A Hybrid Web Recommender System Based on Q-Learning

Nima Taghipour
Amirkabir University of Technology
Department of Computer Engineering
424, Hafez, Tehran Iran
0098 (21) 22286275
n-taghipour@aut.ac.ir

Ahmad Kardan
Amirkabir University of Technology
Department of Computer Engineering
424, Hafez, Tehran Iran
0098 (21) 64542729
aakardan@aut.ac.ir

ABSTRACT

Different efforts have been made to address the problem of information overload on the Internet. Recommender systems aim at directing users through this information space, toward the resources that best meet their needs and interests. Web Content Recommendation has been an active application area for Information Filtering, Web Mining and Machine Learning research. Recent studies show that combining the conceptual and usage information can improve the quality of web recommendations. In this paper we exploit this idea to enhance a reinforcement learning framework, primarily devised for web recommendations based on web usage data. A hybrid web recommendation method is proposed by making use of the conceptual relationships among web resources to derive a novel model of the problem, enriched with semantic knowledge about the usage behavior. With our hybrid model for the web page recommendation problem we show the apt and flexibility of the reinforcement learning framework in the web recommendation domain, and demonstrate how it can be extended in order to incorporate various sources of information. We evaluate our method under different settings and show how this method can improve the overall quality of web recommendations.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information Filtering.*

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Recommender Systems, Personalization, Machine Learning, Reinforcement Learning, Web Mining

1. 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 users are provided with more information and service options, 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 [13]. They can be generally defined as systems that guide users toward interesting or useful objects in a large space of possible options [2].

Web page recommendation is considered a user modeling or web personalization task [6]. 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 (WUM) [14] 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 [7,10]. Unlike traditional recommender systems, which mainly base their decisions on user ratings on different items or other explicit feedbacks provided by the user [4] these techniques discover user preferences from their implicit feedbacks, namely the web pages they have visited. More recently, systems that take advantage of a combination of content, usage and even structural information of the websites have been introduced [1,5,6,9,11] and shown superior results in the web page recommendation problem.

In [12] the degree of connectivity based on the link structure of the website is used to evaluate usage based techniques. A new method for generating navigation models is presented in [9] which exploits the usage, content and structure data of the website. Eirikani et al. [5,6] 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 WUM and optimal sequence alignment to find similarities between user sessions [1].

In this paper we exploit this idea to enhance a reinforcement learning solution, we had devised for web recommendations based on web usage data [16]. 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 (Section 2.2). To address these problems, we made use of the conceptual relationships among web pages and derived a novel model of the problem, enriched with semantic knowledge about the usage behavior. We used existing methods to derive a conceptual structure of the website. Then we came up with new definitions for our states, actions and rewarding functions which capture the semantic implications of users browsing behavior. Our hybrid model for the web page recommendation problem shows the flexibility of the reinforcement learning framework for the recommendation problem and how it can be extended 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: in section 2 we overview the usage-based method which is the basis of our method. We represent our hybrid approach in section 3 and evaluate the method in section 4. Section 5 concludes the paper with recommendation for future work.

2. BACKGROUND

In this section we overview our method proposed in [16] that forms the basis of our new solution. The proposed method exploits Reinforcement Learning (RL) to make recommendations from web usage data.

2.1 Web Recommendations Based on Reinforcement Learning

Reinforcement learning [15] 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 his 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 visited and previously recommended pages respectively:

$$\begin{aligned} V_s &= < p_{t-w+1}, p_{t-w+2}, \ldots, p_t > \\ R_s &= < R_{t-w'+1}, R_{t-w'+2}, \ldots, R_t > \end{aligned} \tag{1}$$

Where P_i and R_i indicates the *i*th 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 [1,10,7] 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:

1. ***Assume*** $\delta(s, a) = s'$
2. $P_R = V_{s',w} \cap R_{s'} = P_{t+1} \cap R_{s'}$
3. ***If*** $P_R \neq \emptyset$
4.   ***For each*** *page r* ***in*** P_R
5. $r(s,a,s') = r(s',P_{t+1}) +=$ $reward(Dist(R_{s'}, r), Time(P_{t+1}))$
6.   ***End For***
7. ***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 our 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} and the actions in that state that frequently lead to S_i. This alternative function is used to reduce the number of the states dramatically by reducing *w'* [16].

2.2 Observations on System Performance

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 [2,11] when new pages are added to the website. In situations like these the system was unable to make any decisions regarding the pages to recommend to the users. 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 our state space cannot guarantee a high quality recommendation, to be more accurate it can be said that even a high Q-value cannot guarantee a high quality recommendation by itself. Simply put, when a pattern has few occurrences in the

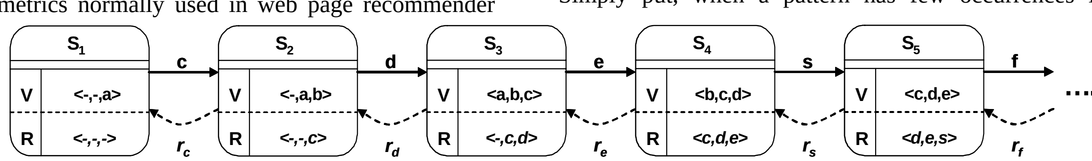

Figure 1- States and actions in the recommendation problem

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 [10]. 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. In summary, 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 need or behavior.

This is actually 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 (or URL). The system relies solely on usage data and thus is unable to make any generalization regarding the usage data it has. One common solution to this problem is to incorporate some semantic knowledge about the items being recommended into the system. This enables the system to infer higher level knowledge from the usage patterns which can prevent the mentioned problems.

3. INCORPORATING CONCEPT HIERARCHIES IN THE MODEL

One successful approach used to enhance web usage mining, is exploiting content information to transform the raw log files into more meaningful semantic logs [5,1] and then applying data mining techniques on them. In a typical scenario pages are mapped to higher level concepts e.g. catalogue page, product page, etc and a user session consisting of sequential pages will be transformed to a sequence of concepts followed by the user. Consequently, generalized patterns are extracted from these semantically enhanced log files which can then be used for personalization.

We decided to exploit the same techniques in our system to improve our state and action model. In order to make our solution both general and applicable, we avoided using an ad-hoc concept hierarchy for this purpose. Instead we chose to exploit hierarchical and conceptual document clustering which can provide us with semantic relationships between pages without the need of a specifically devised ontology, concept hierarchy or manual assignment of concepts to pages. An important factor in our selection was the ability of the method to perform incremental document clustering, since we prefer to come up with a solution that is able to cope with the changes in the web site content and structure. In order to map pages to higher level concepts, we applied the DCC [8] clustering algorithm on the web pages. It is an incremental hierarchical clustering algorithm which is originally devised to infer user needs and falls in the category of conceptual clustering algorithms as it assigns labels to each cluster of documents. We use this method to organize our documents similar to the manner in which they're assigned to nodes of a concept hierarchy. It should be noted that the output of other more sophisticated approaches like the one proposed in [6] for generating C-Logs could also be used for this purpose without affecting our general RL model.

3.1 Conceptual States and Actions

After clustering the web pages in the hierarchy, our state and action definition change as follows. Instead of keeping a sequence *V* of individual page visits by the user, each state would consist of a sequence of concepts visited by the user. Also the actions are now recommendation of pages that belong to a specific concept. In order to do so we need a module to find the node each page belongs to in the concept hierarchy and transform each usage log to a sequence of concepts in the training phase. The other aspects of the system like the reward function and the learning process would remain the same, e.g. an action *a* recommending a concept *c* is rewarded if the user visits a page belonging to concept c later in his browsing session.

This definition results in a much smaller state-action space as now the state space size is dependant on the number of distinct page clusters instead of the number of distinct web pages in the website. Consequently, the learning process will become more efficient and the system will have a more general model of users' browsing behavior on the site. With this generalized definition, the chance of confronting an unseen state will be much less and actually minimized as our evaluation results show. We'll no longer make decisions based on weak usage patterns as now the states represent a generalized view of many single visit sequences, and the average number of times a state is visited in user sessions is now 10.2 times the average visit of states in the usage-based setting. A general view of the system is depicted in Figure 2.

In the test phase, the user's raw session will be converted to a semantic session, the corresponding state will be found and the page cluster with the highest value is identified. When a concept is chosen as the action, the next step would be to recommend pages from the chosen cluster(s). Initially we chose to recommend the pages with a probability corresponding to their similarity to the cluster mean vector. This new definition of actions enables the system to cover a wider range of pages to be recommended as our evaluations show, and also the potential ability of avoiding the "new item" problem as any new page will be categorized in the appropriate cluster and have a fair chance of being recommended.

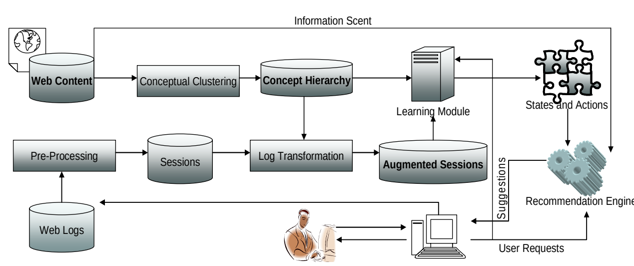

Figure 2- Architecture of the Recommender System

3.2 A Content-Based Reward Function

We can also make use of the content information of web pages and their relative positioning in the concept hierarchy in our reward function. The new rewarding function takes the content similarity of the recommended and visited pages into account. The basic idea behind this method is to reward recommendation of a concept c in S_i which might not be visited in S_{i+1} but is semantically similar to the visited page v, or more precisely, to the concept that v belongs to. The reward function $r(s,a,s')$ will be as shown in Equation 2.

$$UBR(Dist(R_{s'}, r), Time(P_{t+1})) \times CBR(r, C(P_{t+1})) \quad (2)$$

Here *CBR* represents the content-based reward of an action and *UBR* is the usage based reward which is our previous reward function. In order to compute the content based reward we use the method for computing similarity of nodes in a concept hierarchy proposed in [1].

In this method a probability $p(c)$ is assigned to each concept node c which is proportional to the frequency of pages belonging to this node and its descendants in user sessions. The information content of each node is then defined as:

$$I(c) = -\log p(c) \quad (3)$$

Then a set LCA is found which contain the least common ancestors, those occurring at the deepest level, of the pair of concept nodes. And the similarity score between those are computed as:

$$Sim(c_1, c_2) = \max_{a \in LCA}\{I(a)\} \quad (4)$$

The *CBR* will be equal to the similarity score. This method seems specifically appropriate for the off-line training phase where recommendations are evaluated using the web usage logs. In this phase actions are predictions of the user's next visit and web pages are not recommended to the user in the on-line browsing sessions. As a result actual user reactions towards pages cannot be assessed and the assumption is made that users interest toward a recommendation can be estimated as function of conceptual similarity between the recommended and visited pages.

The situation is a bit different when the system provides on-line recommendations to the user. Here the usage-based reward is weighed more heavily than the reward based on content similarity. This is based on the idea that the overall judgment of users can be trusted more than the content similarity of pages, since satisfying user information need is the ultimate goal of personalization.

3.3 Selection of Pages in a Concept

Based on the actions, we can decide which concept the user is interested in. In order to make recommendations, we should select a page belonging to that concept, which is not a trivial task especially when we're faced with large clusters of pages. Our initial solution was to rank pages with respect to their distance from cluster center. Our experiments show that this method does not yield in accurate recommendations. In order to enhance our method we exploited the content information of web pages and the hyperlinks that the users have followed in each state. The text around the chosen hyperlinks in each page has been used as an indicator of user information need in user modeling, based on the information scent model [3]. We also employ the information scent to compute a vector representing user information need in each state. The method we used is basically similar to [3], using the text around the hyperlink, the title of the out going page etc., with the exception that we assign more weight to the hyperlinks followed later in each state. After computing this vector we use the cosine based similarity to find the most relevant pages in each selected page cluster for recommendation.

4. EXPERIMENTAL EVALUATION

4.1 Experimental Evaluation Setting

We evaluated system performance in the different settings described above. We used simulated log files generated by a website traffic simulator to tune our rewarding functions. The log files were simulated for a website containing 700 web pages. We pruned user sessions with a length smaller than 5 and were provided with 16000 user sessions with average length of eight. As our evaluation data set we used the web logs of the university website. This dataset contains 15000 sessions and 673 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.

4.2 Evaluation Metrics

We used the metrics proposed in [1] to evaluate our system performance. We also used some modifications of these metrics as needed. The metrics used are: *Hit Ratio (HR)*: Percentage of Hits, *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.

4.3 Sensitivity to Visited Sequence Window Size

The first experiments were performed to evaluate system sensitivity to the size of visited concept sequence V in our states. To evaluate the choice of different window sizes, regardless of other parameters e.g. the page selection method, we used a new metric called *Concept Hit Ratio (CHR)* and *Concept Predictive Ability (CPA)* which are based on recommendation and visit of concepts instead of pages. Our evaluations indicate the best performances are achieved when using window sizes of 3 and 4 (Table 1). This is due to the fact that smaller values of w keep insufficient information about navigation history and larger values of w result in states that are numerous and less frequently visited, as the average session length in our data is 8.6. We choose w=3 in the rest of the experiments as it results in smaller number of states with a negligible decrease in accuracy.

Table 1: Comparison of different window sizes

Window Size	Metric		
	CHR	CPA	RQ
1	42	76	1.88
2	63	81	3.21
3	79.50	96	2.80
4	81.30	98	2.11
5	66.66	95	3.78

Table 2: Comparison of different recommendation methods

Method	Metric			
	HR	PA	CR	RQ
UB-RL	49.81	55.06	13.17	2.21
CIS	32.11	67.12	7.21	3.90
HCM	33.09	91.01	12.56	3.76
HFreq	42.12	93.40	26.80	3.31
HIS	46.28	94.10	25.76	2.89

4.4 Comparison with Other Methods

We compared the proposed method with the previous usage-based approach *(UB-RL*) and a content-based approach that uses the info scent model to recommend pages from the whole website *(CIS)*. The latter method was used because of the promising results achieved in the system while using the page selection method based on information scent. Note that *UB-RL* has shown superior results than common usage-based methods [16]. We used three different methods for page selection in our hybrid approach: based on the distance from cluster mean *(HCM)*, using the frequency of occurrence in user sessions *(HFreq)* and the one based on Information Scent *(HIS)*. The results presented here are based on having 5 as the maximum number of recommendations in each stage; we also experimented with 3 and 5 as the thresholds which resulted in the same relative performance of the methods. As our evaluation shows (Table 2), *HIS* out performs the rest of the methods except with respect to *HR*, compared to *UB*-RL. Note that the *UB-RL* method shows a much lower *PA*, as it's a purely usage-based approach. The rather low *HR* value achieved by *HCM* indicates the importance of page selection method in the process. It is also an indicator of the existing trade-off between generalized and detailed knowledge. As we can see this approach has a high *CHR* value (Table 1), but because of the information loss occurred at a higher level of abstraction and lack of an appropriate page selection method (at lower level of abstraction), it performs even worse than *CFreq* which is based on a rather simple metric, i.e. popularity of a page. The weaker performance of *CIS* might be considered as evidence in support of the importance of usage patterns in accurate inference of user information needs.

5. CONCLUSION

In this paper we described a method to enhance a Reinforcement Learning technique, devised for web recommendations from web usage data. We showed the restrictions that a usage-based system inherently suffers from and demonstrated how combining conceptual information regarding the web pages can improve the system. Our evaluation results show the flexibility of the RL paradigm to incorporate different sources of information to improve the quality of recommendations.

This work can be extended in various ways. One is to find more sophisticated methods for organizing a website into a concept hierarchy. More accurate methods of assessing implicit feed-back can also be used to derive a more precise reward function. 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, devising a model to infer higher level goals of user browsing, similar to the work done in categorizing search activities can be another future direction.

6. ACKNOWLEDGMENTS

We would like to thank Professor Francesco Ricci for his insightful comments on this work.

7. REFERENCES

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

[2] Burke, R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 2002.

[3] Chi, E. H., Pirolli, P., Pitkow, J. Using Information Scent to Model User Information Needs and Actions on the Web. Proceedings of Human Factors in Computing Systems, 2001.

[4] Deshpande, M., Karypis, G. Item-based top-N recommendation algorithms. ACM Transactions on Information Systems (TOIS), 2004.

[5] Eirinaki, M., Vazirgiannis, M., Varlamis, I. SEWeP: Using Site Semantics and a Taxonomy to Enhance the Web Personalization Process, in Proc. of the 9th SIGKDD Conf. 2003.

[6] Eirinaki, M., Lampos, C., Paulakis, S., Vazirgiannis, M. Web Personalization Integrating Content Semantics and Navigational Patterns. In Proceedings of the sixth ACM workshop on Web Information and Data Management WIDM 2004.

[7] Fu, X., Budzik, J., Hammond, K. J. Mining navigation history for recommendation. Intelligent User Interfaces, 2000.

[8] Godoy, D., Amandi, A. Modeling user interests by conceptual clustering. Special Issue on the Semantic Web and Web Services, Information Systems Journal, , 2005.

[9] Li, J., Zaiane, O. R. Combining Usage, Content and Structure Data to Improve Web Site Recommendation, 5th International Conference on Electronic Commerce and Web, 2004

[10] Mobasher, B., Cooley, R., Srivastava, J. Automatic Personalization based on Web Usage Mining. Communications of the ACM. 43 (8), pp. 142-151, 2000.

[11] Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J. Integrating web usage and content mining for more effective personalization. In EC-Web, pages 165–176, 2000.

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

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

[14] Srivastava, J., Cooley, R., Deshpande, M., Tan, P.N. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations, 1(2):12–23, 2000.

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

[16] Taghipour, N., Kardan, A., Shiry Ghidary, S. Usage-Based Web Recommendations: A Reinforcement Learning Approach. Proceedings of the 1st ACM Recommender Systems Conference. 2007.