Context-aware Ontological Hybrid Recommender System For IPTV

6 authors, including:

Mohammad Wahiduzzaman Khan
Multimedia University

Gaik-Yee Chan
Multimedia University

Fang-Fang Chua
Multimedia University

Su-Cheng Haw
Multimedia University

Context-aware Ontological Hybrid Recommender System For IPTV

Mohammad Wahiduzzaman Khan[1], Gaik-Yee Chan[2*], Fang-Fang Chua[3], Su-Cheng Haw[4]
[1,2,3,4]Faculty of Computing and Informatics
Multimedia University, Cyberjaya, Malaysia
[1]avy164@gmail.com
[2]gychan@mmu.edu.my
[3]ffchua@mmu.edu.my
[4]sucheng@mmu.edu.my

Muhsin Hassan[5], Fatimah Almah Saaid[6]
[5,6]Customer Insight and Analytics, TM Unifi
Telekom Malaysia, Kuala Lumpur, Malaysia
[5]muhsin.hassan@gmail.com
[6]almah@tm.com.my

***Abstract*—With the huge growing amount of information continuously produced, shared and available online, finding relevant and beneficial contents or services at a single or few clicks have become almost impossible. Most of the time, we will be returned with thousands of irrelevant web links. As such, a recommender system which recommends contents or services that likely meet the user's needs is crucial, especially in the IPTV domain when the choices for program selection has no time and physical boundary restriction. The two major recommendation techniques are content based and collaborative filtering. Nevertheless, such techniques still suffer from several problems such as cold start, data sparsity and over specialization. Our proposed system namely COHRS is a context-aware recommender system based on ontological profiling under the IPTV domain. Ontological approach improves user profiling process and thus improving the accuracy of a recommendation system. Experimental evaluations indicate that COHRS is able to overcome the drawbacks such as over specialization, data sparsity and inefficiency issue of most traditional recommender systems.**

***Keywords*—clustering; collaborative filtering; content-based; hybrid approach; ontology**

I. Introduction

Internet Protocol TeleVision (IPTV) is a digital distribution of television service that is provided through ranges of internet protocol such as User Datagram Protocol (UDP) and Real-time Transport Protocol (RTP). The key difference between IPTV and traditional television service is that there is no time restriction. User can watch a particular program anytime he or she wants. IPTV also carries a wide variety of programs where user is not bound to any specific type. IPTV unlashes a new freedom to the users. But the wide variety of programs sometimes brings trouble for the user because most people are overwhelmed when they are provided with lots of choices in which some of these choices may not be of interest to them. Users may then make bad choice or no choice at all. Therefore, we require an intelligent system which can predict whether a certain program will be preferred by the user in considering the contextual information such as user's age, gender, occupation, location and time of watching of the program. Such kind of system is known as contextual based recommender system. Additionally, contextual information has to be linked in a more meaningful way ontologically so as to provide deeper knowledge about the users or users' profile for better quality recommendation.

Generally, a recommender system is a system which helps individual in a community to find information and item which is most relevant to him or her [1]. Myriad research has been done in the field of recommender system since the middle of 90's till recently. Recommendation system is divided into two major categories, namely content-based and collaborative filtering approach. But these two approaches suffer from various drawbacks such as the cold start problem whereby the recommender system has very little information about the new user or new item for recommendation; over specialization problem in which the recommender system only recommend certain type of recommendation, hence confining the user's taste to a particular type of item; data sparsity problem whereby the recommender system is not able to recommend item to the targeted user due to small amount of historical transactional information of the targeted user; and last but not least, the gray sheep problem whereby the user having certain peculiar choice of item, so it is hard for traditional recommender system to recommend items to this type of users. Therefore, in this paper, a context-aware hybrid, by combining content-based and collaborative filtering approach and along with ontological user profiling under the IPTV domain, is proposed with the intention to resolve the problems of traditional recommender systems.

II. Background and Related Works

This section discusses the various approaches to recommendation, mainly content-based, collaborative filtering and hybrid.

A. Content-based Recommendation

Content-based recommendation is a recommendation approach whereby recommendations are provided on the basis of item-item similarity [2]. Item-item similarity can be calculated from item feature, properties or meaningful item description. For example, a movie's genre, director, actor can be considered as item features and its synopsis can be considered as meaningful item description. After calculating item-item similarity, a user profile from the previous transactions of the targeted user could then be built. Hence

*Corresponding author: Email: gychan@mmu.edu.my
Phone #: +603 8312 5215

after, based on this user profile and item-item similarity, recommendations can then be provided. For example, there are two movies, 'The Da Vinci Code' and 'Angels and Demons' whose genre belong to Thriller and Mystery category with actor Tom Hanks. So these two movies are closely similar. If a user likes the movie 'The Da Vinci Code', it is highly probable that this user would also like the other movie, 'Angels and Demons' that has similar properties.

Yet, there are some key challenges for content-based approach. First, an efficient method has to be used in order to get the meaningful data from the system. The data extracted from the system must be represented in a way that it should be linked to both user and item in a meaningful way. Additionally, content-based recommendation approach suffers from various drawbacks including over specialization problem and limited content analysis problem. When recommendations are generated on the basis of user's previous transactions and item-item similarity, this shall confine the user's taste on a particular pattern. Hence, a person who has not watched a comedy movie shall never get recommended for the greatest comedy movie [3]. Content-based recommendation approach also suffers from limited content analysis problem when the features that are automatically extracted from the system are not sufficient to provide meaningful analysis. Moreover, automatic feature extraction of multimedia data would be very challenging and manual insertion of features would not be feasible as well.

B. *Colloborative Filtering Approach*

The key idea behind collaborative filtering approach is people who agree in the past will agree in the future [4]. Collaborative filtering approach first builds a user profile on the basis of user previous transaction. Next it compares the targeted user's profile with the rest of the users in the system to find similar users. After finding similar users, predicted users' ratings are calculated for a particular item which is watched by those similar users but not watched by the targeted user. This calculation is based on similarity index between targeted user and those similar users.

Collaborative filtering system can be potential solutions for solving over specialization problem. But collaborative filtering approach suffers from data sparsity problem. Collaborative filtering approach cannot perform well when there is limited information about the target user based on past transactions for finding similar users [4]. Similarly, collaborative filtering approach also suffers from new user problem because the system has very little information about the new user, thus fail to locate similar users [5]. Collaborative filtering approach also suffers from the gray sheep problem. There may be users whose preference would be unusual as compared to that of other users in the system. In such case, it would be very difficult to find similar users using collaborative filtering approach [5].

C. *Hybrid Recommendation*

Hybrid recommendation approach combines more than one recommendation techniques in order to achieve better result and performance. Research done in [6] uses a probabilistic method that describes a unified Boltzmann machine by combining collaborative filtering and content-based approach in a coherent manner. This approach is able to solve the cold start problem. However, the hybrid approach provides very little understanding of users and their preferences. Another research by [7] also makes use of content-base and collaborative filtering approach to solve cold start problem with satisfactory result. Although this system can recommend less popular item, yet it still suffers from limited information about the users.

D. *Ontology*

Ontology, in computer science terminology, represents knowledge in a certain domain, presented in a manner understandable by human and readable by machines. The major components of ontology are class, instance, properties and rules [8]. Different number of features could formulate into a certain class. Class instances are specific and have different values. Relationships between classes are based on binary semantic relationship. For example, the 'User' class and the 'Gender' class could be semantically linked through the property 'has'. Translated into semantic knowledge, it would mean 'User' has 'Gender'.

Hence, compare to traditional keyword-based recommendation system, ontology provides richer semantic knowledge and understanding of a certain domain. Thus motivated, our research therefore focuses on building a context-aware ontological hybrid recommender system with the intention of resolving problems found in traditional approaches as mentioned above.

III. OUR PROPOSED RECOMMENDER SYSTEM-COHRS

Our proposed recommender system is a Context-aware Ontological Hybrid Recommender System (COHRS). An architecture overview of COHRS is shown in Fig. 1 (read from left to right, top to bottom following the steps 1-5).

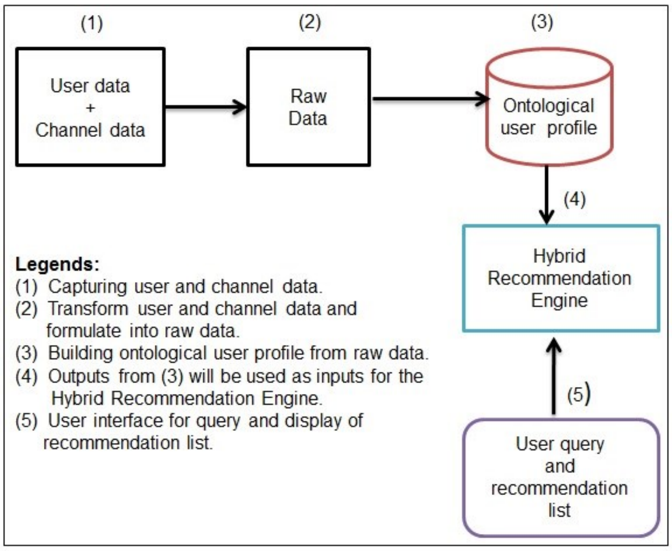

Fig. 1. An overview of COHRS

A. *Capturing User and Channel Data*

Refer to Fig. 1 (Step1), this context-aware ontological hybrid recommender system is using a proprietary IPTV data set which contains both the user and TV channel information. For example, channel usage information will provide the number of minutes a user watches a particular channel. User's contextual information would be user's age, gender, location and so on. The channel level data are channel genre, channel id and channel name. The channel usage data in this data set are scattered and there is no particular pattern to be observed from it. Therefore, there is a need to use intelligent technique to divide the channel usage data into clusters that form into similar channel usage pattern.

Consequently, clustering technique is used for the unsupervised way of classification. Clustering is a technique of grouping the data of similar objects. Each group is known as a cluster and every cluster's values are comparatively similar [9]. We carried out experiment using various clustering techniques such as *K*-means clustering, Expectation Maximization (EM), Hierarchical cluster, Mark density based cluster and Farthest First onto the channel usage data. The aim of this clustering is to classify the channel usage into five different categories from lowest channel usage to highest channel usage that maps into user's preference of zero to one. For instance, if a user's channel usage for a particular channel is very low then our classification algorithm will classify this channel usage into the first cluster which resembles a poor preference value and given a value of 0.2 by our recommendation engine. Subsequent mapping of user's preference is done for the other four categories of channel usage.

***K*-means clustering** is also known as partitioning clustering whereby data in a given data set are divided into *k* different clusters through an iterative method until it converges to a local minimum. This algorithm works in two steps. First, the algorithm chooses explicitly *k* different centers based on distance calculation until all the data of a data set are attached to some centers. Next step is to recalculate the average until it turns into a minimum value [10].

Expectation Maximization (EM) is an iterative process to find maximum similarity of parameter in a mathematical model. EM iteration has basically two major steps, expectation step and maximization step. In expectation step, expectations of the log-likelihood are measured in terms of current parameters. The maximization step will maximize the log-likelihood value of the expectation step. EM thus determines a probability value for each data in a data set belonging to a particular cluster [11].

Farthest First clustering is a modified version of *k*-means clustering algorithm which places each cluster center from further most distance from existing cluster. This algorithm requires less modification and reassignment, therefore perform faster in terms of clustering speed [12].

Hierarchical clustering creates a hierarchical model of data which forms into a tree structure with several small data subsets. The trees could be formed in either bottom up or top down way. The bottom up or the agglomerative approach starts by forming small number of groups with available data sets then progressively merging the groups into one. The top down or divisive approach starts by forming a single cluster with all the data in a given data set. After performing a few successive iterations, each data is then assigned to a particular cluster [13].

Mark Density based clustering formulates the clusters on the basis of density of a data point in a region. It groups together points that are proximately packed together and marking outliers that lie alone in low-density regions. [14].

Experiments are being conducted to determine the most efficient clustering algorithm for our proposed recommender system. All the clustering algorithms mentioned above are run using WEKA 3.8 [15] to evaluate their time performances based on the proprietary IPTV data set, in which channel usage in minutes are classified into five categories.

This proprietary data set consists of 170 instances with minimum channel usage of 1.032 minutes and maximum channel usage of 1540.38 minutes and standard deviation is 306.616.

The time performances of the five clustering algorithms are tabulated as shown in Table I. As can be seen from Table I, both *k*-means clustering and Farther First take the shortest time (0.01 second) in forming the five clusters whereas EM takes the longest time of 0.15 second to form the five clusters. For our COHRS, *k*-means clustering is adequate to be used for classification of a single attribute, the discrete value of channel usage. Refer to Table II for the five clusters generated by *k*-means clustering and the mapping to each user's preference category.

TABLE I. TIME PERFORMENCES OF FIVE CLUSTERING ALGORITHM

Algorithms	Time taken to form 5 clusters (seconds)
K-means clustering	0.01
Expectation maximization(EM)	0.15
Hierarchical cluster	0.06
Mark density based cluster	0.02
Farthest First	0.01

TABLE II. MAPPING OF CHANNEL USAGE TO USER PREFERENCE

Channel usage (minutes)	User preference (0-1)
12.918	0.2 (lowest preference)
78.57	**0.4 (low preference)**
179.316	0.6 (average)
411.114	0.8 (high preference)
919.752	**1.0 (highest preference)**

B. Raw Data Formation

Refer to Fig. 1 (Step 2), the raw data consists of user's profile such as user's id, location, age, gender along with user preference values for channel usage and channel profile which includes channel id and channel genre. These data are stored in column and row format as shown in Table III.

TABLE III. RAW DATA IN ROW AND COLUMN FORMAT

User id	10	10
location	‘P town’	‘P town’
age	42	42
gender	Male	Male
Channel id	Iptv19753	Iptv42337
Preference value	**0.4**	**1.0**
Genre	Indian	Other

The raw data of user id 10 could be transformed ontologically as shown in Fig.2. This ontological user profile could be interpreted as: user id 10 is a male aged 42 years staying in P town and watches channel Iptv19753 with low preference value of 0.4 (Table III Column 2 Row 6, correspond to Table II Column 2 Row 3). In another instance, user id 10 also watches channel Iptv42337 with highest preference value of 1.0 (Table III Column 3 Row 6, correspond to Table II Column 2 Row 6).

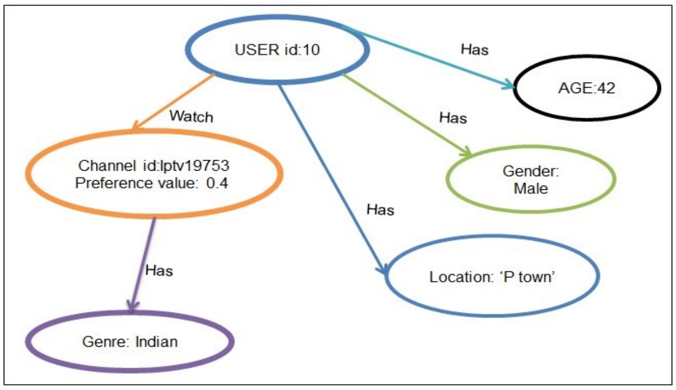

Fig. 2. A sample ontological user profile

C. Building Ontologocal Profile

Refer to Fig. 1 (Step 3), an ontological user profile is built using (1) [16]. Research in [16] uses (1) to create ontological user profile by computing user preference values for different movie genre while for our case, (1) is used to obtain user preference values for the different IPTV concepts.

$$W_{u,f} = \frac{1}{N}\sum_{f\in features(dc)} W_{u,c} \qquad (1)$$

where N is the number of channel watched by user u and $W_{u,f}$ is the ontological preference value for user u for a particular feature or concept f. Feature is the type or genre of a channel, dc is the list of channels with feature f and is the preference value for user u for a particular channel c that has been determined by k-means clustering. Using (1) we calculated user preference for each concept in our system. Table IV shows preference values for different concepts, such as Sports, News, Lifestyles, Kids and others type for user id 5 and user id 10 generated by our ontological hybrid recommendation engine. Refer to Table IV, user id 10 most preferred concept is the “Other” category while user id 5 most preferred categories are “Life style” and “Indian”.

TABLE IV. USER PREFERENCE FOR EACH CONCEPT

Concepts	Preference values for User id 10	Preference values for User id 5
Chinese	0	0
Sports	0.03	0.01
News	0.01	0
Life style	0.09	0.2
Indian	0.02	0.2
Kids	0.03	0.04
Other	0.19	0.12

D. Formation of Recommendation and Display of Results

Refer to Fig. 1 (Step 4 and Step 5), the next step for our hybrid recommendation engine is to calculate ontologically similar users using (2) [17]. In [17], (2) is used to find the similarity value between two users for book recommendation where features represent the different book genre such as ‘mystery’, ‘children’s book’, ‘history’ and so on. In our COHRS, (2) is used to compute the similarity value between two users for recommendation based on the concepts under the IPTV domain.

$$sim(u,v) = \sqrt{\sum_{f\in features}\left(W_{u,f} - w_{v,f}\right)^2} \qquad (2)$$

where Sim(u,v) is the ontological user similarity between user u and user v. We have to calculate ontological user similarity value for all users for the targeted user. The lower this similarity value, the closer would be the similarity. The similarity value between user id 10 and five other users are tabulated as shown in Table V. From Table V, user id 10 has the closest similarity with user id 5 who has the lowest similarity value (0.1) as calculated from (2).

TABLE V. FIVE OTHER USER SIMILARITY VALUES FOR USER ID 10

User id	1	2	3	4	**5**
Similarity values	0.19	0.21	0.25	0.15	**0.1**

Next is to find k-most similar users. For our initial experiment, k=5 is used. This means we will consider 5 most similar users. Similar user can be calculated by sorting the similarity values in ascending order. For example, in Table V, the five most similar users for user id 10 in ascending order would be, user id 5, 4, 1, 2, and 3.

Subsequent to this is to calculate the predicted user preference for a particular channel for the targeted user. This is to find the channel that has already been watched by those similar users but not watched by the targeted user. This predicted user preference is calculated based on (3) [18]. Research in [18] uses (3) to calculate predicted user rating for movie domain whereas for COHRS, (3) is used to compute predicted user rating for IPTV domain.

$$P_{u,i} = \overline{r_u} + \frac{\sum_{v \in K} Sim(u,v) * (r_{v,i} - \overline{r_v})}{\sum_{v \in K} Sim(u,v)} \qquad (3)$$

where $P_{u,i}$ is predicted user preference of user u for channel i. $\overline{r_u}$ is the average user rating for the targeted user, $r_{v,i}$ is the user preference of user v for channel i and $\overline{r_v}$ is the average user preference for user v. K is the list of k-most similar users. After calculating all the predicted user preferences for targeted user, the list is sorted in descending order and top-N items are picked for the targeted user. For our initial experiment, N is equal to 5. Table VI shows the preliminary results of top-5 channels for user id 10.

TABLE VI. TOP-5 CHANNELS FOR USER ID 10

Channel id	Genre	Predicted preference value
iptv174	Other	1.00
iptv11080	Life style	0.70
iptv18586	Other	0.67
iptv29330	Indian	0.64
iptv10969	News	0.47

We will now consider some contextual information like user age, gender and location. First we need to calculate user similarity on the basis of age, gender and location.

Table VII shows user similarity values of five other users for user id 10. It can be seen that user id 10 has highest contextual similarity with user id 14 (similarity value of 3). This means that they are similar in terms of age, gender and location. Therefore, the k-most, in this case, k=5, similar users for user id 10 in descending order would be user id 14, 15, 12, 13 and 11.

Subsequent to this, it is required to calculate predicted user preference for the targeted user. First is to find out the list of channels that has already been watched by k-most similar users but not watched by the targeted user using (3). After calculating all the predicted user preferences for the targeted user, the list is sorted in descending order and top-N items are picked for the targeted user. For our initial experiment, N is equal to 5. Table VIII shows the preliminary results of top-5 channels for user id 10.

TABLE VII. USER CONTEXTUAL SIMILARITY

User id	11	12	13	**14**	15
Similarity value	0	1	1	**3**	2

TABLE VIII. TOP-5 CHANNELS FOR USER ID 10 BASED ON CONTEXTUAL INFORMATION WHEN K=5

Channel id	Genre	Predicted preference value
iptv29330	Indian	0.90
iptv244	Other	0.87
iptv181	News	0.67
iptv8031	News	0.64
iptv28977	News	0.62

IV. PERFORMANCE EVALUATION

In this section, we will discuss the performance of our recommendation engine based on various scenarios.

Our proposed context-aware recommendation engine will not confine user taste into the genre he or she likes most rather it broaden the user taste by recommending various ranges of item. Table VI and Table VIII recommending various genre types including Lifestyle, News, Indian and Others to user id 10. Thus, it solves the over specialization problem.

Another major problem of any recommender system is the new user problem. When a new user comes into the system, it is very difficult to recommend item to this new user as there is very little information about his preference. However, by considering contextual information such as age, gender and location, COHRS is able to recommend items based on the contextual information. In order to test this, a new user with id 100, age 45, gender is male and location is 'P town' (same age, gender and location as user id 10) is manually inserted into the system. COHRS successfully recommends five channels to this new user based on contextual similarity (refer to Table IX). COHRS, thus, addresses the new user or cold start problem.

TABLE IX. TOP-5 CHANNELS FOR NEW USER ID 100

Channel id	Genre	Predicted preference value
iptv182	Other	0.81
iptv244	Other	0.52
iptv11080	Life style	0.52
iptv29330	Indian	0.50
iptv8031	News	0.31

Another drawback of traditional recommender system is the new item problem. When a new item comes into the system, it has insufficient number of ratings to enable the traditional recommendation engine to recommend it to certain users. Experiment is conducted by manually inserting a channel, 'mamamia', to the system and stated that it has been watched by user id 5, who is one of the k-most similar user for user id 10. COHRS is able to calculate preference value for this new item and recommend it to user id 10 (refer to Table X). COHRS thus successfully resolves the new item or cold start problem.

TABLE X. NEW TOP-5 CHANNELS FOR USER ID 10

Channel id	Genre	Predicted preference value
iptv174	Other	1.00
iptv11080	Life style	0.70
mamamia	**Other**	**0.69**
iptv18586	Other	0.67
iptv29330	Indian	0.64

The data set used in COHRS consists of 18 users with 58 channels information. With such comparatively little information, yet COHRS is able to recommend channel for user id 10 (refer to Table VI and VIII). This means COHRS is able to resolve the data sparsity problem.

Traditional content-based recommendation engine needs to maintain n*n matrix for finding the similarity values between each item, where n is the number of items in the system. Additionally, for traditional collaborative filtering, there is a need to maintain m*m matrix for keeping similarity values between each user in the system, where m is the number of users in the system. However, for COHRS where data are represented as classes and stored as instances in a hierarchical and graphical format (refer to Fig. 2), it is thus not required to maintain n*n or m*m matrix. To elaborate further, suppose a hybrid recommender system of content-based with collaborative filtering is used, then for 18 users and 58 items, the system has to maintain 18*18 users matrix and 58*58 items matrix for the computation of user-user similarity and item-item similarity values which is memory intensive. Furthermore, if a new user or a new item comes into the system, the matrix becomes 19*19 and 59*59 respectively for similarity user and item computation which is time and memory intensive. However, for COHRS using graph based or ontological architecture where data are represented as classes and stored as instances in the system, it requires only one time access for computation of similarity value. Hence, scalability or performance issue if not totally eliminated, but would at least be reduced by using COHRS.

V. CONCLUSION AND FUTURE WORK

In this paper, we have demonstrated how our proposed context-aware ontology hybrid recommendation system can be used to overcome the drawbacks of traditional content-based and collaborative filtering recommendation system's problems such as the cold start problem, data sparsity problem and over specialization problem. It is seen that COHRS is also memory and time efficient as it does not have to maintain matrix for computation of either item-item or user-user similarities. Further enhancement can be made possible by adding more contextual features like time of watching and occupation of a user. Evaluating performance of our proposed recommender system can be further extended to using large scale real time data sets for computation of 10 to 100 most similar users. In order to determine the accuracy of recommendation, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) could be used as measurements for the difference between predicted user preference and actual user preference. The lower the error measures, the higher would be the recommendation accuracy.

ACKNOWLEDGEMENT

This work is supported by the funding of TM R&D from the Telekom Malaysia, Malaysia.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," Recommender Systems Handbook, pp. 1–35, May 2010.

[2] Z. Çataltepe, M. Uluyağmur, and E. Tayfur, "Feature selection for movie recommendation," Turkish Journal Of Electrical Engineering & Computer Sciences, vol. 24, pp. 833–848, 2016.

[3] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," The Adaptive Web Lecture Notes in Computer Science, pp. 325–341, 2007.

[4] J.B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative filtering recommender systems," In Adaptive Web, LNCS 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin Heidelberg: Springer-Verlag, 2007,pp. 291 – 324.

[5] L. Zhang, Q. Tao, and P. Teng, "An Improved Collaborative Filtering Algorithm Based on User Interest," Journal of Software, vol. 9, no. 4, Jan. 2014.

[6] A. Gunawardana and C. Meek, "A unified approach to building hybrid recommender systems," Proceedings of the third ACM conference on Recommender systems - RecSys 09, 2009.

[7] K. Pripuzic, I. Podnar Zarko, V. Podobnik, I. Lovrek, M. Cavka, I. Petkovic, P. Stulic, and M. Gojceta. "Building an IPTV VoD Recommender System: An Experience Report," Proceedings of the 12th International Conference on Telecommunications (ConTEL'13),Zagreb, Croatia: IEEE, 2013. 155-162

[8] Z. Bahramian and R. A. Abbaspour, "An Ontology-Based Tourism Recommender System Based On Spreading Activation Model," ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XL-1-W5, pp. 83–90, Oct. 2015. O.A. Abbas, "Comaprison between data clustering algorithms," in International Arab Journal of Information Technology, vol. 5(3), July 2008.

[9] O.A. Abbas, "Comaprison between data clustering algorithms," in International Arab Journal of Information Technology, vol. 5(3), July 2008.

[10] R. Loohach and K. Garg, "Effect of Distance Functions on Simple K-means Clustering Algorithm," International Journal of Computer Applications, vol. 49, no. 6, pp. 7–9, 2012.

[11] Sharmila and R.C. Mishra, "Performance evaluation of clustering algorithms," in International Journal of Engineering Trends and Technologies, vol. 4, Issue 7, July 2013.

[12] N. Sharma, A. Bajpai, and M. R. Litoriya, "Comparison the various clustering algorithms of Weka tools," Int. J. Emerg. Technol. Adv. Eng., vol. 2, no. 5, May 2012.

[13] J. Han, M. Kamber, and A. K. H. Tung, "Spatial clustering methods in data mining," Geographic Data Mining and Knowledge Discovery, pp. 188–217.

[14] M. Ester, HP. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in second International Conference on Knowledge Discovery and Data Mining (KDD'96), E. Simoudis, J. Han, and U. Fayyad (Eds.). AAAI Press 226-231.

[15] Weka 3.8, Datamining software, https://www.cs.waikato.ac.nz/ml/weka/

[16] I. Cantador, A. Bellogín and P. Castells. "A multilayer ontology-based hybrid recommendation model," in AI Communications, vol. 21, Issue 2-3, pp. 203-210. April 2008, IOS Press.

[17] Z. Zhang, L. Gong, and J. Xie. "Ontology-based collaborative filtering recommendation algorithm," in Advances in Brain Inspired Cognitive Systems, vol. 7888, D. Liu, C. Alippi, D. Zhao and A. Hussain, Eds. Springer, Berlin, Heidelberg:LNCS, 2013, pp. 172-181.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens," Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW 94, 1994.