Proceedings of the Twenty-First International FLAIRS Conference (2008)

Utilizing Content to Enhance a Usage-Based Method for Web Recommendation based on Q-Learning

Nima Taghipour
Department of Computer Engineering
Amirkabir University of Technology, Tehran, Iran
n-taghipour@aut.ac.ir

Abstract

The problem of information overload on the Internet has received a great deal of attention in the recent years. Recommender Systems have been introduced as one solution to this problem. These systems aim at directing the user toward the items that best meet her needs and interests. Recent studies have indicated the effectiveness of incorporating domain knowledge in improving the quality of recommendations. In this paper we exploit this approach to enhance a reinforcement learning framework, primarily devised for web recommendations based on web usage data. A hybrid, i.e. content- and usage-based, web recommendation method is proposed by incorporating web content information into a model of user behavior learned form usage data. Content information is utilized to find similarities between usage scenarios, i.e. users' seeking their information needs, and new recommendation strategies are proposed that are based on this enhanced model of user behavior. We evaluate our method under different settings and show how this method can overcome the shortcomings of the usage-based approach and improve the overall quality of recommendations.

Introduction

The volume of information available on the internet is increasing rapidly with the explosive growth of the World Wide Web and the advent of e-Commerce. While in one hand, users are provided with more information and service options, on the other hand it has become more difficult for them to find the “right” or “interesting” information, the problem commonly known as information overload. Recommender systems have been introduced as a solution to this problem (Resnick and Varian 1997). They can be generally defined as systems that guide users toward interesting or useful objects in a large space of possible options (Burke 2002).

One popular application area for recommender systems is web content recommendations. Web recommendation is considered a user modeling or web personalization task (Eirinaki et al. 2004). One research area that has recently contributed greatly to this problem is web mining. Most of the systems developed in this field are based on web usage mining (Srivastava et al. 2000) which is the process of applying data mining techniques to the discovery of usage patterns form web data. These systems are mainly concerned with discovering patterns from web usage logs and making recommendations based on the extracted navigation patterns (Fu et al. 2000; Mobasher et al. 2000a). Unlike traditional recommender systems, which mainly base their decisions on user ratings on different items or other explicit feedbacks provided by the user, these techniques discover user preferences from their implicit feedbacks, namely the web pages they have visited. More recently, hybrid methods that take advantage of domain knowledge, i.e. content, usage and even structural information of the websites, have been introduced (Bose et al. 2006; Eirinaki et al. 2003; Li & Zaiane 2004; Mobasher et al. 2000b; Nakagawa & Mobasher 2003) and shown superior results in the web page recommendation problem.

In (Nakagawa & Mobasher, 2003) the degree of connectivity based on the link structure of the website is used to evaluate effectiveness of different usage based techniques for web sites with different structures. A new method for generating navigation models is presented in (Li & Zaiane 2004) which exploits the usage, content and structure data of the website and addresses parallel information needs of the user. Eirinaki et al. (2004, 2003) use the content of web pages to augment usage profiles with semantics using a domain-ontology. Most recently, concept hierarchies were incorporated in a novel recommendation method based on web usage mining and optimal sequence alignment to find conceptual similarities between user sessions (Bose et al. 2006).

In this paper we exploit this idea to enhance a reinforcement learning solution, devised for web recommendations based on web usage data (Taghipour et al. 2007). Although the mentioned technique has shown promising results in comparison to common techniques like collaborative filtering and association rules, an analysis of the system's performance, showed that this method suffers from the problems commonly faced by other usage-based techniques. We tackle these problems by proposing a hybrid, i.e. content- and usage-based, web recommendation method by exploiting web content information in the model of user behavior learned form

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

usage data. We devise content models for user navigation sequence and utilize the content information to find the regularities and similarities between usage scenarios, i.e. users' seeking their information needs by browsing the web. New recommendation strategies are proposed based on this enhanced model of user behavior. Content-wise similar usage scenarios are exploited as cases of user information need, based on their sequential interactions with web content, which also indicate what items should be recommended to satisfy the information need. Our hybrid model for the web page recommendation problem emphasizes the flexibility of the reinforcement learning framework for this problem and how it can be utilized to incorporate other sources of information. We evaluate our method under different settings and show how this method can improve the overall quality of web recommendations.

The organization of the paper is as follows: First, we overview the usage-based method which is the basis of our method. Next, we give detailed descriptions of our hybrid methods. Afterwards we evaluate the methods and finally comes our conclusion along with recommendations for future work.

Background

In this section we overview the method proposed in (Taghipour et al. 2007) that forms the basis of our new solution. The proposed method exploits Reinforcement Learning (RL) to make recommendations from web usage data. We also point out the main weaknesses of the method which we aim to overcome.

Web Recommendations Based on Reinforcement Learning

Reinforcement learning (Sutton & Barto 1998) is primarily known in machine learning research as a framework in which agents learn to choose the optimal action in each situation or state they are in. The goal of the agent is to learn which actions to perform in each state to receive the greatest accumulative reward, in its path to the goal state. To model the problem as reinforcement learning, they use the analogy of a game in which the system is constantly trying to predict the next pages of the user session, knowing her previous requests and the history of other users browsing sessions.

Using the notions of N-Grams, each state S at time t consists of two sequences V, R indicating the sequence of last w visited and w' recommended pages respectively:

$$V_s = < p_{t-w+1}, p_{t-w+2}, \ldots, p_t > \qquad (1)$$
$$R_s = < R_{t-w'+1}, R_{t-w'+2}, \ldots, R_t >$$

Where P_i and R_i indicate the ith visited and recommended page in the state (Figure 1). Reward for each action would be a function of $V_{s'}$ and $R_{s'}$ where S' is the next state. A state S' is rewarded when the last page visited belongs to the recommended pages list. To completely define the reward common metrics normally used in web page recommender systems are taken into account. One aspect to consider is when the visited page was actually predicted by the system, in order to reward recommendations that shorten the browsing sessions. Another factor commonly considered in theses systems (Mobasher et al. 2000a; Fu et al. 2000) is the time the user spends on a page. The common assumption is that the more time the user spends on a page the more interested he probably is in that page. The rewarding can be summarized as:

Algorithm 1: Usage Based Reward Function

Assume $\delta(s, a) = s'$
$P_R = V_{s',w} \cap R_{s'} = P_{t+1} \cap R_{s'}$
If $P_R \neq \emptyset$
  For each *page r* **in** P_R
    $r(s,a,s') = r(s',P_{t+1}) \mathrel{+}= reward(Dist(R_{s'}, r), Time(P_{t+1}))$
  End For
End If

Where $Dist(Rs', r)$ is the distance of page r from the end of the recommended pages list, and $Time(P_{t+1})$ is the time user has spent on the last page of the state.

As the sequence of previously recommended pages R is restricted to a constant number w', the effect of each action is limited to w' next states and the system was mostly successful in recommending pages visited around w' steps ahead. This tends to limit system’s prediction ability as large numbers of w' make the state space enormous. To overcome this problem a modification is devised in reward function. The basic idea is that when an action/recommendation is appropriate in state S_i, indicating the recommended page is likely to occur in the following states, it should also be considered appropriate in state S_{i-1}, the actions in S_{i-1} that frequently lead to S_i.

Limitations of the Usage-Based Approach

In our evaluation of the system, we noticed that although we were faced with a rather large number of states, there were cases where the state resulted from the sequence of pages visited by the user had actually never occurred in the training phase. Although not the case here, this problem can be also due to the infamous "new item" problem commonly faced in collaborative filtering (Burke 2002; Mobasher et al. 2000b) when new pages are added to the website. In situations like these the system was unable to make any decisions regarding what pages to recommend.

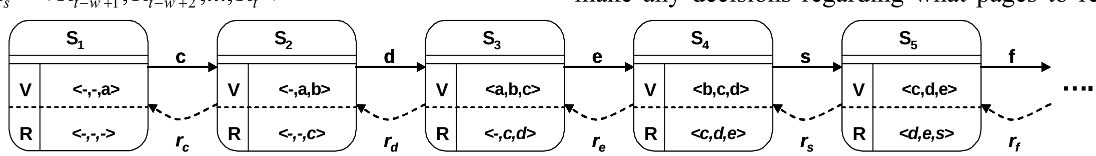

Figure 1: States and actions in the recommendation model

Moreover, the overall coverage of the system on the website, i.e. percentage of the pages that were recommended at least once, was rather low (55.06%). Another issue worth considering is the fact that the mere presence of a state in the state space cannot guarantee a high quality recommendation, i.e. a high Q-value cannot guarantee a high quality recommendation by itself. Simply put, when a pattern has few occurrences in the training data it cannot be a strong basis for decision making, a problem addressed in other methods by introducing metrics like support threshold in association rules (Mobasher et al. 2000b). Similarly in our case a high Q-value, like a high confidence for an association rule, cannot be trusted unless it has strong supporting evidence in the data. Generally, there are cases when historical usage data provides no evidence, or evidence that's not strong enough, to make a rational decision about user's behavior.

This is a problem common in recommender systems that have usage data as their only source of information. Note that in the described setting, pages stored in the *V* sequence of each state *S* are treated as items for which the only information available is their id. The system relies solely on usage data and thus is unable to make any generalization. One common solution to this problem is to incorporate some semantic knowledge about the items being recommended, into the system. In the next section we describe our approach for adopting this idea

Exploiting Content Implications of the Usage-Based Navigational Model

Motivation

In the usage-based model of the problem, each state models the information need of the user as the (sub)sequence of web pages visited by the user and the interest he has shown in each page, e.g. by the time spent visiting the page. Since no additional information beside the page id is used in the *V* sequence, this method fails to recognize any new sequence of page visits and consequently fails to make predictions. The same is true regarding the action/recommendations, as the system would only be able to recommend the exact pages it has seen before. In this section we will elaborate our approach to address these issues based on using the content of the pages in each state.

We devise content models for the states and exploit these models to find the states representing similar user information needs. Having this knowledge, we will be able to make use of the learned user behavior in satisfying similar information needs. For this purpose we devise new recommendation strategies that make use of clusters of similar states and the appropriate recommendations, for those states to derive a new aggregate score for predictions, in a *k*-NN fashion. The basic idea is that whenever the user browsing results in a state S_x which has been visited no or few times before, the system would find a set $Sim_k(S_x)$ containing the *k* most similar states to S_x and compute new recommendation scores for each candidate web page *r* by utilizing the scores, i.e. Q-values, of recommending *r* in each state $S_i \in Sim_k(S_x)$. For example the aggregation can be done by a weighted sum of the Q-Values, such as:

$$Score(S_x, r) = \frac{\sum_{Si \in Sim_k(S_x)} W(S_i) \times Q(S_i, r)}{\sum_{Si \in Sim_k(S_x)} W(S_i)} \quad (2)$$

In the following sections we will present different strategies for the recommendation phase.

Similarity between the States

Finding similarities between states is not a trivial task. First, we need to model the content information encapsulated in the state. Any state contains a fixed number (*w*) of sequential page visits by a user. The content information of this sequence can be modeled in various ways e.g. *w* bag of words, a vector aggregating all page contents in a state, etc. We present two approaches to model the content information of the states in attempt to capture a model of user information need based on her traversal on the website:

Content Sequence Model (CSM). In this approach we exploit various sources of information about each web page p_j in the *V* sequence, combine this information into an aggregated vector $PC(p_j)$ in the vector space model and finally derive a content model $SC(S_i)$ for each state *S* as an ordered set of these *w* vectors. The content vector of each web page is computed by combining the content of the web page, the terms in the URL of the page and the anchor text of the hyperlinks pointing to the web page $InLink(p_j)$. This model is adopted from (Eirinaki et al. 2003):

$$\begin{aligned}\overrightarrow{PC}(p_j) &= \alpha.\overrightarrow{Conten}(p_j) + \beta.\overrightarrow{InLink}(p_j) + \gamma.\overrightarrow{URL}(p_j)) \\ &= (t_{i1}, t_{i2}, t_{i3}, \ldots, t_{iN}) \\ &, where \quad \alpha + \beta + \gamma = 1\end{aligned} \quad (3)$$

$$SC(S_i) = < \overrightarrow{PC}(p_{i1}), \overrightarrow{PC}(p_{i2}), \ldots, \overrightarrow{PC}(p_{iw}) > \quad (4)$$

Where *N* is the total number of terms as commonly considered in the vector space model and the weights t_{ij} are computed using the *tf.Idf* metric. Having defined the content model the next step is to devise the similarity function $CSM\_Sim(S_i,S_j)$. We compute this similarity as the weighted cosine-based similarity *(CosSim)* of corresponding pages in the states. More weight is assigned to pages that appear later in the user session based on the assumption that users are *browsing towards* their information need. The similarity is computed as:

$$\frac{\sum_{k=1}^{w} CosSim(PC(p_{ik}), PC(p_{jk})) \times pweight(k)}{\sum_{k=1}^{w} pweight(k)} \quad (5)$$

Information Scent Model (ISM). Information Scent (Chi et. al. 2001) is a model of user information need based on the information foraging theory. In this model user's information need is modeled using the text of the hyperlinks followed by the user in her browsing session on the website and applying a spreading activation algorithm. We refer the interested reader to (Chi et. al. 2001) for details of this approach. We exploit this algorithm to achieve a model of user information need based on the sequence of visited pages in each state. The result of this procedure is also a vector in the vector space model and the similarity between states will again be computed based on the cosine of the corresponding vectors.

Organization of Similar States Another issue worth mentioning is that regardless of the choice of the content model, the process of finding similar states has to be time-efficient as this process is performed during the online recommendation generation phase. In this regard, the states are clustered based on their content model in the offline training phase and the search space for finding k similar states will be reduced to the corresponding cluster of the given state S_x. We incorporated our similarity functions and modified the Dc-tree clustering algorithm (Wong and Fu 2000) for forming state clusters in our experiments. This algorithm is an incremental hierarchical clustering algorithm specifically devised for finding clusters of web pages. The incremental feature of the algorithm specifically desired as it enables us to accommodate new states into the appropriate cluster when needed.

Recommendation Generation

The performance of the system relies heavily on the recommendation strategy. We propose two approaches for web page recommendation by exploiting the content information of the navigation model, each with different motivations.

Recommendation based on an Aggregate of Similar States (RASS). This strategy is analogous to the approach mentioned as the motivation. Here, we exploit the fact that the usage-based model generates accurate recommendations when it is provided with sufficient usage data (Taghipour et al. 2007). So, whenever the user session results in a state S_u which has been frequently visited by previous users, recommendations are made solely based on the values of $Q(S_u, a)$ for different actions.

On the other hand, the usage-based method loses its accuracy when faced with less frequently visited sequence of pages is completely useless when faced with new sequences. So, whenever S_u is a new or rarely visited state, an aggregate prediction of similar states will be exploited for recommendation. The pseudo code in Algorithm 2 summarizes the procedure.

If S_u is a new state the corresponding content model $SC(S_u)$ of the state is computed. This model is then compared to the mean content vector of the state clusters and the nearest cluster is found based on Equation (5). In this similarity computation, *pweight(k)* is also dependant to the time the user has spent on each pages p_{uk}, assuming pages the user has spent more time on as stronger indicators of her interest. In the last step recommendation scores for candidate recommendations/actions will be computed using the nearest states $S_i \in Sim_k(S_u)$ in the selected cluster and by applying Equation (2).

Note that in this step we tend to assign more weight ($W(S_i)$) to the states which have occurred more frequently in the usage data. The whole procedure is the same where S_u resides in our state space but it has been visited less than a given threshold. The only difference is that in this case we know the cluster the state belongs to in advance and there's no need to compute the corresponding content model and find nearest clusters.

Algorithm 2: RASS

Assume Current State is S_u
If (S_u **is not** a new state)
  If Usage_Support(S_u) > min_sup
    Return BestActions(S_u)
    Exit;
  Else
    $C \leftarrow FindCluster(S_u)$
  End If
Else
  Build $SC(S_u)$
  $C \leftarrow ClosestCluster(SC(S_u))$
End If
$Sim_k(S_u) \leftarrow FindSimilar(C, SC(S_u), k)$
For each *Action r*
  Compute $Score(S_u, r)$ using Equation (2)
  Store $Score(S_u, r)$ in a set *Recs*
End For
Return BestActions(*R*ecs)

Using this strategy, we will be able to find portions of sessions which are both semantically similar to our session, in sense of the user information need, and have a rather stronger usage support. Then we recommend a combination of actions appropriate for those states. We expect to be able to cover a higher portion of user sessions and also gain higher recommendation coverage on the web pages. Although this method manages the new sequences (even those containing new pages added to the website) it still fails to recommend those newly added pages (at least before a deal of experimentation).

Inferring a Content-based Model for Recommendations from Similar States (CMSS). In this strategy, again we exploit the Q-Values of actions in similar state to a given state S_u, but this time instead of using the actions as they are, i.e. prediction of page *id*s, we derive an aggregate content model predicting the content of pages that might satisfy user information needs. The procedure is basically similar to Algorithm 2. Again a set *Recs* is found containing the best actions from states in $Sim_k(S_u)$. Afterwards, the content model of the top m pages, those with higher $Score(S_u, r)$ in *Recs*, will be used to derive an aggregate content model of pages that might satisfy the user's information need. Then this content model *(RecQ)*, as a weighted vector of terms, will be used as a *query* on pages of the website. The ranked web pages retrieved for the query will then be used as the recommendations to be presented to the user. *RecQ* is computed as normalized weighted sum of the content vectors of pages in *Recs*:

$$\mathrm{Re}\,cQ_R = \frac{\sum_{r_i \in Top_m(\mathrm{Re}\,cs)} \overrightarrow{PC}(r_i) \times Score(S_u, r_i)}{\sum_{r_i \in Top_m(\mathrm{Re}\,cs)} Score(S_u, r_i)} \qquad (6)$$

This strategy is also a hybrid of usage and content-based approaches, and it differs from content-based recommendations where content similar to the items accessed so far by the user is retrieved for recommendations. Here, the previous usage patterns are exploited and the content query used for recommendations is based on a prediction of the content that would follow the sequence of web pages accessed by the user. An important feature of this strategy is the possibility of recommending new pages added to the web site, as the content queries would be evaluated against all the web pages, including the new pages or even pages with dynamic content.

Experimental Evaluation

Experimental Setting

We evaluated system performance in different settings described above. As our evaluation data set we used the web logs of the university website. This dataset contains 20000 sessions and about 680 pages. 60% of the data set was used as the training set and the remaining was used to test the system. For our evaluation we presented each user session to the system, and recorded the recommendations it made after seeing each page the user had visited similar to the original approach presented in (Taghipour et al. 2007).

We used the metrics proposed in (Bose et al. 2006) for evaluation. These metrics are: *Hit Ratio (HR)*: Percentage of *Hits* in recommendation lists, *Predictive Ability (PA)*: Percentage of pages recommended at least once, *Click Reduction (CR)*: average percentage of pages skipped because of recommendations and *Recommendation Quality (RQ):* average rank of a correct recommendation in the list.

Sensitivity to Parameters

Different choice of parameters would result in different types of system performance. *min_sup* (Algorithm2) would determine when recommendations for the existing states are made based on the usage-based approach, and when similar states would be exploited. We evaluated the usage-based method and analyzed the relationship between the frequencies of each state's *V* sequence in training sessions and the average accuracy of recommendations made for the state. These results showed significance reduction of accuracy for states with support values lower than 0.31%, the ratio that we chose as the *min-sup* threshold. Another parameter is *k* the size of the S_x neighborhood $Sim_k(S_x)$. We evaluated the system with different *k* values as shown in Figure 2. We have a local optimum for *k* in both strategies (*k*=15 for *RASS* and *k*=20 fro *CMSS*). Increasing *k* at lower value ranges improves the performance by bringing in the knowledge stored in similar states. After reaching a local optimum the performance stays rather flat as we obtain an almost fixed set of best actions, with highest $Score(S_u,r)$. Increasing the neighborhood beyond 30 seems to bring noise into the model which hurts the performance in spite of the dampening effect of low $W(S_i)$ values for the less similar neighbors.

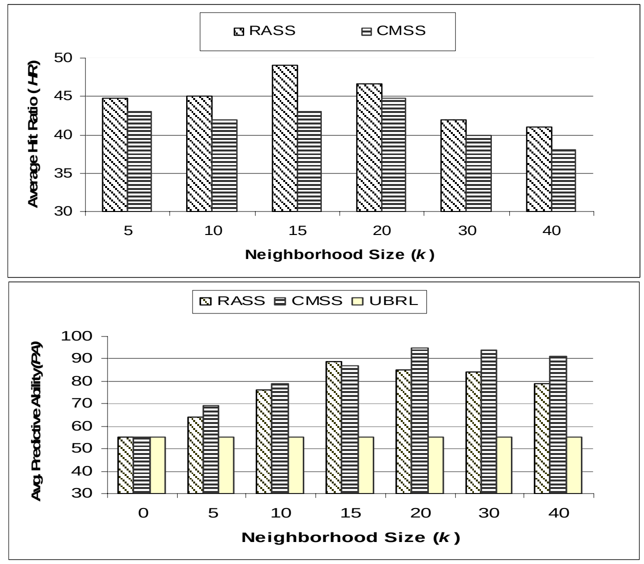

Figure 2: Effect of Parameter *k* on Hit Ratio (Top) and Predictive Ability (Bottom) of the Method

Comparison to Other Methods

We compared our proposed hybrid methods with the previous usage-based *(UB-RL)* and a content-based approach that uses the info-scent model to recommend pages *(CIS)*. Note that *UB-RL* had shown superior results than common usage-based methods (Taghipour et al. 2007). Combination of the chosen strategy (*RASS/CMSS*) and content model (*CSM*/*ISM*) result in 4 different methods. The results presented here are based on having a maximum of 7 recommendations in each stage. We also experimented with 5 and 10 as the thresholds which resulted in the same relative performance of the methods. As show in Table 1, the *RASS-CSM* out performs the rest of the methods (including the accurate *UB-RL* method) in sense of *HR*, while also achieving high levels of coverage on the website. We can see how all our hybrid methods manage to dramatically outperform *UB-RL* in sense of *PA* and fairly compete with the, not so accurate, content-based *CIS* method. Both of the methods based on the *CMSS* strategy achieve highest levels of *PA* (and *CR*), due to the diversity of the content queries. The much higher *HR* values of these methods compared to the pure content-based *CIS* approach could be considered as evidence in support of the importance of actual usage patterns in accurate inference of user information needs and behavior.

Table 1: Comparison of different recommendation methods

Session window size *w*=3, min_sup=0.31%, k=15 (RASS), k=20 (CMSS)

Method	Metric				
	HR	PA	CR	RQ	*PA-N*
UB-RL	47.91	55.06	14.17	3.22	-
CIS	34.11	67.12	9.31	5.23	65.41
RASS-CSM	50.02	94.91	23.56	3.51	-
RASS-ISM	49.22	93.40	21.70	3.67	-
CMSS-CSM	46.15	96.10	25.90	4.01	90.21
CMSS-ISM	45.12	94.30	25.73	4.33	88.34

Finally, we can see that the more elaborate *CSM* content model results in relatively better performance than the *ISM* approach, but the marginal difference proves the accuracy of the *ISM* approach, which is favorable due to less computation load, especially for the online recommendation phase. We also evaluated the ability of the methods to recommend new pages, omitted from training sessions in another simulation, shown by the evaluation metric *PA-N*. It can be seen that the *CMSS* strategies manage to achieve a high coverage on these pages while other approaches fail, as expected.

Conclusion and Future Work

We presented methods to enhance, and overcome the restrictions, of a usage-based web recommender system based on RL. We identified and tackled the weaknesses that a usage-based method inherently suffers from and incorporated content information regarding the usage patterns to improve the system. Our evaluation results emphasize the flexibility of the RL paradigm to combine different sources of information in order to derive predictions of future content needs and improve the quality of recommendations.

This work can be extended in various dimensions. One possibility is a recommendation method that uses a combination of recommendations generated by the two strategies, in order to benefit from each, i.e. better accuracy of the *RASS*, diversity of *CMSS* and its ability to accommodate new pages. Another option is using a content-based, or a hybrid, reward function in the RL model which considers the similarity of pages in *R* to the visited page. Integration of other sources of domain knowledge e.g. website topology or a domain-ontology into the model can also be another future work for this paper. Finally, we propose integration of some long lasting characteristics of the user, e.g. her long-term interests or a model of her browsing strategy, into the state model.

References

Bose, A., Beemanapalli, K., Srivastava, J., Sahar, S. 2006 Incorporating concept hierarchies into usage mining based recommendations. *Proc. 8th WEBKDD workshop.*

Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*. 12:331-370.

Chi, E. H., Pirolli, P., Pitkow, J. 2001. Using information scent to model user Information Needs and Actions on the Web. *Proc.s of Human Factors in Computing Systems*.

Eirinaki, M., Vazirgiannis, M., Varlamis, I. 2003. SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process. *Proc. of the 9th ACM SIGKDD Conference*.

Eirinaki, M., Lampos, C., Paulakis, S., Vazirgiannis, M. 2004. Web Personalization Integrating Content Semantics and Navigational Patterns. *Proc. of the sixth ACM workshop on Web Information and Data Management*.

Fu, X., Budzik, J., Hammond, K. J. 2000 Mining navigation history for recommendation. *Proc .of 4th International Conference on Intelligent User Interfaces*.

Li, J., Zaiane, O. R. 2004. Combining usage, content and structure data to improve web site recommendation. *5th International Conference on Electronic Commerce and Web*. EC-WEB.

Mobasher, B., Cooley, R., Srivastava, J. 2000. Automatic personalization based on web usage mining. *Communications of the ACM*. 43 (8), pp. 142-151.

Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J. 2000. Integrating web usage and content mining for more effective personalization. In *Proc. of International Conference on Electronic Commerce and Web*. EC-WEB.

Nakagawa M., Mobasher, B. 2003. A Hybrid Web Personalization Model Based on Site Connectivity. *Proc. of 5th WEBKDD workshop*.

Resnick, P., Varian, H.R. 1997. Recommender Systems. *Communications of the ACM*, 40 (3):56-58.

Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N. 2000. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1:12–23.

Sutton, R.S. Barto, A.G. 1998. Reinforcement Learning: An Introduction, *MIT Press, Cambridge*.

Taghipour, N., Kardan, A., Shiry Ghidary, S. 2007. Usage-Based Web Recommendations: A Reinforcement Learning Approach. *Proc. of the ACM 2007 Recommender Systems Conference*.

Wong, W. and Fu, A. 2000. Incremental document clustering for web page classification. Proc. International conference on Information Society in the 21st century: Emerging Technologies and New Challenges.