Coupling Web Usage Mining with Content Mining for Recommendation Systems

Rehab Duwairi
Department of Computer Science and Engineering
Qatar University
Qatar
Rehab.Duwairi@qu.edu.qa

Ebraheem Elhaddad
Department of Computer Science,
Jordan Univ. of Sc. & Tech.
Jordan
alhaddad92@yahoo.com

Mohammed Al-Towaiq,
Department of Math. & Stat,
Jordan Univ. of Sc. & Tech
Jordan
towaiq@just.edu.jo

ABSTRACT

In this paper, we introduce a novel approach for combining web usage mining and content mining. The approach is based on generating recommendations that take into consideration users' navigation history as well as contents of pages visited. The integration of usage and content data can be performed at the offline phase or at the online phase of the recommendation system. The former is referred to as pre-mining integration; while the latter is called post-mining integration. The current work is concerned with pre-mining integration. The main rationales of the combination are to improve the quality of recommendations generated; overcome the new item problem and to improve the clustering of users' navigation patterns. In our experiments, the quality of clustering of user sessions is improved by 35% when compared with quality of clusters generated using usage patterns only. Also, the recommendations generated using the proposed approach are improved by 24% and 8% for the recall and precision respectively and when compared with recall and precision obtained by using usage data only.

Keywords: recommender systems, collaborative filtering, web usage mining, web content mining, and web personalization.

1. INTRODUCTION

Web personalization is any action done by the web sites' administrator through computer based techniques to customize the web site for specific user or users. The personalization can be represented as recommended links, products or targeted advertisements appearing to the users [8]. There are at least three approaches for building web personalization and recommendation systems, content-based, collaborative filtering and web usage mining [2].

The content-based approach tries to accurately anticipate the next user request, based on the content analysis of the recently visited pages by the user. These predictions are based on calculating the similarity of the current visited pages and the user's profile or model. Pages that reflect the user's interests and had never been visited by the user can then be recommended [3].

The collaborative filtering approach, by comparison, is one of the most popular web personalization and recommendation technologies and it uses peer opinions to predict the interests of others [9]. Here, instead of using the web pages content itself, the web pages' ratings are used to generate recommendations. One of the well known collaborative filtering techniques is the *K*-Nearest-Neighbor (*K*NN). Given the target user transaction, *K*NN compares this transaction with a transaction history of other users in order to find the *K* most similar users. The similarity can be based on ratings of items, access to similar pages, or purchase of similar items [2]. Neighborhood items are then used in the recommendation for the target user. Collaborative filtering overcomes some content based limitations. The recommendations of items for users are based on user explicit ratings, instead of the items' content. The advantage of this approach, when compared with pure content based approaches is that it can capture "pragmatic" relationships among items based on their intended use [2]. However, collaborative filtering suffers from some problems such as: The cold start problem: initially, there will not be enough ratings and it completely denies any information that can be extracted from the content [7].

Web usage mining is concerned with the application of data mining tools to web logs in order to discover users' usage patterns; this helps in understanding and better serving the needs of web-based applications [10]. Web usage mining is divided into offline and online components. The data preparation and the data mining techniques are applied at the offline component whereas the online component involves the recommendations to the active users; these recommendations are based on the information extracted from the offline component.

Hybrid approaches of content-based and collaborative filtering, content-based and web usage mining are also addressed. The combination is done to improve the overall system performance by combining the advantages of each technique.

A novel approach is introduced in this paper. It is basically based on a combination between the usage-mining of server log file and content mining for the corresponding web pages. The proposed approach consists of two components, namely, the online and offline components. The offline component consists of three major tasks, data preparation and preprocessing, generating similarity matrixes and clustering. The online component consists of two tasks, classification and recommendation set generation.

This paper is organized as follows: section 2 provides an overview of the proposed framework. Section 3 describes the dataset that was used. Section 4, on the other hand, describes the preprocessing and transformations that were applied to the dataset to make it suitable for the work at hand. Section 5 describes the process of generating the session similarity matrices (in usage, content, and usage-content spaces). Section 6, by comparison, describes the framework that was used to cluster users' sessions based on their similarity. Section 7, describes the process of assigning a new session to the most similar cluster to that session. Section 8 describes the recommendation process – where a set of pages is recommended to the user. Finally, section 10 draws the conclusions of this work.

2. COUPLING WEB USAGE AND CONTENT MINING

The dataset, which was used in this work, is split into training and testing subsets. The training sessions are the input to the offline component and the testing sessions are the input to the online component. At the offline component, several processing tasks were applied on the training sessions to extract users' patterns. These patterns are used, by the online component, to produce the recommendation set. The recommendation set is a group of web pages (URLs) that is predicted for the user to visit within her/his current session. The dataset preprocessing, generating similarity matrixes and clustering tasks are performed at the offline component, whereas classification and recommendation tasks are performed at the online component.

The dataset reformatting and irrelevant information elimination are done at the preprocessing task. We build three matrices, namely, content, usage and usage-content similarity matrices. The content similarity matrix holds the similarity between every two sessions in content, whereas the usage similarity matrix holds the similarity in usage. The usage-content similarity matrix is built by adding every element in the usage similarity matrix to its corresponding element in the content similarity matrix. The usage-content similarity matrix is used in the clustering task. After that, we cluster the users' sessions to *k* clusters using the *k*-means clustering algorithm. A new session is classified to the most appropriate cluster at the classification task which is the first task of the online component. The cluster's mean is used by the recommendation task to generate recommendations to the new active session.

3. DATASET DESCRIPTION

In our work, we used *msnbc.com* anonymous web data [5]. The dataset describes users' visits to the msnbc website for the entire day of September, 28, 1999. Each line in the dataset corresponds to a user access session to the website. Page requests are at the level of page category rather than URL level. All the pages are classified by a site administrator to seventeen topics which are "frontpage", "news", "tech", "local", "opinion", "on-air", "misc", "weather", "health", "living", "business", "sports", "summary", "bbs" (bulletin board service), "travel", "msn-news", and "msn-sports". These topics are assigned the numbers from 1 to 17 in the dataset, so 1 corresponds to "frontpage", 2 to "news", 3 to "tech" and so on. The number of URLs per category varies from 10 to 5,000. The dataset size is 11.9 MB and it contains 989,818 sessions.
Table 1 presents a sample of the dataset holding 2 different sessions, the first session s_0 represents one single session containing two web page visits, both visits correspond to topic 1 "frontpage", while session s_1 contains 7 webpage visits, six of them correspond to topic 1 "frontpage" and one of them corresponds to topic 11 "business". The space is used as a separator between any two consecutive web page visits in any session throughout the entire dataset.

Table1: A sample of user sessions in the dataset

Session Number	Sessions web visits
S_0	1 1
S_1	1 1 1 11 1 1 1

4. PREPROCESSING

Irrelevant and outlier sessions are removed from the dataset. Irrelevant sessions are sessions of length 1 (i.e. consist of a visit to one page only). Outlier sessions are sessions which consist of visits to 32 or more pages (such sessions are usually generated by automated programs) [4]. In total, 365,435 irrelevant sessions were eliminated and this left 624,383 in the dataset. The total number of outlier sessions eliminated is 9,813 distinct sessions. This leaves 613,698 sessions from the row dataset.
User sessions are of different lengths and the same topics which are visited by several users do not have to agree on the order of their appearance. To facilitate the automatic extraction of sessions that have common interests, every session is represented as a vector of length 17 (Recall 17 is the number of topics in the dataset). The first column represents topic 1 ("frontpage"), and the second column represents topic 2 "news" and so on. The Content-Matrix has the following definition:

$$C(i, j) = \begin{cases} n & \text{if session } i \text{ has visited topic } j \\ 0 & \text{if session } i \text{ did not visit topic } j \end{cases}$$

Where i is the session number, j is the topic number (from 1 to 17) and n is the number of visits performed by session i to topic j.
In usage preprocessing, we need to reformat the dataset in an appropriate format for capturing automatic usage behavior and comparing the site users based on their site navigation rather than their content interests. Analogues to the content-space, we project each user session in a usage-space. With n web pages identified uniquely by their URLs, $U = \{url_0, url_1 \ldots url_{(n-1)}\}$; and a set of m user sessions $S = \{s_0, s_1 \ldots s_{(m-1)}\}$, where m is the total number of sessions and each $s_i \in S$ is a non-empty subset of *U*. The following formula illustrates the representation of any session s_i in the usage-space.

$$s_i = (url_0, t), (url_1, t), \ldots, (url_{(n-1)}, t)$$

t is the time a user spent viewing a given webpage. We used binary weights in our work because the dataset doesn't involve information a bout the page view duration information [1]. This means that if a session refers to a page, then 1 is placed in the corresponding cell in the vector; 0 is placed otherwise.
Unfortunately, the dataset is recorded by the level of webpage category (topics) instead of the level of web page's URLs. To overcome this problem, we assume that the total number of URLs for all the 17 site's topics, is 10,000 distinct URL and we will refer to each web page by an id number instead of the page URL and to avoid duplicate ids we dedicate a unique range of ids for each topic. These ranges are proportional to the distribution of every topic in the dataset.

5. GENERATING SIMILARITY MATRIXES

We generated three similarity matrixes between all sessions, namely, content, usage and usage-content similarity matrixes. We gave these three matrixes the following symbols *C (m, m)*, *U (m, m)* and *UC (m, m)* respectively, where m is the number of sessions in the dataset. We used the cosine similarity measure to determine the similarity between any two sessions. These matrixes were used in the next step to cluster users' access sessions according to their similarity in usage and content.

Generating Content Similarity Matrix: This matrix is a two dimensional matrix where every cell contains a measure of the similarity between two sessions, here the content similarity between two sessions gives us an indicator to which level the content of the two sessions are similar. We apply the cosine similarity measure between every two sessions in the dataset using the content vectors representation. Equation 1 presents the mathematical expression of the cosine similarity between two vectors in the same space.

$$sim\ (v_i, v_j) = \frac{\sum_{l=1}^{l} v_{il} v_{jl}}{\sqrt{\sum_{l=1}^{l} v_{il}^2 \sum_{l=1}^{l} v_{jl}^2}} \quad (1)$$

Where: v_i, v_j: are the vectors of the sessions number i and j and l: is the length of the vectors.
As an example, every column in the row $C(1, m)$ of the matrix records the similarity between session number 1 and sessions from 0 to (m-1).

Generating Usage Similarity Matrix: This matrix is a two dimensional matrix where every cell contains a measure of the similarity between two sessions. The usage similarity between two sessions gives us an indicator on the similarity in browsing the website between two sessions**.** With the users' sessions represented in the usage-space, we know exactly what URLs a user visited in every session. As much as two sessions visit the same web pages, the usage similarity between them increases. Here also we use the cosine similarity as a similarity measure but the vectors now in the usage-space.

Generating Usage-Content Similarity Matrix: The goal behind integrating usage and content similarity matrixes is to have a measure of the similarity between users' sessions in both usage and pages' content. This matrix will be the input for the clustering task where similar sessions are grouped together in the same cluster. We generate the usage-content similarity matrix by adding every cell of the usage similarity matrix to the corresponding cell of the content similarity matrix. Note that the similarity values may be greater than 1 because in the Usage-Content Matrix, we are just adding corresponding similarity values from the Usage Matrix and the Content Matrix.

6. CLUSTERING

In this step, we group similar sessions based on their webpage content and based on their website usage. We have used the well known *k*-means clustering algorithm. This algorithm has three main issues, namely, choosing initial clusters' means, convergence criteria and determining the best value of *k*. We will describe how we implemented them in the following sub-sections. Sessions clustering results in a set of clusters, $C = \{c_1, c_2 \ldots c_i\}$, in which each c_i $(1 \le i \le k)$ is a subset of the set of user access sessions S, and k is the number of clusters.

Choosing Initial Clusters Means: To start the clustering algorithm, we need initial means to be used in the first clustering iteration. We used the Java random class to generate a random number in the range from 0 to the number of sessions to be clustered. Then we use this number as an index to the selection of the initial mean session. We have used this procedure to choose the initial means for the k clusters.

Convergence Criteria: After choosing the initial means, the Euclidian Distance is calculated between the candidate sessions to be clustered and the means of all clusters. The candidate session is placed in the cluster where the distance between this session and the cluster mean is the smallest. This operation is performed on all users' sessions until all of them are clustered in k clusters. Clustering all sessions is repeated several times until a convergence criterion is met. We assumed that convergence is achieved if the sessions stop shuffling between the clusters. This criterion works as follows: after each iteration, clusters' means are recalculated. The means of iteration i are compared with the means of iteration *(i -1)*. The convergence criterion is met when all the means of iteration i are equal to their corresponding means of iteration *(i -1)*; otherwise, the clustering continues for another iteration.

Determining the Best Value of *k*: Like other partitioning clustering algorithms, the number of clusters k is necessary to be specified in advance as an input for the *k*-means clustering algorithm. The best number of clusters here means that partitioning of users sessions will give us the best groups of similar users. Our approach is to apply the *k*-means algorithm to the access sessions using k values varying from 2 up to 15. For each k, we calculate clustering quality measure called cluster compactness (C_{mp}) and cluster separation (S_{ep}) proposed in [6] to measure the quality of the resulting clusters. Equations 2 and 4 give the mathematical expressions of (C_{mp}) and (S_{ep}) respectively. Equation 5 presents a measure that combines the two measures which is called overall cluster quality (O_{cq}).

$$C_{mp} = \frac{1}{k} \sum_{i=1}^{k} \frac{s(c_i)}{s(D)} \quad (2)$$

Where k is the number of clusters generated on the data set D, $\sigma(Ci)$ is the standard deviation of the cluster c_i, and $\sigma(D)$ is the standard deviation of the data set D. Equation 3 presents the mathematical expression of the standard deviation for any set Q.

$$s(Q) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} d^2\left(v_i, v_{(mean)Q}\right)} \quad (3)$$

Where $d(v_i, v_{(mean)Q})$ is the Euclidean distance between two vectors v_i and $v_{(mean)Q}$, N is the number of members in Q, and $v_{(mean)Q}$ is the mean of Q. The smaller the C_{mp} value, the higher the average compactness in the output clusters:

$$S_{ep} = \frac{1}{k(k-1)} \sum_{i=1}^{k} \sum_{j=1 j \ne i}^{k} \exp\left(-\frac{d^2\left(v_{(mean)ci}, v_{(mean)cj}\right)}{2s}\right) \quad (4)$$

where s is the standard deviation of the data set D, c is the number of clusters, *exp()* is the function that returns an exponential value, $v_{(mean)ci}$ is the centroid of the cluster c_i, $v_{(mean)cj}$ is the centroid of the cluster c_j, *d()* is the distance measure used by the clustering algorithm, and $d^2\left(v_{(mean)ci}, v_{(mean)cj}\right)$ is the Euclidian Distance between the centroid of cluster c_i and the centroid of cluster c_j. Similar to C_{mp}, the smaller the S_{ep} value, the larger the overall dissimilarity among the output clusters.

$$O_{cq}(b) = b * C_{mp} + (1-b) * S_{ep}, \quad (5)$$

Where $b \in [0,1]$ is the weight that balances the measures C_{mp} and S_{ep}. We used $O_{cq}(0.5)$ to give equal weights to the two measures. Therefore, the lower the O_{cq} value, the better quality of the overall output clusters. Naturally, the most satisfying quality score indicates the best partition of access sessions, while the corresponding k suggests the best number of clusters on the access sessions.

7. CLASSIFICATION

Classification is the first step in the online component of the recommendation system. We use testing sessions selected randomly from the dataset to simulate active user sessions. The testing sessions are divided into two parts, namely, classification part and recommendation part. The first part is used to stimulate an active user session and the second part is used to stimulate the future visits. The two parts are represented in the content-space. We calculated the mean vector of every cluster and used this mean as a representative of the clusters' members. The mean vector of a cluster is calculated by counting the total number of visits for every topic, in the content-space, in all cluster's sessions. With k cluster representatives, we compute the similarity $S_{im}(s_i, P_k)$, where s_i is the testing session to be classified and P_k is the representative of cluster k. The testing session is classified to the cluster with the highest S_{im}. Here we use the cosine similarity measure. Certainly, it's the well known K Nearest Neighbors (KNN) classification method, with $K = 1$. So now every testing session is classified to the nearest cluster based on web page's content and usage. The second part, recommendation part, represents the users' predicted future visits and it is used to evaluate the recommendations of the system.

8. RECOMMENDATION

The clusters' means represent the patterns of their clusters' members. Now, we need to choose from these patterns the recommendation set for the new active user. In a specific cluster mean, not all topics have the same importance; some of the topics are visited more than others. We use a threshold to filter out the least significant topics in every cluster mean. The threshold is a specific percentage of the total number of pages visited in all the clusters sessions. A 1% threshold means that any topic which was visited in less than 1% of the total number of cluster's sessions will be excluded from the cluster's mean. To choose the best threshold value, we tried values from 1% up to 25%.

9. EXPERIMENTATION AND RESULT ANALYSIS

Assessment Methodology: We use precision and recall to assess the prediction capabilities of our recommendation system. Our assessment strategy is based on comparing our recommendation system that couples the usage and content to generate recommendations for an active user to a recommendation system that uses only usage data to generate recommendations. The comparison is carried out using the recall and precision parameters. The two systems are assessed using the same training and testing data in all case studies.

Case Study: After all the preprocessing steps have been performed, we have 613,698 different sessions. We split out these sessions into 409,132 sessions (two thirds of data) for training and 204,566 sessions (one third of data) for testing. We took the training part starting from session number 0 up to session number 409,131 and the testing part from session number 409,132 up to 613,697. In this paper, we are going to discuss the results for the first case study which consists of the first 30,000 sessions (from session number 0 up to session 29,999) from the training data and the first 10,000 sessions from the testing sessions.

Experiments: The number of clusters (k) was determined experimentally, we have tried different values of k from 2 to 15. 15 was chosen as a maximum value for k due to the small size of resulting clusters which holds less than 1% of the total number of sessions using k values larger than 15. Figure 1 shows the clusters' quality when the number of clusters ranges from 2 to 5 for both usage and content based recommendation and pure usage based recommendation. Recall that the measure for cluster quality used here is a minimization function. It is clear from the figure that the quality of clusters generated using usage and content data is better than the quality of clusters generated using content data only. This impacts the quality of the recommendations as well.

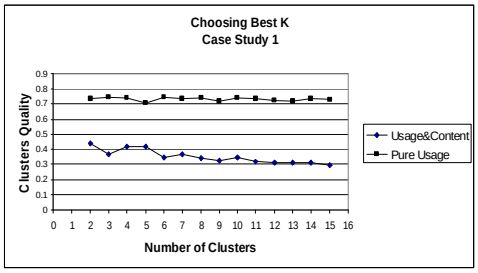


Figure 1: Cluster quality for both usage & content and pure usage

We experimented with several values for thresholds to determine the interestingness of topics within a given cluster. These thresholds range from 1% to 25%. n% means that a topic is considered uninteresting to a cluster if less than n% of that cluster's sessions accessed a topic. The precision and recall values are plotted against several values of thresholds as shown in figure 2. The values of precision and recall are the average precision and recall for all testing sessions. We observe that, with the increase of the threshold value, the precision increases, whereas the recall decreases. This continues until the two curves intersect at threshold=6, then the curves almost settle.

Figure 3 shows the precision values for the recommender system that combines usage and content data and the recommender system that uses usage data only. It is clear that combining usage and content data increases the precision. Figure 4 shows the recall values for the two cases. Recall values for the usage only case are higher than the recall values of usage and content case for threshold values less than 12. However, this behavior is reversed for threshold values greater than 12.

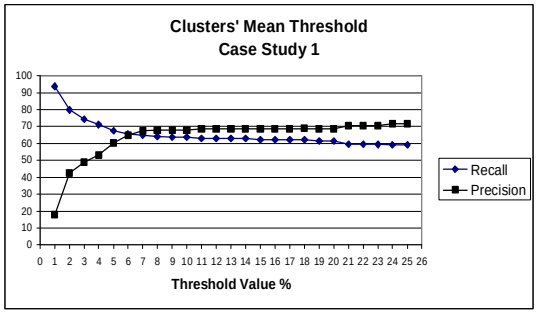


Figure 2: Choosing clusters' mean threshold for case study 1.

10. CONCLUSION

This paper has presented a novel approach for coupling content and usage data in order to generate recommendations to users. The recommendations are in the form of suggesting web pages to the user to visit based on comparing that user's interests with similar users who have visited the website earlier. The current user's interests are discovered by mining the content and usage properties of the few pages she/he has visited. The experiments that were conducted showed that coupling the usage and content data when generating recommendations improves the quality of clusters and the quality of recommended pages. The quality of clusters is computed using cluster compactness and separations; both are minimizations functions. The quality of the recommended web pages is determined using precision and recall.

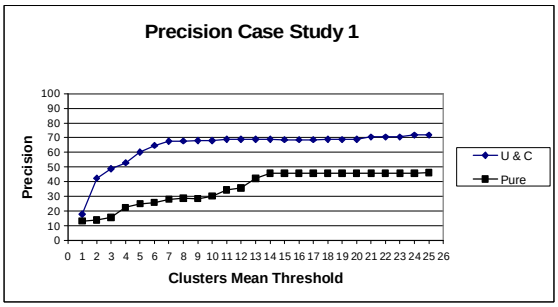


Figure 3: Precision for case study 1.

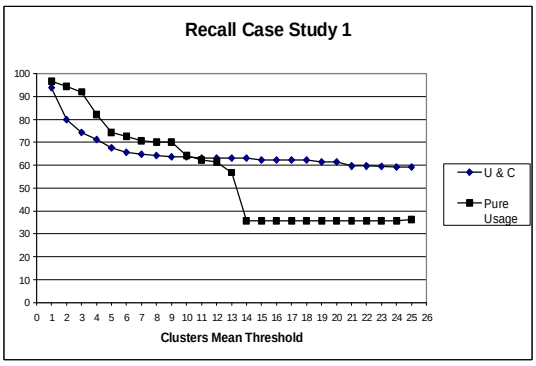


Figure 4: Recall for case study 1.

REFERENCES

[1] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, M. Sartin. Combining Content-Based and Collaborative Filters in an Online Newspaper, **Proceedings of ACM SIGIR Workshop on Recommender Systems**, Aug. 1999.

[2] H. Dai, B. and Mobasher, A road map to more effective web personalization: Integrating domain knowledge with web usage mining. **Proceedings of the International Conference on Internet Computing**; 2003, pp. 58–64; Las Vegas, Nevada.

[3] B. D. Davison. Predicting web actions from html content. **Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia** (HT'02), 2002: pp. 159–168; College Park, MD.

[4] L. Haibin, K. Vlado: Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. **Data & Knowledge Engineering**, Vol. 61, No. 2, 2007, pp. 304-330.

[5] S. Hettich, S. D. Bay. **The UCI KDD Archive** [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.

[6] J. He, T. Ah-Hwee, L. T. Chew, Y. S. Sam. On quantitative evaluation of clustering systems, **Clustering and Information Retrieval**, Kluwer Academic Publishers, 2003: pp. 105–134.

[7] B. M. Kim, Q. Li, C. S. Park, S. Kim. A New Approach for Combining Content-based and Collaborative Filters, **Journal of Intelligence Information Systems**, Vol. 27, 2006; pp. 79-91.

[8] B. Mobasher, H. Dai, T. Luo, Y. Sun, J. Zhu, Integrating web usage and content mining for more effective personalization. P**roceedings of the First International Conference on Electronic Commerce and Web Technologies**, 2000: pp. 165–176; Springer-Verlag.

[9] Q. Li, M. K. Byeong, An approach for combining content-based and collaborative filters. Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages, 2003; pp. 17-24 Sappro, Japan.

[10] J. Strivastrava, R. Cooley, M. Deshpande, and P. N. Tan. Web usage mining: discovery and applications of usage patterns from web data. **SIGKDD Explorations**, Vol. 1, 2000; pp. 12-23.