ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 3, Issue 1, January 2014*

# COMBINATION OF WEB PAGE RECOMMENDER SYSTEMS

**MR. S. R. PATIL[1] PROF. S.K. SHIRGAVE[2]**

COMPUTER SCIENCE AND TECHNOLOGY, SHIVAJI UNIVERSITY, KOLHAPUR, MAHARASHTRA, INDIA[1]

DKTES' TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI, MAHARASHTRA, INDIA.[2]

**Abstract:** World Wide Web is the biggest source of information. Though the World Wide Web contains a tremendous amount of data, most of the data is irrelevant and inaccurate from users' point of view. Consequently it has become increasingly necessary for users to utilize automated tools such as recommender systems in order to discover, extract, filter, and evaluate the desired information and resources. Web page recommender systems predict the information needs of users and provide them with recommendations to facilitate their navigation. Web content and Web usage mining techniques are employed as conventional methods for recommendation. The most common Web usage mining techniques used for recommender system are Markov models, Association rules and Clustering. These techniques have strengths and weaknesses. Combining different systems to overcome disadvantages and limitations of a single system may improve the performance of recommenders. Hybrid recommender systems can be used to avoid the drawbacks or limitations of previous recommendation method. They combine two or more method to improve recommender performance. In this paper, the four recommender systems are combined by using different hybridization methods. The effects of the hybrid recommenders are examined by comparing the results of hybrid system against the results of single recommendation method. Result shows that the hybrid recommender provides successful recommendation when the recommended page is generated by all the systems of the hybrid.

**Keywords:** Web usage mining, Recommender Systems, Hybridization Methods.

## I. INTRODUCTION

Different Web usage mining techniques have been used to develop efficient and effective recommendation systems. User satisfaction is the most important part of the recommender system. Today the quality of recommendations and the user satisfaction with such systems are still not most favorable. Recommender systems are not favorable for quality of recommendations and user satisfaction. Methods used for the recommender system focuses on the different characteristics of the user. As a result, for the same data set, two recommender systems show the two different results. The most common Web usage mining techniques used for recommender system are Markov models, Association rules and Clustering. These techniques have strengths and weaknesses. For example lower order Markov models lack accuracy because of the limitation in covering enough browsing history; whereas higher order Markov models usually result in higher state space complexity. Association rule mining is a major pattern discovery technique. The main limitation of association rule mining is that many rules are generated, which result in contradictory predictions for a user session. Second limitation is that association rule mining is a non-sequential mining technique that does not preserve the ordering information among pageviews in user sessions. Recommendation system based clustering can capture a broader range of recommendations, though this is sometimes at the cost of lower prediction accuracy. Another drawback is Clustering methods are unsupervised methods, and normally are not used for classification directly. Consequently, combining different systems to overcome disadvantages and limitations of a single system may improve the performance of recommenders. Hybrid recommender systems can be used to avoid the drawbacks or limitations of previous recommendation method. They combine two or more systems to improve recommender performance. In this paper, hybrid recommender methods combining the results of different recommender systems are constructed in the following way: Initially recommender system is implemented separately then the resulting predictions are combined by using hybrid recommender methods. Four hybridization methods are used namely weighted, Hit-Ratio based mixed method, switching and frequency based ranking. In this paper, the effects of the hybrid recommenders are examined.

This is achieved by comparing the results of hybrid system against the results of single recommendation method and its performance is evaluated based on the correct prediction of the next request of a user, namely Hit-Ratio. Our detailed experimental results show that when choosing appropriate combination methods and modules, hybrid approaches achieve better prediction accuracy.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 3, Issue 1, January 2014*

## II. PROBLEM STATEMENT

To design and develop hybrid recommender system to provide improved recommendations, which can be used for personalization.

## III. PROPOSED SYSTEM

The system contains two phases Off line phase and On line phase.

*A. Off line phase:*

The off line phase contains two components as follows

*1. Data pre-processing:*

The data pre-processing includes four main processes namely data collection, data cleaning, user identification and sessionization.

*2. Pattern extraction component:*

The pattern extraction component consists of four different modules each of which is a recommender system using a different technique. These modules are clustering, association rule discovery, markov model and click-stream tree.

*B. On line Phase:*

The On-line Phase also consists of two components:

*1. Recommendation Engine:*

The work consists of the implementation of Recommendation engine which consist of four recommender techniques namely Recommender model based on clustering user sessions, Association Rule discovery, click-stream tree and Markov model**.**

*2. Hybridization Component:*

The system combines results of multiple recommender models together to produce a single output. In the hybridization component weighted, Hit-Ratio based mixed method, switching and frequency based ranking methods are used to combine recommendation.

## IV. IMPLEMENTATION

*A. Data Pre-processing:*

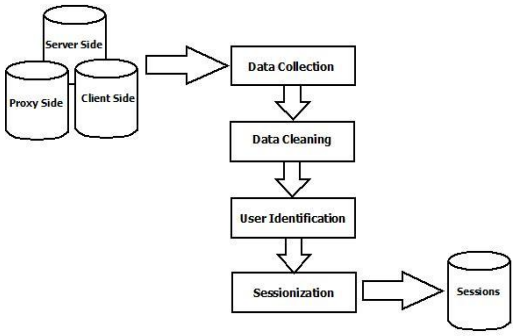


Fig. 1 Phases of data pre-processing technique

The data pre-processing technique contains four processes. These processes are:

*1. Data Collection*

The main data source in Web Usage Mining and personalization process is information residing on the Web sites logs. The key input to the pre-processing phase is the server logs. There are three most common sources of data.

*2. Data cleaning:*

The data in the original Web user log files are raw; hence, not all the log entries are valid for Web Usage Mining. Thus the main purpose of this process is to clean all the log files. All log entries with file name suffixes such as gif, JPEG, jpeg, GIF, jpg, JPG removed. As well as the entire request from the Web spiders are also removed from Web log files.

*3. User Identification:*

More general method to identify the user is:

- A new IP indicates a new user.
- The same IP but different Web browsers, or different operating systems, in terms of type and version, means a new user.

The user identification process is used, to identify the user on the basis of above mentioned methods.

*4. Sessionization:*

This process is used as one of the time-oriented heuristic methods for session identification. In this system, session-duration based heuristic method is used for the sessionization. The session-duration-based method aims to set a session duration threshold. If the duration of a session exceeds a certain limit, it could be considered that there is another access session of the user. Discovered from empirical findings, a 30-min threshold for total session duration has been recommended [Magdalini Eirinaki,Michalis Vazirgiannis(2003)]. Result of session identification is sessions as shown in the figure 1.

*B. Pattern extraction component*:

The pattern extraction component consists of four modules.

*1. Recommender system based on clustering user sessions*

A cluster is a collection of objects that are similar to each other and are dissimilar to the objects belonging to other clusters. Clustering is the technique used to group together items that have similar characteristics.

The main task in the session clustering is to assign a weight to Web pages visited in a session. The weight needs to be well determined to analyze a user's interest in a Web page.

Let P be the set of Web pages accessed by user in Web server logs, P = {p1, p2...pm} each of which is uniquely represented by its URL. Let S be a set of user access sessions. S={s1,s2,…..sn}, Representation of each session is as vector model $s_j$= {w($p_1$,$s_j$), w($p_2$,$s_j$),….. w($p_m$, $s_j$)}, where w($p_i$, $s_j$) is weight assigned to the $i^{th}$ Web page in $j^{th}$ session. The w ($p_i$, $s_j$) needs to be determined to capture user interest in a Web page in the user session. Interest of a Web page is calculated by using frequency and duration. Frequency is the number of visits of a Web page and is given by Equation [Vlado Kesˇelj, Haibin Liu, (2007)],

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 3, Issue 1, January 2014*

$$\text{Frequency} = \frac{\text{NumberOfVisit(Page)}}{\sum_{\text{Pages} \in \text{VisitedPages}} (\text{NumberOfVisit(Page)})}$$

Duration is defined as the time spent on a page, i.e. the difference between the requested times of two adjacent entries in session. Duration is calculated as [Vlado Kesˇelj, Haibin Liu, (2007)],

$$\text{Duration(Page)} = \frac{\frac{\text{TotalDuration (Page)}}{\text{Length (Page)}}}{\max_{\text{Page} \in \text{VisitedPages}} \left(\frac{\text{TotalDuration (Page)}}{\text{Length (Page)}}\right)}$$

where Duration of a Web page is further normalized by the max ''Duration'' of pages in the session. System uses the average duration of the relevant session as "Duration" of last accessed Web page.

User's interest is always calculated with two strong indicators i.e. "Frequency" and "Duration". Interest degree of a Web page in the users is given by [Vlado Kesˇelj, Haibin Liu, (2007)],

$$\text{Interest(Page)} = \frac{2 \times \text{Frequency(Page)} \times \text{Duration(Page)}}{\text{Frequency(Page)} + \text{Duration(Page)}}$$

Every user access session is transformed into an m-dimensional vector of weights of Web pages, i.e. s= {w1, w2,,,, wm}, where m is the number of Web pages visited in all user access sessions.

To generate cluster vectorized sessions, K-means clustering algorithm is used. K-means is a prototype-based, simple partitional clustering technique which attempts to find a user-specified k number of clusters. These clusters are represented by their centroids (a cluster centroid is typically the mean of the points in the cluster).

The clustering process of K-means is as follows:

1. The algorithm is composed of the following steps:
2. Partition object into k non-empty subsets randomly.
3. Compute the centroids of the clusters
4. The set membership of the each object is decided by assigning that object to the nearest cluster centroid.
5. When all objects have been assigned, the value of the k centroids recalculated.
6. If none of the objects changed membership in the iteration, then generate final sets of the clusters otherwise repeat steps 3 and 4.

The usage pattern for each cluster is represented by the center of that cluster. The center of a cluster ct can be computed by calculating the mean vectors of the sessions assigned to the cluster :

$$\bar{\mu}_t = \langle w(p_1), w(p_2), , , w(p_n) \rangle$$

where w (pj) for cluster ct is given by

$$w(p_j) = \frac{1}{|c_t|} \sum_{s_i \in c_t} w\left(p_j, s_i\right)$$

In the recommendation step, the cosine similarity metric is used to find a similarity value sim ($\bar{s_a}$, $\bar{\mu}$) between each cluster center $\bar{\mu}$ and the active user session $\bar{s_a}$ given by,

$$\bar{s_a} = \langle w(p_1, s_a), w(p_2, s_a), , , w(p_n, s_a) \rangle$$

The best matching cluster is selected if that cluster has the highest similarity value, sim ($\bar{s_a}$, $\bar{\mu}$). A recommendation score is calculated by multiplying each weight in the cluster center vector by the similarity value of that cluster. The recommendation score of a page $p_i = p$ is calculated as follows

$$\text{rec}(\bar{s_a}, p_i) = \sqrt{w(p_i) \times \text{sim}(\bar{s_a}, \bar{\mu})}$$

In this way, recommendation score is generated for each page and the first k pages with the highest recommendation score are added to the recommendation set.

*2. Recommender system based on Click-Stream Tree*

This technique makes use of Click-Stream Tree to generate recommendations. This recommendation technique consists of four steps. The first step is data pre-processing step to The first step is data pre-processing step to identify unique users and user sessions The second step is to calculate the similarities between all pairs of sessions by using a similarity measure. In the third step, the sessions are clustered based on those similarities using the graph partitioning algorithm and last step is to build Click-Stream Tree for each cluster.

Pre-processing step is already discussed. Similarity measure is used for calculating the similarities between all pairs of sessions. The similarity between sessions is calculated such that only the identical matching of sequences has a similarity value 1.

Two terms, alignment score components and local similarity components are defined as two components of the similarity measure. The alignment score component computes how similar the two sessions are in the region of their overlap. If the highest value of the score matrix of two sessions, si and sj, is σ and the number of matching pages is M in the aligned sequence, then the alignment score component $s_a$ is:

$$s_a(s_i, s_j) = \frac{\sigma}{s_m * M}$$

The local similarity component computes how important the overlap region is. If the length of the aligned sequences is L, the local similarity components$_l$ is:

$$s_l(s_i, s_j) = \frac{M}{L}$$

Then the overall similarity between two sessions is given by

$$\text{Sim}(s_i, s_j) = s_a(s_i, s_j) * s_l(s_{i_i}, s_j)$$

The result of the previous step is pair-wise similarities of all user sessions. In this step, graph partitioning algorithm used to create the clusters. A graph is constructed whose vertices are user sessions. If the similarity value between $s_i$ and $s_j$ is greater than 0 then there will be an edge between two vertices ($s_i$, $s_j$) and this edge is weighted by this similarity value. The problem of clustering user sessions is formulated

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 3, Issue 1, January 2014*

as the problem of partitioning graph G into k disjoint sub graphs $G_m$, $(m \in [1, ..., k])$. Each disjoint sub graph represents a cluster. Each cluster contains the user sessions.

Click-Stream Tree (CST) is used to represent each unique user session in a cluster as a branch of a tree. As a result CST is generated for each cluster. Each CST has a root node, which is labeled as "null". Each node except the root node consists of two fields: data, count. Data field consists of page number. Count Field registers the number of sessions represented by the portion of the path arriving at that node. The child of each node in the CST is ordered in the count-descending. The Click-Stream Trees produced are used for the recommendation set generation.

Main goal of the recommendation set generation is to recommend the pages in least amount time. In On-line recommendation system, the speed of the recommendation engine is of huge significance. In order to reduce the search space, user sessions are clustered and represented by CST.

For the first two pages of the active user session, all clusters are searched to select the best path. For next request of the active user, top-N clusters that have higher recommendation scores among other clusters are selected for producing further recommendation sets. The last visited page used to build the data field. Model finds first node from the CST of a cluster that has same data field as the requested page number. Start with that node and go back until the root node (or until the active user session has no more pages to compare) to calculate the similarity of that path to the active user session. Calculate the similarity of the optimal alignment. To obtain the recommendation score of a path, the similarity is multiplied by the relative frequency of that path, which is defined as the count of the path divided by the total number of paths. The recommendation score is calculated for the paths which contain the data field in the cluster. The path has the highest recommendation score selected as the best path for generating the recommendation set for that cluster. The pages of child nodes are recommended to the active user.

*3. Recommender system based on association rule discovery:*

Association rules capture the relationships among items based on their patterns of co-occurrence across transactions. Association rules that reveal similarities between the Web pages derived from user behavior can be simply utilized in recommender systems. To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence. In Web Usage Mining the support is defined as follows. The Support for a page is the number of sessions that contain the page where as confidence of the association rule ($X \rightarrow Y$) is the conditional probability that a session having X also contains Y.

The system makes use of the Apriori algorithm to find the groups of pages occurring frequently together in many user sessions. The basic intuition is that, any subset of a large item set must be large. Therefore, the candidate item sets having k items can be generated by joining large item sets having k-1 items, and deleting those that contain any subset that is not large. This procedure results in generation of a much smaller number of candidate item sets.

Candidate item sets generated from the previous step are used as input for recommendation engine to make recommendation. System uses a fixed-size sliding window over the current active session to capture the current user's history depth. For example, if the current session (with a window size of 3) is $< A, B, C >$, and the user references the pageview D, then the new active session becomes $<B, C, D>$. The recommendation engine matches the current user session window with item sets to find candidate pageviews for giving recommendations. The recommendation value of each candidate pageview is based on the confidence of the corresponding association rule whose consequent is the singleton containing the pageview to be recommended. If the rule satisfies a specified confidence threshold requirement, then the candidate pageview is added to the recommendation set.

*4. Recommender system based on Markov model:*

Markov models are well-suited for modeling and predicting a user's browsing behavior on a Web-site . User's navigation behavior is mainly targeted by the Markov model. This denotes the input for Markov model is the User's navigation behavior i.e. user's sequentially accessed Web pages and the goal is to recommend the Web pages to the user. Three parameters are used to represent Markov model. i.e. $< A; S; T >$, where A denote the set of all probable actions that can be performed by the user; S denotes set of all probable states used to built Markov model; and T is a $|S| \times |A|$ Transition Probability Matrix (TPM), where each entry $t_{ij}$ corresponds to the probability of performing the action j when the process is in state i. Once the states of the Markov model have been identified, the transition probability matrix can be generated. Markov model uses the training set to generate the transition probability matrix. The transition probability matrix used to make prediction for Web sessions by only considering the user's previous action. The first k pages with the highest transition probability are added to the recommendation set.

*C. Hybridization Techniques*

The purpose of a hybridization block is to combine multiple recommender sets together to produce a single output. Hybridization process contains multiple techniques. These techniques are as follows

*1. Weighted*

A weighted Web recommender is the simplest design of hybrid recommenders in which the score of a recommended

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 3, Issue 1, January 2014*

item is computed from the results of all of the available recommendation techniques present in the system. Three phases namely training phase, candidate generation phase and scoring phase are used to generate final recommender set.

Each individual recommender processes the training data in the training phase. In the second phase, each module of the hybrid generates a candidate set consisting of k different pages. The pages in each candidate set is ordered such that an individual recommender thinks that the first page in the candidate set is most likely accessed next. In the scoring phase, this technique gets all the candidate sets generated by recommender systems. Items in the candidate set are weighted by each system and the final score is computed by a linear combination of the weights. Then Items are sorted by the combined score and the top k items are shown to the user.

*2. Hit-Ratio based mixed method*

A mixed hybrid presents recommendations of its different modules side-by-side in a combined list. However, the challenge of these types of hybrids is the integration of ranked pages in each recommendation set into the final recommendation set. Three phases same as previous technique are used to generate final recommender set.

Initially, in the training phase, training data are applied to each recommender system. In candidate set generation phase each recommendation module generates a candidate set consisting of k pages, on the basis of active session. The system assumes that each module generates uniformly accurate recommendations so that it assigns equal weight to every module. The system finds the best and worst modules according to their Hit-Ratio for the last page of the user session. Hit-Ratio is defined as follows: A hit is declared if any one of the four recommended pages is the next request of the user. The Hit-Ratio is the number of hits divided by the total number of recommendations made by the system. Then system selects the two best modules and combines the individual candidate sets to get a final recommendation set which consists of k pages.

*3. Switching*

Here the idea mentioned is that the modules may have not consistent performance for all types of the users. So that a switching hybrid selects a single recommender module from among its different modules based on a selection criterion. This selection criterion depends on the performance of the individual recommenders.

This hybridization technique follows same first two steps as of the previous technique. In the second step, each of the modules generates its individual candidate set. The system has decided one of the selecting criteria as "Length of user sessions". To produce the result, session length can be increased one by one. i.e. a user session with 4 pages, the number of pages in the user sessions is increased as 1, 2, 3 and 4. These results show that one of the recommended modules has better Hit-Ratio than the other recommendation module's Hit-Ratio. Hit-Ratio is defined different way as follows: A hit is declared if any one of the four recommended pages is the next request of the user. The Hit-Ratio is the number of hits divided by the total number of testing session. The switching hybrid selects one of the individual recommendation module based on this switching criterion. The recommendation sets generated by the selected module which set as the final recommendation sheet for the user.

*4. Frequency based Ranking*

The ranking hybrid first combines the individual recommendation sets of its modules into one recommendation set and then applies a ranking method to sort the pages in this set. First, each of the modules are the hybrid generated a recommendation set. The combined recommendation set is obtained by the union of the individual recommendation sets:

$$CRS = \bigcup_{i=1}^{4} RS_i$$

The system computes the scores for the pages by using the ranking method and on the basis of these scores the pages are ranked. The final recommendation set is generated from the first k pages and recommended to the active user.

The ranking method is also called as a Web page popularity method since this method assigns the score to every page on the Web site that reflects its popularity. The score of each page depends on the total number of visits on that page. The Score of the page is defined as the ratio of the total number of visits on the page and number of pages. Once the scores are computed rank is assigned to each page and the final recommendation set is generated from top k rankers.

## V. EXPERIMENTS

For experiments, Synthetic dataset for dktes.com (SDD)1 and hyperreal.org (SDH)2 are used. Log data of dktes.com and hyperreal.org site is present in extended log format which is supported by Microsoft Internet Information Server (IIS).

Total 16693 log entries from SDH dataset and 284187 log entries from SDD dataset are processed for the system. In the data cleaning step, first the irrelevant log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG are eliminated and all the log files are cleaned.

Table I presents some statistics of the preprocessed experimental dataset, including both training and testing sets.

[1] http://www.dktes.com/
[2] http://www.hyperreal.com/

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 3, Issue 1, January 2014*

TABLE I
Statistics of experimental dataset

| Attributes | SDD | SDH |
|---|---|---|
| Total access entries | 284187 | 16693 |
| Clean access entries | 55883 | 8968 |
| Different access users | 10000 | 1979 |
| Accessed web pages | 895 | 876 |
| Identified sessions | 1491 | 996 |
| Sessions for Training | 1151 | 754 |
| Sessions for Testing | 340 | 242 |

Clusters are created by using K-Means algorithm. WEKA machine learning tool is used to implement this clustering method. For the SDD, total nine clusters whereas for SDH total six clusters are created.

Association rules are generated by using the Apriori Algorithm. Apriori Algorithm available in WEKA machine learning tool is used. Total 60,000 rules for SDH and 25000 rules for SDD are generated. Following table shows sample of generated rules by Apriori algorithm.

A set of experiments are conducted with all of recommender systems. Figure 2 shows the results of these experiments as the Hit-Ratio of each recommender system. As shown in the graph, Clustering and Association Rule (AR) are having less Hit-Ratio compared with Markov Model (MM) and Click-Stream Tree Model (CST). The reason for this could be that recommender systems that consider the order of visiting pages have a better performance compared with the other models that represent user sessions in a different way (e.g., time spent on page or co-occurred pages).

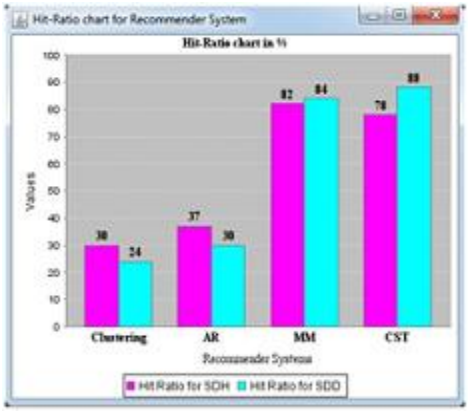


Fig. 2 Hit-Ratios in % for Recommender Systems.

The Hit-Ratio for the hybrid recommenders is shown in figure 3.

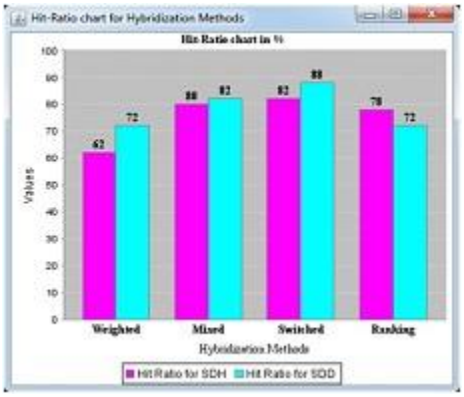


Fig. 3 Hit-Ratios in % for hybridization methods.

As can be seen from the figure 3, the switched hybridization method has highest Hit-Ratio whereas weighted method has lowest hit ratio among the Hybridization methods. These results also show that recommendation accuracy is directly proportional to the switching criteria. The aim of this paper is to examine the effects of the hybrid recommenders. For this reason system is used to take the results of the hybrid recommender against the results of its modules.

## VI. Conclusion

By analyzing the results of the hybridization methods, following conclusions are drawn:

1. The hybrid recommender provides successful recommendation when the recommended page is generated by all the modules of the hybrid.
2. To increase performance of the hybrid recommender system, choice of hybridization method is crucial.
3. Comparison between Hit-Ratio of Recommender systems and Hit-Ratio of Hybridization methods shows that there is a correlation between performance of the modules and the performance of the hybrid recommender methods. Any improvement of the Hit-Ratio of the modules will also have a positive impact on the performance of the hybrid recommender that uses these modules.

## REFERENCES

[1] Agrawal R., & Srikant R., "Fast algorithms for mining association rules," in *J. B. Bocca, M. Jarke, & C. Zaniolo (Eds.), Proceedings of the 20th international conference on very large data bases, VLDB,* 1994, pp. 487–499.

[2] Agrawal R., Swami A, Imieliński T., "Mining association rules between sets of items in large databases", *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*, 1993, p. 207.

[3] Barragáns-Martínez A. B., Costa-Montenegro E., Burguillo J. C., Rey-López, M., Mikic-Fonte, F. A., & Peleteiro, A. "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition." *Information Sciences, 180(22),* 2010 p. 4290–4311.

[4] Burke R. "Hybrid recommender systems: Survey and experiments". *User Modeling and User-Adapted Interaction, 12(4),* 2007, 331–370.

[5] Deshpande M., & Karypis G., "Selective Markov models for predicting Web page accesses," *ACM Transactions on Internet Technology (TOIT),* 4(2), 2004, 163–184.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

*International Journal of Advanced Research in Computer and Communication Engineering*
*Vol. 3, Issue 1, January 2014*

[6] Gündüz S. & Özsu M. T., "A Web page prediction model based on Click-Stream Tree representation of user behavior", in *Proceedings of 9th ACM international conference on knowledge discovery and data mining (KDD),Washington, DC, USA*, August, 2003.

[7] Magdalini Eirinaki and Michalis Vazirgiannis, "Web Mining for Web Personalization" *Communications of the ACM*, vol. 3, No. 1, Feb. 2003 pp.2-21.

[8] Mobasher B., Dai H., Luo T., & Nakagawa M., "Effective personalization based on association rule discovery from Web usage data" in *Web information and data management*, 2001, pp. 9–15.

[9] Tao Luo, Bamshad Mobasher, Honghua Dai, Miki Nakagawa, "Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization," *Data Mining and Knowledge Discovery 6(1),* 2002, p.61–82.

[10] Vlado Kes˘elj, Haibin Liu "Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests". *Data & Knowledge Engineering 61*, 2007, 304–330.

[11] Wang Q., Makaro D. J., and Edwards H. K., "Characterizing customer groups for an e-commerce website," *EC'04, USA*, 2004, p. 218-227.

.