_2$ (U_9)	UC_3	U_4	S_2, S_4, S_6, S_7, S_8, S_9, S_{12}, S_{13}	0.88
Test_User$_3$ (U_{10})	UC_3	U_2	S_2, S_3, S_5, S_7, S_8, S_{11}, S_{12}, S_{14}	0.5
Average Precision				**0.513**

Algorithm 3.1: User-Centric-Clustering-KNN (UCC-KNN). Combines KNN with user centric model to generate recommendation vector

Input: User clusters, Test users

Output: Recommendation vector based on K nearest neighbours of a test user

Method:

1. Let the user clusters generated are UC_1, UC_2, ………. UC_k
2. For each test user $U_t \in \{U_1, U_2, \ldots\ldots, U_{60}\}$ repeat through steps 3 to 8
3. Find the nearest user cluster UC_i such that min (Proximity (U_j,UC_i)), where $1 \leq i \leq k$, $1 \leq j \leq 60$
4. For each user $U_i \in UC_i$
5. Compute the Euclidean distance from test user $U_j \in U_t$ as $d(U_j,U_i)$
6. Sort the users $\in UC_i$ in the ascending order of distance computed in step 4
7. Consider first K users from the ascending order list found in step 5
8. Generate recommendation vector (RS) from K users identified in step 6

Similar approach is followed for the user-song matrix obtained from Last.fm and the Experimentation results are shown in the section 4.

3.2 UCC-W-KNN

The User-Centric Clustering model with KNN is efficient but the important drawback is that it gives equal importance to all the neighbors. To overcome this limitation, an improved version of KNN is proposed in the paper to enhance the performance of music recommendation system.

3.2.1. Assigning weights to neighbors

Weights are assigned to the neighbors using the equations 3.1 and 3.2 in weighted KNN. Weighted KNN, assign more weight to the users who are nearby compared to the users who are farther away. Any function whose value decreases as the distance increases can be used to assign weights in a weighted KNN based recommendation system. One of the simple functions used in the research work is the inverse distance function as shown in equation 3.1.

Distance based weight function (DWF)

$$\text{Weight}(U_i) = \frac{1}{\text{distance}(U_t, U_i)} \qquad (3.1)$$

Where U_t is the test user and U_i is nearest neighbor and 1<=i<=k

Here distance is calculated using basic distance measures such as Euclidean, Manhattan and Supremum.

Dual weight-based function (DuWF)

Weights are calculated by using the formula given below in equation (3.2)

$$Weight(U_i) = \frac{d(U_i,U_k) - d(U_i,U_j)}{d(U_i,U_k) - d(U_i,U_1)} \text{ X } \frac{d(U_i,U_k) + d(U_i,U_j)}{d(U_i,U_k) + d(U_i,U_1)} \qquad (3.2)$$

Where U_i is the neighbor for whom weight is calculated

U_j is the list of nearest neighbors $\{U_1,U_2, \ldots U_k\}$, where U_1 is the first neighbor and U_k is the last neighbor in the list.

3.2.2. Recommendation vector with weighted KNN

Let us consider $\{U_1, U_2, U_3 \ldots\ldots U_k\}$ are the K nearest neighbors of a test user U_t. Different weights (**DWF and DuWF)** are assigned to the neighbors using equations 3.1 and 3.2. Let us assume that the corresponding weights are $\{W_1, W_2, W_3, \ldots\ldots, W_k\}$. To find Top-N recommendations from the neighbours, based on the weights, the proportionate number of recommendations are calculated using the formula given in equation 3.3.

$$N_{Songs(U_i)} = \frac{W_i}{\sum_j^k W_j} * N \qquad (3.3)$$

Where $1 \leq j \geq k$, N is the number of Top-N recommendations

Algorithm shown in 3.2 is applied to obtain K nearest neighbors for each test user and assign weights to the neighbors. Recommendation vector is created by considering the $N_{songs}(U_i)$ as shown in equation 3.3 from each neighbor.

Algorithm 3.2: User-Centric-Clustering-Weighted-KNN (UCC-W-KNN). Recommends Top-N Songs for test users

Input: User Clusters, Test Users

Output: Top-N recommendations for test users

Method:

1. Let user clusters generated with algorithm 3.1are $UC_1, UC_2, \ldots\ldots\ldots UC_k$
2. Let UC_i be the nearest user cluster to a test user U_t
3. For each user $U_i \in UC_i$ repeat through steps 4 to 8
4. Compute Euclidean distance of test user U_t and U_i which is represented as d (U_t, U_i)
5. Sort the users $U_i \in UC_i$ such that $d(U_t,U_i) < d(U_t,U_{i+1})$
6. Consider first K users from the sorted list in step 4. Let this list be $\{U_1, U_2, \ldots\ldots, U_K\}$
7. Compute the weight of each neighbor obtained in step 6 by using the equations 3.1 and 3.2
8. Add the number of songs from each neighbor based on the weights obtained in step 7 to generate Top-N recommendation vector by using the equation for N_{songs}

For sample user-song matrix shown in Table 3.1, consider K=2 to illustrate working of user-centric clustering model with weighted KNN to generate recommendation vector for the test users. Test users are mapped to the nearest user cluster and nearest neighbors and distance to the neighbors is shown in Table 3.3. Weights are assigned to the neighbors based on the DWF. Songs are proportionately added to the recommendation list based on the weights. Precision is computed with the recommendation vector suggested for each test user. Similarly, weights are assigned to the users in the user clusters formed with the original data obtained from Last.fm. Recommendations are generated based on the weights and shown in the next section.

4. Experimentation Results

In the process of recommendation with the user-centric clustering model, test users are mapped to the nearest user cluster. But all the users in the mapped user cluster might not be equally similar to the test user even though the user cluster is the nearest possible cluster among the existing user clusters. To address this issue, for each test user find the nearest neighbors from the mapped user cluster. Only the nearest neighbors are used to generate recommendation vector. Algorithm shown in 3.1 is applied to combine KNN with the user-centric clustering model. Implementation of UCC model with KNN is repeated for different values of number of neighbors i.e., K. Experimentation results suggested that as the K value is increased Precision of the UCC-KNN also increased as shown in Table 3.4 and Figure 3.1.

Table 3.3 Recommendation vector generated for sample test users with UCC-W-KNN

	Mapped User Cluster	**Nearest neighbors and distance**	**Weightage**	**Songs Recommended**	**Precision**
Test_User$_1$ (U_8)	UC_1	U_1-3.317 U_5-6.856	0.301 0.145	S_1, S_3, S_6, S_{10}, S_{12}, S_{13}	0.16
Test_User$_2$ (U_9)	UC_3	U_4-6.557 U_2-7.616	0.152 0.131	S_2, S_4, S_6, S_7, S_8, S_9, S_{12}, S_{13}	0.88
Test_User$_3$ (U_{10})	UC_3	U_2-7.616 U_4-8.718	0.131 0.114	S_2, S_4, S_6, S_7, S_8, S_9, S_{12}, S_{13}	0.5
Average Precision					**0.513**

Precision, Recall, and Accuracy values in case of All Songs, Top-50 Songs and Top-20 Songs recommendation methods are shown in Figure 3.1. The Top-20 Songs recommendation method has performed better compared to other two methods.

Table 3.4. Performance evaluation of UCC-KNN for different K values

No.of songs / No.of neighbours	K = 2			K = 5			K =10		
	P	R	A	P	R	A	P	R	A
All songs	0.16	0.15	0.87	0.16	0.24	0.83	0.16	0.33	0.77
Top-50 songs	0.33	0.01	0.92	0.38	0.02	0.90	0.41	0.02	0.91
Top-20 songs	0.38	0.01	0.92	0.44	0.01	0.90	0.45	0.01	0.91

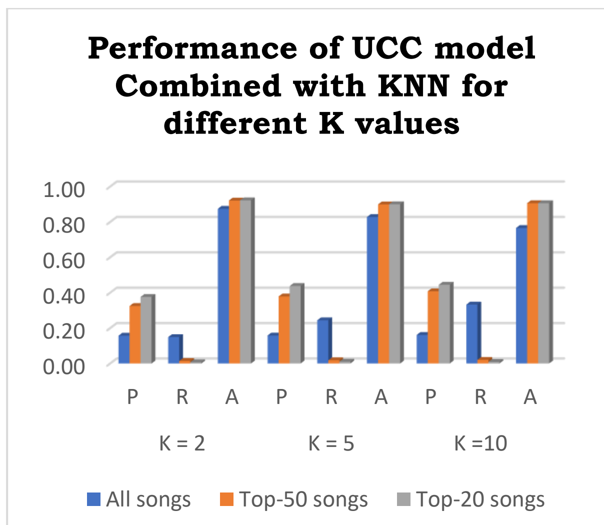

Figure 3.1 Showing the performance of UCC-KNN with different K values

Test users are mapped to the nearest user clusters by using the UCC model. Weights are assigned to the neighbour users present in the mapped user cluster by using DWF and DuWF methods. Based on the weights, the number of songs added to the recommendation vector from the user is computed using the equation 3.3. Recommendation vector formed from all the neighbours is compared with the actual user vector. Precision, Recall and Accuracy are computed. UCC-W-KNN performs better compared to UCC-KNN as shown in Table 3.5. Neighbour near to a test user is assigned more weight and added a greater number of songs to the recommendation list compared to the neighbours far from test users. UCC-W-KNN algorithm is repeated with different values of K such as 2,5,10. Experimentation results suggest that as the K value increased Precision value also increased. Precision, Recall and Accuracy values are obtained with DWF weight function in case of All Songs, Top-50 Songs and Top-20 Songs recommendation methods and results are shown in Figure 3.2.

Table 3.5 Performance evaluation of UCC-W-KNN recommendation system with DWF

No.of songs / No.of neighbours	K = 2			K = 5			K =10		
	P	R	A	P	R	A	P	R	A
All songs	0.16	0.15	0.87	0.16	0.24	0.83	0.16	0.33	0.77
Top-50 songs	0.36	0.03	0.91	0.39	0.03	0.91	0.41	0.03	0.91
Top-20 songs	0.40	0.08	0.92	0.48	0.02	0.92	0.49	0.02	0.92

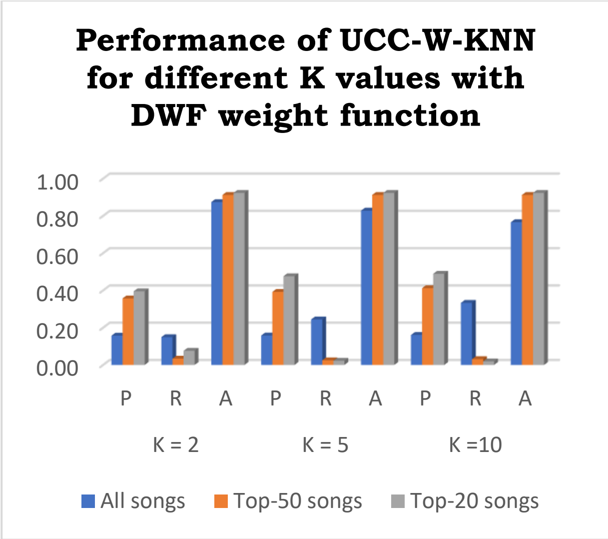

Figure 3.2 Showing the Performance of UCC-W-KNN with DWF for different K values

Table 3.6 and Figure 3.3 shows the Precision Recall and Accuracy values obtained with DuWF weight function in case of All Songs, Top-50 Songs and Top-20 Songs recommendation method. UCC-W-KNN with two weight assigning functions are explored to generate a recommendation vector. Algorithm is implemented for different values of K. DuWF weight function based UCC-W-KNN has performed slightly better compared to DWF.

Table 3.6 Performance evaluation of UCC-W-KNN recommendation system with DuWF

No.of songs / No.of neighbours	K = 2			K = 5			K =10		
	P	R	A	P	R	A	P	R	A
All songs	0.16	0.15	0.87	0.16	0.24	0.83	0.16	0.33	0.77
Top-50 songs	0.36	0.03	0.91	0.39	0.03	0.91	0.42	0.03	0.91
Top-20 songs	0.40	0.08	0.92	0.48	0.02	0.92	0.49	0.02	0.92

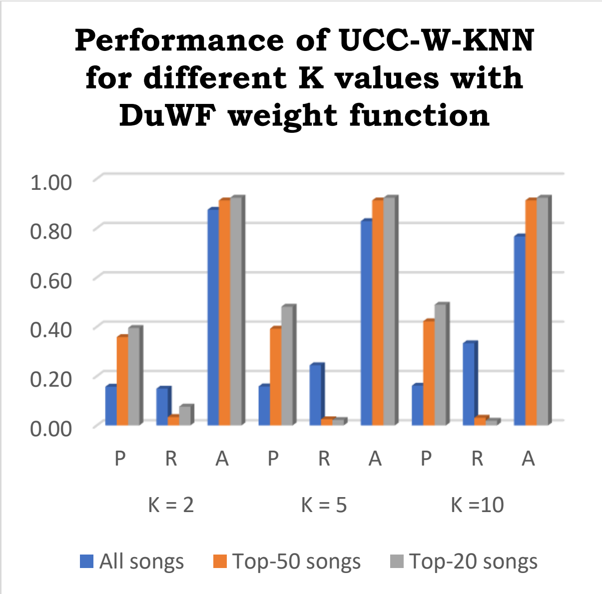

Figure 3.3 Showing the Performance of UCC-W-KNN with DuWF for different K values

5. Conclusion and Future Scope

User-Centric Clustering (UCC) model is enhanced in this paper by combining with KNN. Test users are mapped to a nearest user cluster. From the mapped user cluster, nearest neighbors are identified to provide recommendations. As the results suggest that UCC-KNN has performed better compared to UCC model. UCC KNN is further improved by assigning weights to the neighbors. Based on the weights, varying number of recommendations are added to the recommendation vector. Nearest neighbor is contributing more to the recommendation vector compared to a far neighbor. Experimentation results shown with two different weight functions indicate that DuWF function has performed slightly better than DWF weight function.

As music is so different in nature that providing accurate recommendations by considering users listening profile may not be possible. Other features such as Emotion of a music and lyrics constituting the music may also be added in future to design a more precise music recommendation system.

References

[1]. Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo & Mehdi Elahi, Current challenges and visions in music recommender systems research, International Journal of Multimedia InformationRetrieval volume 7, pages95–116, 2018.

[2]. Kuo, F.F., Shan, M.K, A Personalized Music Filtering System Based on Melody Style Classification, IEEE international Conference on Data Mining, pp. 649–652, 2002.

[3]. Billy Yapriady, Alexandra L. Uitdenbogerd ,Combining Demographic Data with Collaborative Filtering for Automatic Music Recommendation, International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, Lecture Notes in Computer Science book series, pp 201-207, 2005.

[4]. Cheng Z, Shen J, On effective location-aware music recommendation. ACM Trans Inf Syst 34(2):13, 2013.

[5]. Schedl M, Investigating country-specific music preferences and music recommendation algorithms with the LFM-1b dataset. Int J Multimed Inf Retr 6(1):71–84, 2017.

[6]. 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 17(6):734–749, 2005.

[7]. Elahi M, Ricci F, Rubens N, Active learning in collaborative filtering recommender systems, Hepp M, Hoffner Y (eds) E-commerce and web

technologies, Lecture Notes in Business Information Processing, vol 188. Springer, pp 113–124,2014.

[8]. Hu R, Pu P, Enhancing collaborative filtering systems with personality information. In: Proceedings of the fifth ACM conference on recommender systems, RecSys'11. ACM, New York, NY, USA, pp 197–204, 2011.

[9]. Rashid AM, Karypis G, Riedl, Learning preferences of new users in recommender systems: an information theoretic approach. SIGKDD Explore News 10:90–100,2008.

[10]. M.Sunitha, Dr.T.Adilakshmi, Mir Zahed Ali, Enhancing User-Based Collaborative Filtering with user Correlations for Music Recommendation System, International Journal of Advanced Trends in Computer Science and Engineering, ISSN:2278-3091, Vol 9, No 5, Sep-Oct, 2020