Journal of Universal Computer Science, vol. 19, no. 15 (2013), 2224-2240
submitted: 15/5/13, accepted: 30/8/13, appeared: 1/9/13 © *J.UCS*

A Personalized Recommender System Based on a Hybrid Model

Wedad Hussein
(Ain Shams University, Faculty of Computer and Information Sciences, Cairo, Egypt
wedad.hussein@fcis.asu.edu.eg)

Rasha M. Ismail
(Ain Shams University, Faculty of Computer and Information Sciences, Cairo, Egypt
rashaismail@fcis.asu.edu.eg)

Tarek F. Gharib
(King Abdulaziz University, Faculty of Computing and Information Technology, Jeddah
Saudi Arabia
Ain Shams University, Faculty of Computer and Information Sciences, Cairo, Egypt
tfgharib@kau.edu.sa)

Mostafa G. M. Mostafa
(Ain Shams University, Faculty of Computer and Information Sciences, Cairo, Egypt
mgmostafa@cis.asu.edu.eg)

Abstract: Recommender systems are means for web personalization and tailoring the browsing experience to the users' specific needs. There are two categories of recommender systems; memory-based and model-based systems. In this paper we propose a personalized recommender system for the next page prediction that is based on a hybrid model from both categories. The generalized patterns generated by a model based techniques are tailored to specific users by integrating user profiles generated from the traditional memory-based system's user-item matrix. The suggested system offered a significant improvement in prediction speed over traditional model-based usage mining systems, while also offering an average improvement in the system accuracy and system precision by 0.27% and 2.35%, respectively.

Keywords: Recommender Systems, Web Usage Mining, Next Page Prediction
Categories: I.5.1, I.5.3, I.5.4, L.2.2

1 Introduction

Web personalization could be defined as the process of tailoring a web site to the needs and preferences of specific users. Given the huge amount of information available on the World Wide Web it became very important to interact with the user, understand his behavior and be one step ahead of him. Next-Page prediction techniques make use of the information stored in Web server logs to build a model of users' behavior and these models are used to anticipate the user's next page based on his profile.

Next page prediction improves on the friendliness of a web site. It also reduces network latency by prefetching required pages. Also these prediction techniques are essential for e-Commerce applications to recommend suitable content and offer personalized advertisements.

Recommender systems take advantage of the preferences of a group of users to make individual recommendations. They help users locate interesting objects among a huge set of available objects. Web-based recommender systems are important tools for locating information and for websites to recommend to their users products or services that meet their preferences.

There are two main approaches to recommender systems, memory-based (also known as nearest neighbor) methods and model-based methods [Kumar 09].

Memory based recommender systems store all ratings or opinions of all users and generalize from them at the time of making recommendations [Pham 11]. The techniques used by memory-based recommender systems allow for recommendations that are tailored to the needs of each individual user, however, the size of data that needs to be stored affects their scalability.

Model-based methods use data mining techniques to develop a model of user behavior [Sandvig 08]. Examples of these methods include Bayesian network analysis, rule based approaches and association analysis. Web mining builds models based mainly on record user behavior as opposed to subjective ratings.

In Web Mining, model-based techniques generate recommendations based on the general browsing behavior of all users and they treat users anonymously. The only information held about the user is his current session, and if two users share the same path the recommendations presented to both of them would be exactly the same disregarding any previous visits of the users.

In this paper we propose a web usage mining system for the next page prediction that integrates some of the techniques used in memory-based recommender systems. Most of the computationally intensive processes are performed offline. We suggest the use of clustering to group individual user profiles and, accordingly, frequent patterns. When predicting the next page reference the system compares the current user path only to patterns belonging to the same cluster.

The remainder of this paper is organized as follows: Section 2 offers an overview and comparison of both categories of recommender systems. Section 3 introduces the suggested system while section 4 presents experiment design and system evaluation. Section 5 contains the conclusions.

2 Background

2.1 Memory-Based Recommender Systems

Memory-based systems (user-based or item-based) are based on the fact that users often like the items which are preferred by other users who have agreed with them in the past. They use the entire user-item rating database to generate recommendations [Pham 11]. Memory-based methods can be classified into three groups: Collaborative filtering, content-based techniques and hybrid techniques.

Generally, *user-based collaborative filtering* techniques search for the “Neighborhood” of the user; that is the group of users exhibiting similar behavior to the current user. To achieve this, the system builds a user-items matrix containing the ratings of users to all items whenever available.

User-based Collaborative filtering methods could be used for one of two purposes [Vozalis 03]:

- Prediction: Generates a value indicating the expected rate of an item of the current user.
- Recommendation: Produces a list of N items that the user is expected to like (Top N recommendations).

Collaborative filtering suffers from a set of problems. One of the most prominent problems with these systems is the "Cold Start" problem. Cold start could refer to both recommending newly added items or making recommendations for new users [Rashid 08]. [Park 09] suggests a predictive feature-based regression model that makes use of all the information available on users and items like demographic information and item content data to overcome the cold start problem. The integration of other sources of information was also proposed in [Castillejo 12] which suggested the integration of information from social networks into the recommendation process.

In [Rosaci 13] a multi-agent recommender system was suggested for recommending multimedia content. The multi-agents were intended for collecting user profiles over multiple sites, thus overcoming the new user problem and also for handling the same user accessing content using different devices.

Another problem with Collaborative filtering is the sparsity of the user-item matrix. Users only rate or view a very small portion of all the items available. The similarity between users (or items) is often derived from few overlapping ratings and it is hence a noisy and unreliable.

In [Shinde 12] clustering was applied to the user-item matrix and the active user profile were compared to cluster centroids as opposed to individual users and the prediction is made based on the joint opinions of the users in the cluster. This approach reduces the effect of sparsity on the recommendation process while also dealing with the scalability problem. The collaborative filtering algorithm can be very extensive and they grow non-linearly as the number of items and users grow.

Content-based recommender systems do not use ratings but, instead, they compare the content of items to the user profile. This is usually done using vector-space model [Ruotsalo 10]. Content-based recommender systems can overcome the “cold start” problem, that new items for which little or no user ratings are available, but they are generally less accurate than Collaborative filtering systems [Gunawardana 09]. Content-based systems can recommend new items to users based on their features, on the other hand they only recommend items that are similar to what the user has viewed before which is referred to as "over specialization" [Park 09].

Hybrid recommender systems have been proposed to overcome some of the previous problems by combining techniques from both Collaborative Filtering and Content-based filtering [Gunawardana 09]. There are different ways for combing content and collaborative filtering. One method is to generate recommendations from both techniques separately and later combining the recommendation lists. Another method is to incorporate content information into the data collected by collaborative filtering systems [Shinde 11].

[Kardan 13] suggested a hybrid recommender system targeted for asynchronous discussion groups that consists of three parts. The first part is a collaborative filtering part that uses association rules to find relations between posts that a user likes, then a content-based section extends the user profile with content information. Finally, the hybrid filtering section uses rules generated by collaborative filtering to choose among the recommendations of content-based filtering.

[Kazienko 06] offered a comparison of the different approaches to recommender system and which of the previously mentioned problems did each solve.

2.2 Model-Based Recommender Systems

Model-based methods use data mining techniques to build models for user ratings. Web mining builds models based mainly on user behavior rather than subjective ratings. These models could be built using a variety of techniques, we next explore some of the related techniques.

Clustering can be applied at different stages of the mining process and it also could be applied on different components like access patterns or web pages. Nadi et al [Nadi 11] proposed a method that uses both document and user clustering. Classic TF-IDF method was used to cluster web documents and an access matrix was used to reflect user accesses to document clusters. Clustering was again applied in the matrix to group similar users. In [Anitha 10] Clustering was applied to transactions generated from web server logs then Markov Model was used to predict the user’s next page. Clustering was also applied to transactions in [Jalali 08] but the transactions were represented as a graph and a graph partitioning algorithm was applied to find groups of strongly correlated pages.

User sessions are best modeled as sequences of page accesses. *Frequent sequences* are mined from server logs and the user’s current path is also modeled as a sequence pages. In [Jalali 09] they suggested user classification using longest common subsequence. The navigation patterns are generated offline based on the work in [Jalali 08]. The classification algorithm finds the navigation pattern with the highest degree of similarity to the active user’s session. The recommended list is ranked in term connectivity between pages in the adjacency matrix.

The integration of semantic information into the different stages of the web mining process has been suggested to enhance the overall performance of the system and give meaningful recommendations [Hassanzadeh 12, Babu 12]. In [Thakur 12] the user's current path is matched with semantic annotated navigational patterns to generate recommendations. [Mabroukeh 09] used semantic distance for candidate pruning during frequent navigational pattern generation process.

2.3 Memory-Based vs. Model-Based techniques

Memory-based methods, as mentioned before, suffer from scalability issues. On the other hand, Model-based methods perform the computationally intensive model building offline which makes them scale better [Asjana 12].

Since the learning process of Memory-based techniques are performed online, these methods adapt quickly to changes in the users’ interests. But, for Model-based techniques the learning process needs to be incremental or the model should be re-built periodically to accommodate new data.

The most serious problem with Memory-Based approaches are the sparsity of user-item rating matrix where each user only rates a small set of items. The similarity between users (or items) is often derived from few overlapping ratings and it is hence a noisy and unreliable.

As mentioned above the new user problem is a recurring problem in memory-based collaborative recommender systems. It occurs when a new user is added to the system and there is not enough information making a good selection of the user's neighbors. As a consequence, the recommended items have a poor correlation with the user's interests [Tkalčič 11]. This problem is not present in model-based systems because, in these systems, a general model is built and is applied later to all users.

3 The Proposed Hybrid Personalized Recommender System

In this paper we suggest a recommendation system that relies on model-based techniques for generating recommendations, but integrates memory-based techniques to direct the prediction process. The prediction of the next page is done by comparing the active user path to the set of frequent association patterns. On the other hand, the user-item matrix is clustered and the association patterns are clustered accordingly. When the prediction is made the search is restricted to patterns that are assigned to the current user's cluster.

The system works over two phases, an offline phase and an online phase. Figure 1 shows the architecture of the suggested system.

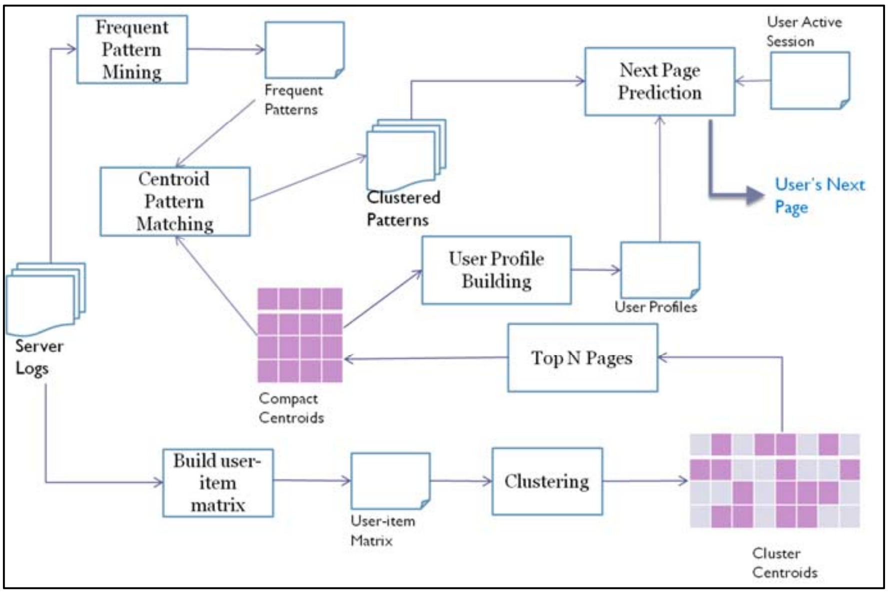


Figure 1: The Proposed Hybrid Personalized Recommender System Architecture

3.1 Offline Phase

The offline phase involves the preparation of the data items later needed to perform online recommendation. It involves preprocessing, mining frequent usage patterns and the building of the user-item matrix for the memory-based part of the suggested system.

Web server logs store the history of user requests and are the main data sources for web usage mining. Web servers usually collect the following information about a single page reference: the user accessing the page, access time, request method ("GET" or "POST"), the URL of the required page, transmission protocol, return code and the number of bytes transmitted. Entries in a web server log are sorted chronologically. Web usage mining focuses on the user, access time and URL fields. Some preprocessing needs to be performed on raw server logs like the removal of image references, access errors and robot references.

For the representation of *frequent access patterns*, we chose to use Web association patterns. We adopted the process used in [Karam 06] for the generation of frequent patterns. The paper uses a 30 minute timeout for the separation of user sessions. Maximal Forward References (MFR) [Chen 96] transform user sessions into transactions suitable for mining frequent association patterns. A Maximal Forward Reference is defined as the chain of references from the first page in a user's session until a backward reference is encountered. The transaction database consists of the set of all MFRs generated from user sessions. Finally, the Apriori Algorithm for association rule mining was applied to the transaction database to generate frequent patterns.

The *user-item matrix* suggested represents the access history and interests of users. The rows in the matrix represent users, while the columns represent web pages. The entry in the matrix is the average time spent by a specific user on a specific page; a zero entry indicates that the user did not visit the page.

Clustering is then applied in the generated matrix to obtain generalized access patterns. The generated user-item matrix is very sparse. To overcome the problems associated with sparsity, we used Singular Value Decomposition (SVD) to transform the matrix before clustering.

The feature vector we use is the user profile, the features are the pages of the web site and the values are the average time spent by the user on the page. We used k-Means as our clustering algorithm and using cosine similarity for the distance measure. The cosine similarity between two vectors A and B is given in equation (1).

$$\text{Cosine Sim} = \frac{\mathrm{A.B}}{\|\mathrm{A}\|\|B\|} \qquad (1)$$

where here $\|X\| = \sqrt{\sum_{i=1}^{n} (X_i)^2}$

The k-Means algorithm starts with a set of seeds, referred to as cluster "centroids". In every iteration of the algorithm, each point in the data is the assigned to the nearest cluster and centroids are recalculated as the average of all points in the cluster. The process continues until no changes in the clusters occur. In our work the initial centroids were selected randomly from the set of users.

```
Algorithm: Offline processing of the hybrid recommender system
Inputs: Web Server Logs L, Set of Frequent Patterns FP
Outputs:user-item matrix M, Cluster Centroids C, Clustered Patterns
Begin
From L Calculate Average time spent, Group By user, page
For each user u_i
        For each page p_j
                If average time spent exists
                        M[u_i, p_j] = average time spent
                else    M[u_i, p_j] =0
C=KMeans_Clustering(M)
For each cluster centroidc_k in C
        sort pages in c_k
        TopN [k] = Top N pages in c_k based on average time spent.
For each useru_i
        sort pages in vector M[u_i]
        For each page p_j in M[u_i]
                IfM[u_i,p_j] ≠ 0 andp_j ∉ TopN[c_ui]
                        Add p_j to user profile
For each frequent Pattern P_i
        For each cluster c_j
                if similarity(P_i, TopN [c_j]) > threshold
                        Add Pi to Patterns [c_j]
End
```

Figure 2: Steps for the Offline Phase of the Suggested System

After obtaining the clustering results and calculating the cluster centroids we only saved a compressed version of the centroids. That is, we kept only the top N pages as determined by the average time spent on the page. These sets of pages are later on considered to be the representatives of the clusters.

After performing clustering, the frequent patterns are assigned to cluster centroids based on the similarity calculated by equation (2). Let C_i be the set of pages in the compressed centroid of the cluster i, and Pj be the set of pages in the currently tested frequent pattern, the similarity is calculated as:

$$Sim(C_i, P_j) = \frac{|C_i \cap P_j|}{|P_j|} \qquad (2)$$

If the similarity exceeds a threshold t (set to 40% in this paper), the pattern is assigned to cluster C_i. This definition implies that the resulting clusters of patterns are not mutually exclusive.

The 40% threshold was chosen according to experimental results. When the threshold was set to 30%, each cluster of patterns contained on average 77% of the original patterns which meant that the number of comparisons needed to make a prediction were close to the number of comparisons needed without clustering. Therefore, clustering had very small effect on prediction time.

When the threshold was set to 50%, the cluster of patterns contained on average 53% of the original set of patterns. This percentage caused that, when making a prediction, the system could go through the whole list of patterns assigned to the user's cluster and not find a pattern that matches the current user path. Accordingly when using 50% threshold the system failed to generate predictions for 9% of the user paths as opposed to 4% failure when using 40% threshold.

Finally, *user profile* was suggested to represent the deviation of each user from the whole set of users in the same cluster. The user profile is represented by the top M pages, in the user’s whole visit history, that are not present in the compressed centroids. The profiles are used to generate more individualized recommendations. Figure 2 shows a detailed pseudo-code of the steps included in the offline phase.

3.2 Online Phase

Given a user session that consists of a sequence of web page references $S=\{r_1,r_2,\ldots,r_n\}$ we seek to find a prediction process P such that when provided with a prefix S’ of the user session, the system provides a prediction $r_p=P(S')$, where r_p is the page the user is expected to visit after visiting all pages in S’. The prediction is made by searching for the frequent pattern with the Longest Common Subsequence (LCS) of S’. This process generates the “recommendation set”, but only one page from this set is presented to the user as the one he/she is expected to visit next. In this paper we compare three variations to how the next page is selected from the recommendation set. The algorithm for next page prediction is described in Fig. 3.

Case (I): This is the process adopted by most models-based recommender systems that use frequent sequences and Longest Common Subsequence to generate recommendations [Jalali 08, Jalali 09, and Sneha 12]. The active user session is matched against the complete set of frequent patterns generated from the dataset. After generating the recommendation set, the page with the highest support is selected.

Case(ii): If the current user is a known user (a user represented in the user-item matrix and already clustered) the active user session is matched against the set of patterns assigned to his cluster. Else, the current user is assigned to the nearest cluster centroid and the same process is applied. After generating the recommendation set, the pages are sorted in descending order according to support. The system then selects the page with the highest support that belongs to the compressed centroid of the user cluster. If the recommendation set does not contain pages from the user’s cluster the page with the highest support in the whole list is selected.

Case(iii): In this case we integrate the suggested user profile into the prediction process. The same process explained in the previous section is applied but higher priority is given first to pages that belong to the user’s profile.

Algorithm: Predict Next Page
Inputs: User U, User Current path UP, Clustered Frequent Patterns FP, Cluster centroids Cent
Outputs: Predicted Next Page
1. **Begin**
2. Support=0, clust_sup=0, user_sup=0
3. **If** U $\in$ Known Users
4. C_u = Cluster(U)   \\C_u is the user's cluster
5. **Else** C_u =NearestCluster(U, Cent)
6. **For each** pattern P in the set of frequent patterns assigned to cluster C_u
7.   **If** pattern length = (user's path length)+1
8.   **If** pattern pages (p_1,...,p_{n-1}) = user path
9.   **If** support(P)>Support
10.   prediction=p_n
11.   Support = support(P)
12.   **If** $p_n \in$ Centroid(C_u)
13.   If support(P)>clust_sup
14.   clust_prediction=p_n
15.   clust_sup= support(P)
16.   **If** $p_n \in$ Profile(U)
13.   If support(P)>user_sup
14.   user_prediction=p_n
15.   user_sup= support(P)
16. **If** Prediction Method = "With User Profile"
17.   **If** user_prediction is not empty
18.   **return** user_prediction
19.   **Else If** clust_prediction is not empty
20.   **return** clust_prediction
21. **If** Prediction Method = "With Clustering"
22.   **If** clust_prediction is not empty
23.   **return** clust_prediction
24. **return** prediction
25. **End**

Figure 3: Algorithm for Next Page Prediction.

4 Results and Discussion

4.1 Datasets

Our experiments were conducted on logs from two different Web servers, logs from the University of Saskatchewan's (USAK) and from the NASA web servers. Table 1 contains the statistics for the datasets. Training data used for building the model (User-item matrix and mining frequent patterns), while the test data were used for evaluating the prediction process.

The number of clusters used throughout the experiments was fixed at 20 clusters which was the most appropriate number as indicated by the calculation of the silhouette coefficient. As mentioned in section 3.1, to create the compressed cluster centroids, we keep only the top N pages in each cluster centroid. All values recorded in the experiments are averaged over 3 different values of N (200, 300, and 400).

Dataset	**USAK**		**NASA**	
	Training	**Test**	**Training**	**Test**
Starting Date	Jun. 1, 1995	Sep. 1, 1995	Jul. 1, 1995	Aug. 1, 1995
End Date	Aug. 31, 1995	Sep 30, 1995	Jul. 31, 1995	Aug. 14, 1995
Total Recording Period	3 months	1 month	1month	2 weeks
Requests	353,072	158,465	665,017	240,969
Sessions	93,395	47,388	149,598	51,916

Table 1: Datasets Description

4.2 Prediction Speed

We used two parameters to evaluate prediction speed: the average number of frequent patterns that the system has to go through to make a prediction and the average time needed to make a single prediction. We only recorded these values for cases (i) and (ii) discussed in section 3.2, since the case (iii) accesses the same set of patterns as (II). We recorded the values for 4 different values of minimum support.

Using clustering showed a 37.6% and 13.6% reduction in the number of patterns tested for the USAK and the NASA data sets respectively. The reduction in prediction time was 44.4% and 22.2%. Table 2 and Figures 4 and 5 show the comparison results. The reduction in both measures occurred because, when using clustering, the system searches in only a subset of the frequent patterns (the subset assigned to the user's cluster).

It could also be noted from the results that the lower support threshold is, the higher the improvement in prediction speed. This occurs because as the support threshold is reduced, the number of frequent patterns is higher, and therefore the effect of the clustering is more evident. Also it can be seen that the reduction in the predicted time is higher than the reduction in the number of patterns tested, because of the overhead associated with the access and processing of large files.

Data Set	USAK		NASA	
Parameter	Patterns Accessed / Query	Prediction Time (ms.)	Patterns Accessed / Query	Prediction Time (ms.)
Without Clustering	1901.4	1.8	2880.8	2.7
With Clustering	1186.5	1	2489.6	2.1
Reduction %	37.6	44.4	13.6	22.2

Table 2: Comparison of Patterns Accessed / Query

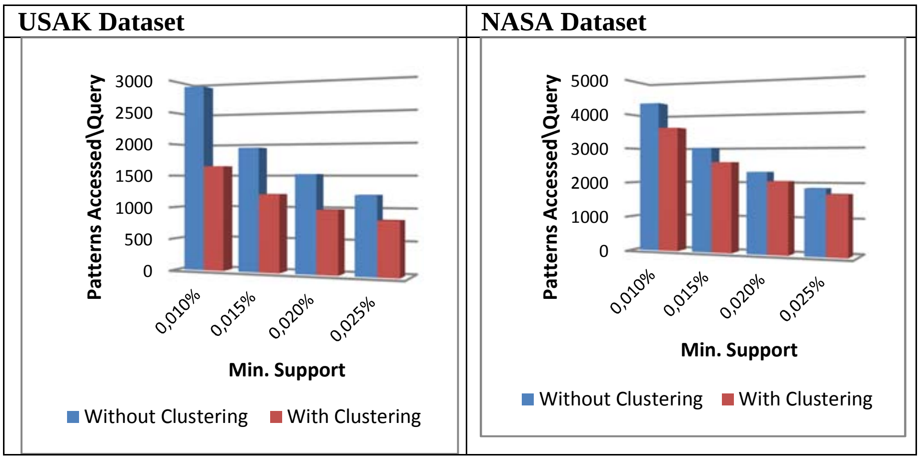


Figure 4: Comparison of Patterns Accessed / Query

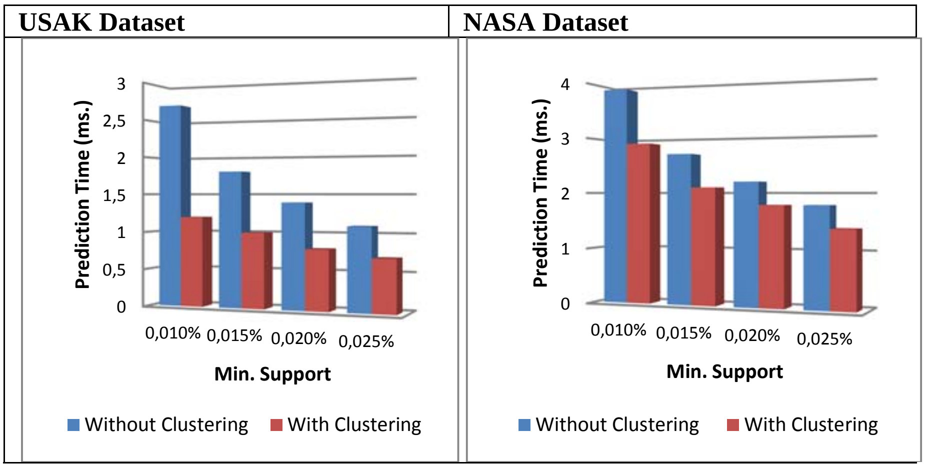


Figure 5: Comparison of prediction time

4.3 Prediction Quality

We used four measures to evaluate the quality of prediction; accuracy, precision, coverage and F1 measure. The results of the evaluation are represented in table 3 and figures 6, 7, 8 and 9.

The accuracy is defined by equation (3). To calculate the number of correct recommendations we use the following definition: given a sequence of page references $S=\{r_1,\ldots,r_k,r_{k+1}\}$ a prediction P is considered correct if $P(r_1,\ldots,r_k)=r_{k+1}$. The results (Figure 6) show that the use of clustering did not reduce the accuracy of the system, even though the system only looks at a portion of the frequent patterns. Actually, the system showed a slight improvement in the accuracy which means that the portion of patterns assigned to a cluster reflects the behavior of its users better. The introduction of user profiles did not affect the accuracy of the NASA dataset, but it improved it even further for the USAK dataset. This happened because the recording period for the USAK training dataset is longer and therefore the capturing of user behavior is more accurate.

	Dataset	**USAK**	**NASA**
Accuracy	Without Clustering	25.2	19.2
	With Clustering	25.21	19.3
	Change%	+0.04	+0.5
	With user Profile	25.3	19.3
	Change%	+0.4	+0.5
Precision	Without Clustering	0.328	0.215
	With Clustering	0.336	0.22
	Change%	+2.4	+2.3
Coverage	Without Clustering	0.289	0.324
	With Clustering	0.276	0.317
	Change%	-4.5	-2.2
F1	Without Clustering	0.307	0.258
	With Clustering	0.303	0.259
	Change%	-1.3	+0.4

Table 3: Evaluation of the Quality of Prediction

$$Accuracy = \frac{\#\text{of correct predictions}}{\text{total}\#\text{ of predictions}}\% \qquad (3)$$

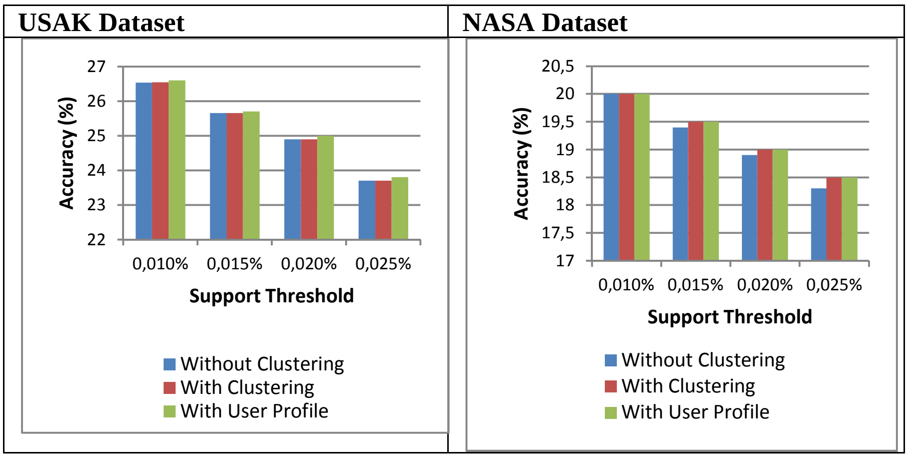


Figure 6: Comparison of Accuracy values

We adopted the definitions of Precision, Coverage and F1 measure used in [AlMurtadha 11] as shown in equations 4, 5, and 6. Where R is the recommendation set, A is the Active user session and *w* is the set of pages in A that the user already visited.

$$precision(R, A) = \frac{|R \cap (A - w)|}{|R|} \tag{4}$$

$$Coverage(R, A) = \frac{|R \cap (A - w)|}{|(A - w)|} \tag{5}$$

$$F1(R, A) = \frac{2 \times precision(R, A) \times Coverage(R, A)}{precision(R, A) + Coverage(R, A)} \tag{6}$$

Again here we recorded the precision, coverage and F1 measure value for only in the original system (without clustering) and after clustering since introducing user profiles does not change the recommendation set, it only changes the page predicted to be the user's next page.

As can be seen from the results (Figure 7), the precision values increase with the use of clustering. Precision measures the number of correct relevant recommendations to the total recommendation set. Given the accuracy values, the number of correct relevant recommendations for both cases is very close but the size of the recommendation set is less in the case of using clusters therefore the values of precision are higher in this case.

On the other hand, since the size of the recommendation sets decreases in the case of clustering, it is likely that some relevant pages are missing from the recommendation set and therefore the value of (|R∩(A-*w*)|) decreases with the use of clustering causing the value of coverage to decrease (Figure 8).

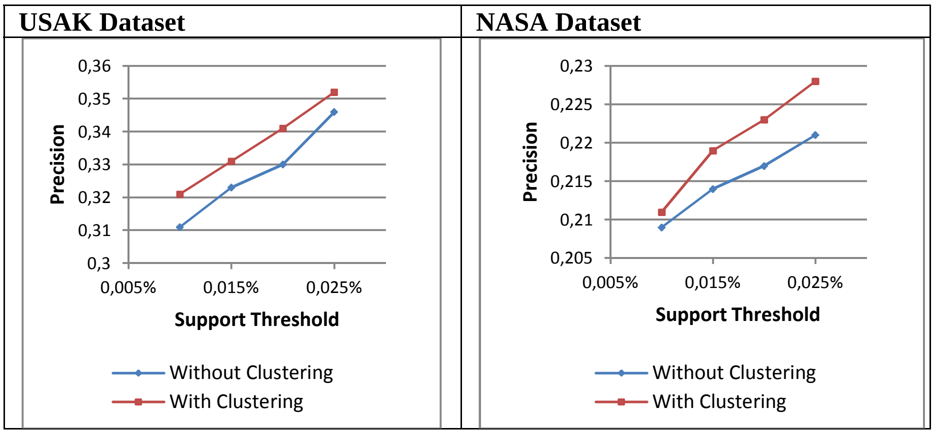


Figure 7: Comparison of Precision values

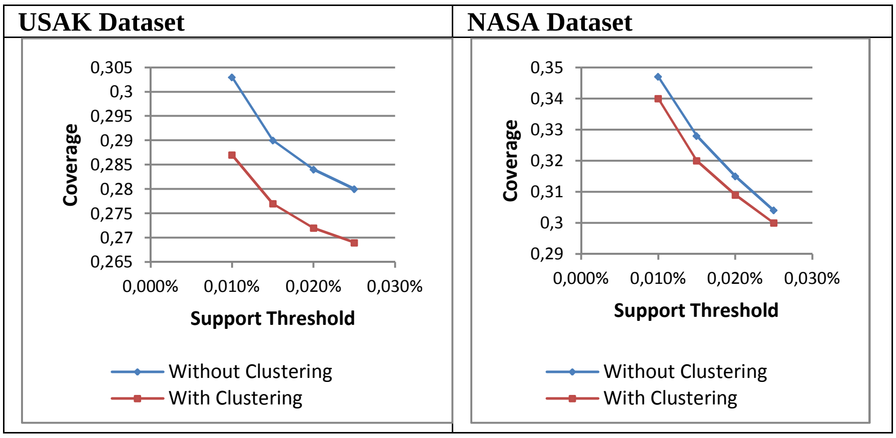


Figure 8: Comparison of Coverage values

Coverage is the ratio between the number of relevant Web pages retrieved and the total number of web pages that actually belongs to the user session. The decrease in the values of coverage is higher in the case of USAK dataset and therefore it affects the values of F1 measure which also decreases slightly with the use of clustering. This does not happen for the NASA dataset since the decrease in the coverage value is only 2.2%. The F1 measure attains its maximum value when both accuracy and coverage are maximized.

Also we could see that, as the value of minimum support increases, the values of precision increase. This is due to the fact that the increase in the minimum support prunes more patterns and therefore the size of the recommendation set become smaller. On the other hand the values of Coverage decrease with the increase in the support threshold because the size of the recommendation set decreases and therefore the value of (|R∩(A-*w*)|) decreases as well.

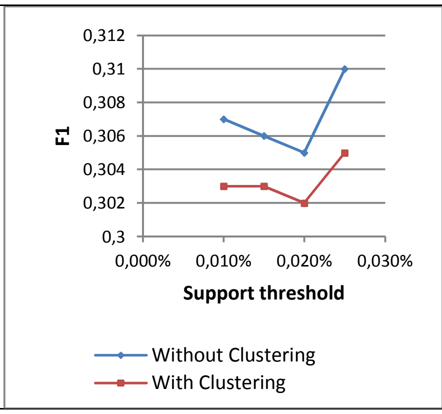
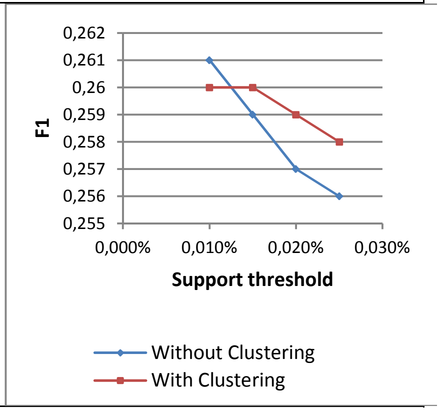
USAK Dataset	NASA Dataset
	

Figure 9: Comparison of F1 Measure values

5 Conclusions

Recommender systems are built for the purpose of predicting user behavior and recommending suitable items. There are two main techniques for recommender systems; memory-based and model-based. Memory-based methods have problems with dealing with new users as well as scalability issues, while model-based techniques do not offer individualized tailored recommendations. In this paper we introduced a hybrid recommender system for the next page prediction that integrates model-based and memory-based recommendation techniques. The system takes advantage of the user-item matrix used in Memory based methods to direct the model-based methods in generating individualized recommendations. Clustering is performed on the user-item matrix and user profiles are generated accordingly. Frequent browsing patterns generated from Model-based methods are also clustered in accordance with the results of the user-item matrix clustering. When making a prediction, the system looks at a portion of the frequent patterns guided by the clustering results. We evaluated the prediction speed in terms of the number of patterns that the system needs to go through to make a single prediction and the time needed to make the prediction. The suggested system showed a 25.6% and a 33.3% average improvement in the number of patterns accessed and prediction time respectively over traditional model-based system. This improvement happened while offering a 0.27% and a 2.35% improvement in the average accuracy and precision of the system respectively. This means that the reduction in prediction time did not compromise the prediction quality.

References

[AlMurtadha 11] AlMurtadha, Y., Bin Sulaiman, N., Mustapha, N.,Udzir, N.I.: IPACT: Improved Web Page Recommendation System Using Profile Aggregation Based On Clustering of Transactions, In American Journal of Applied Sciences 8 (3), 2011, 277-283.

[Anitha 10] Anitha, A.: A New Web Usage Mining Approach for Next Page Access Prediction, International Journal of Computer Applications, 2010, 7-10.

[Asjana 12] Asjana, M.,López,V. F., Muñoz,M. D., Moreno,M. N.: Semantic Web Mining for Book Recommendation, In *Advances* in Intelligent Systems and Computing Vol 171, 2012, 101-109.

[Babu 12] Babu, K.G., Komali, A., Mythry, V., Ratnam, A.S.K.: Web Mining using Semantic Data Mining Techniques, International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-3, July 2012, 168-171.

[Castillejo 12] Castillejo, E., Almeida, A., López-de-Ipiña, D.: Alleviating cold-user start problem with users' social network data in recommendation systems, Preference Learning: Problems and Applications in AI (PL-12) at ECAI 2012, Montpellier, France, August 2012.

[Chen 96] Chen, M.S., Park, J.S., Yu, P.S.: Data mining for path traversal patterns in a web environment, In Proceedings of the 16th International Conference on Distributed Computing Systems, 1996, 385-392.

[Gunawardana 09] Gunawardana, A., Meek, C.: A Unified Approach to Building Hybrid Recommender Systems, In Proceedings of the third ACM conference on Recommender systems (RecSys 09),2009, 117-24.

[Hassanzadeh 12] Hassanzadeh, H., Keyvanpour, M.R.: Semantic Web Requirements through Web Mining Techniques, International Journal of Computer Theory and Engineering, Vol. 4, No. 4, August 2012, 616-620.

[Jalali 09] Jalali, M., Mustapha,N.,Sulaiman, M. N., Mamat,A.: A Recommender System Approach for Classifying User Navigation Patter ns Using Longest Common Subsequence Algorithm, In American Journal of Scientific Research , 2009, 17-27.

[Jalali 08] Jalali, M., Mustapha,N.,Sulaiman, M. N., Mamat,A.: Web User Navigation Pattern mining approach based on Graph Partitioning algorithm, Journal of Theoretical and Applied Information Technology vol. 4, 2008, 1125-1130.

[Kardan 13] Kardan, A.A., Ebrahimi, M.: A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. Information Sciences 219, 93–110, 2013.

[Karam 06] Karam,O. H.,Hamad,A. M., Hussein, W.: Classification Rules for Pre-Analysis Filtering of Web Transactions, Fourth ACS/IEEE Conference on Computer Systems and Applications (AICCSA06), March, 2006, 764-770.

[Kazienko 06] Kazienko, P., Kolodziejski, P.: Personalized Integration of Recommendation Methods for E-commerce, In International Journal of Computer Science and Applications (IJCSA) 3(3), pp. 12-26, 2006.

[Kumar 09] Kumar, A.,Thambidurai,P.:Collaborative Web Recommendation Systems Based on an Effective Fuzzy Association Rule Mining Algorithm (FARM), In Indian Journal of Computer Science and Engineering Vol 1, 2009, 184-191.

[Mabroukeh 09] Mabroukeh, N.R., Ezeife, C.I.: Using Domain Ontology for Semantic Web Usage Mining and Next Page Prediction, In Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009, 1677-1680.

[Nadi 11] Nadi, S.,Saraee, M.H.,Bagheri, A.: A Hybrid Recommender System for Dynamic Web Users, In International Journal Multimedia and Image Processing (IJMIP), Volume 1, Issue 1, 2011, 3-8.

[Park 09] Park, S. T., Chu, W.: Pairwise preference regression for cold-start recommendation, Proceedings of the third ACM conference on Recommender systems - RecSys '09, 2009, 21-28.

[Pham 11] Pham, M. C., Cao, Y., Klamma, R., Jarke, M.: A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis, In Journal of Universal Computer Science 17(4),2011, 583-604.

[Rashid 08] Rashid, A., Karypis, G., Riedl, J.: Learning Preferences of New Users in Recommender Systems: An Information Theoretic Approach, Web Mining and Web Usage Analysis, , Las Vegas, Nevada, USA, August 24-27, 2008.

[Rosaci 13] Rosaci, D., Sarnè, G.M.L.: Recommending Multimedia Web Services in a Multi-Device Environment, Information Systems 38 (2) , pp. 198-212. 2013.

[Ruotsalo 10] Ruotsalo, T.:Methods and Applications for Ontology-Based Recommender Systems, Doctoral Dissertation, Department of Media Technology, Faculty of Informration and Natural Sciences, School of Science and Technology, Aalto University, June 2010.

[Sandvig 08] Sandvig,J.J.,Mobasher, B., Burke, R.: A Survey of Collaborative Recommendation and the Robustness of Model-Based Algorithms, In IEEE Data Eng. Bull. 31(2), 2008, 3-13.

[Shinde 12] Shinde, S.K., Kulkarni, U.: Hybrid Personalized recommender System using centering-bunching based clustering algorithm, Expert Systems with Applications, 39(2012), 1381-1387.

[Shinde 11] Shinde, S.K., Kulkarni, U.V.:Hybrid Personalized Recommender System Using Fast K-medoids Clustering Algorithm, Journal of Advances in Information Technology, Vol. 2, No. 3, August 2011.

[Sneha 12] Sneha, Y. S., Mahadevan, G.,Prakash, M., A Personalized Product Based Recommendation System Using Web Usage Mining and Semantic Web, In International Journal of Computer Theory and Engineering Vol. 4, April 2012.

[Thakur 12] Thakur, M., Pandey, G.S.: Performance Based Novel Techniques for Semantic Web Mining, International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012, 317-326.

[Tkalčič 11] Tkalčič, M.,Kunaver, M.,Košir, A.,Tasič, J.: Addressing the New User Problem with a Personality Based User Similarity Measure In Joint Proceedings of the DEMRA 2011 and the UMMS 2011 Workshops at the 19th International Conference on User Modeling, Adaptation and Personalization, UMAP 2011,Girona, Spain, July 11, 2011.

[Vozalis 03] Vozalis, E., Margaritis, K.: Analysis of Recommender Systems' Algorithms, In The Sixth Hellenic European Conference on Computer Mathematics and its Applications (HERCMA 2003), Athens, Greece, 2003.