Web Information Personalization: Challenges and Approaches*

Cyrus Shahabi[1] and Yi-Shin Chen[2]

[1] Department of Computer Science, University of Southern California,
Los Angeles, CA 90089-2561, USA
shahabi@usc.edu
http://infolab.usc.edu/
[2] Integrated Media Systems enter, University of Southern California,
Los Angeles, CA 90089-2561, USA
yishinc@imsc.usc.edu

Abstract. As the number of web pages increases dramatically, the problem of the information overload becomes more severe when browsing and searching the WWW. To alleviate this problem, personalization becomes a popular remedy to customize the Web environment towards a user's preference. To date, *recommendation systems* and *personalized web search systems* are the most successful examples of Web personalization. By focusing on these two types of systems, this paper reviews the challenges and the corresponding approaches proposed in the past ten years.

1 Introduction

The World Wide Web (WWW) is emerging as an appropriate environment for business transactions and user-organization interactions, because it is convenient, fast, and cheap to use. The witness to this fact is the enormous popularity of e-Commerce and e-Government applications. However, since the Web is a large collection of semi-structured and structured information sources, Web users often suffer from information overload. To alleviate this problem, personalization becomes a popular remedy to customize the Web environment for users.

Among all personalization tools, recommendation systems are the most employed tools in e-commerce businesses. Recommendation systems are usually used to help the customers to locate the products they would like to purchase. In essence, these systems apply data analysis techniques to progressively generate a list of recommended products for each online customer. The most famous example in e-commerce is the "*Customers who bought*" feature used in Amazon.com™, which is basically applied to every product page on its websites. With the help of this feature, the Amazon.com™'s system recommends similar products to the current buyer based on the purchase histories of previous customers who bought the same product.

* This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and IIS-0082826, and unrestricted cash gifts from Microsoft, NCR, and Okawa Foundation.

Contrary to the recommendation systems, the personalized web search systems[1] have received little attention from the e-commerce domain, even though search engines have become the indispensable tools in our daily lives. Generally, most modern search engines, e.g., Google[2], Yahoo![3], and AltaVista[4] do not return personalized results. That is, the result of a search for a given query is identical, independent of the user submitting the query. Hence, by ignoring the user's preferences during the search process, the search engines may return a large amount of irrelevance data. To illustrate, consider the search query for the keywords "web usage". By this query, some users may look for the information regarding the usage of the *spider* web, while other users may be interested in documents related to the statistical data about World Wide Web usage.

In summary, both the recommendation systems and the personalized web search systems face the same obstacle of "ambiguity" in users' needs[5]. Moreover, both types of systems share the same challenge of striking a compromise between the amount of processed data and the efficiency of the retrieval process. We review the challenges and the proposed approaches for both system types in the remainder of this paper. Section 2 reviews the work on the recommendation systems. In Section 3, the work on the search systems is discussed.

2 Recommendation Systems

Various statistical and knowledge discovery techniques have been proposed and applied for recommendation systems. To date, most recommendation systems are designed either based on *content-based filtering* or *collaborative filtering*. Both types of systems have inherent strengths and weaknesses, where content-based approaches directly exploit the product information, and the collaboration filtering approaches utilize specific user rating information.

2.1 Content-based Filtering

Content-based filtering approaches are derived from the concepts introduced by the Information Retrieval (IR) community. Content-based filtering systems are usually criticized for two weaknesses:

1. **Content Limitation**: IR methods can only be applied to a few kinds of content, such as text and image, and the extracted features can only capture certain aspects of the content.

[1] Note that the web search system is a more general term than *search engine* where the search system includes search engines, search agents, and metasearch systems.
[2] http://www.google.com
[3] http://www.yahoo.com
[4] http://www.altavista.com
[5] The ambiguity comes from user perceptions or the disagreements among users' opinions.

2. **Over-Specialization**: Content-based recommendation system provides recommendations merely based on user profiles. Therefore, users have no chance of exploring new items that are not similar to those items included in their profiles.

2.2 Collaborative Filtering

The collaborative filtering (CF) approach remedies for these two problems. Typically, CF-based recommendation systems do not use the actual content of the items for recommendation. Collaborative filtering works based on the assumption that if user x interests are similar to user(s) y interests, the items preferred by y can be recommended to x. Moreover, since other user profiles are also considered, user can explore new items. The nearest-neighbor algorithm is the earliest CF-based technique used in recommendation systems [16, 17]. With this algorithm, the similarity between users is evaluated based on their ratings of products, and the recommendation is generated considering the items visited by nearest neighbors of the user. In its original form, the nearest-neighbor algorithm uses a two-dimensional user-item matrix to represent the user profiles. This original form of CF-based recommendation systems suffers from three problems:

1. **Scalability**: The time complexity of executing the nearest-neighbor algorithm grows linearly with the number of items and the number of users. Thus, the recommendation system cannot support large-scale applications such as Amazon.com™, which provides more than 18 million unique items for over 20 million users.
2. **Sparsity**: Due to large number of items and user reluctance to rate the items, usually the profile matrix is sparse. Therefore, the system cannot provide recommendations for some users, and the generated recommendations are not accurate.
3. **Synonymy**: Since contents of the items are completely ignored, latent association between items is not considered for recommendations. Thus, as long as new items are not rated, they are not recommended; hence, false negatives are introduced.

In order to solve these problems, a variety of different techniques have been proposed. Some of techniques, such as dimensionality reduction [11, 8], clustering [29], and Bayesian Network [10, 9], mainly are remedies for the scalability problem. These techniques extract characteristics (patterns) from the original dataset in an offline process and employ only these patterns to generate the recommendation lists in the online process. Although this approach can reduce the online processing cost, it often reduces the accuracy of the recommending results. Moreover, the online computation complexity keeps increasing with the number of patterns.

Some other techniques, such as association rules [30, 11], content analysis [12, 13, 15], categorization [18, 14], are emphasized on alleviating the sparsity and synonymy problems. Basically, these techniques analyze the Web usage data

(from Web server logs) to capture the latent association between items. Subsequently, based on both item association information and user ratings, the recommendation systems can thus generate better recommendation to users. However, the online computation time concurrently increases, as more data are incorporated into the recommendation progress. Additionally, because Web usage data from the server side are not reliable [24], the item association generated from Web server logs might be wrong.

2.3 Yoda

In an earlier work [1], we introduced a hybrid recommendation system - *Yoda*, which simultaneously utilizes the advantages of *clustering*, *content analysis*, and *collaborate filtering* (CF) approaches. Basically, Yoda is a two-step approach recommendation system. During the offline process, Yoda generates cluster recommendation lists based on the Web usage data from the client-side through clustering and content analysis techniques. This approach not only can address the scalability problem by the preprocessing work, but also can alleviate the sparsity and synonymy problems by discovering latent association between items. Since the Web usage data from the client-side can capture real user navigation behaviors, the item association discovered by the Yoda system would be more accurate. Beside the cluster recommendation lists, Yoda also maintains numerous recommendation lists obtained from different experts, such as human experts of the Website domain, and the cluster representatives of the user ratings. By these additional recommendation lists, Yoda is less impacted by the preprocessing work as compared to other systems.

During the online process, for each user who is using the system, Yoda estimates his/her confidence values to each expert, who provides the recommendation list, based on his/her current navigation behaviors through the PPED distance measure [23] and our GA-based learning mechanism. Subsequently, Yoda generates customized recommendations for the user by aggregating across recommendation lists using the confidence value as the weight. In order to expedite the aggregation step, Yoda employs an optimized fuzzy aggregation function that reduces the time computation complexity of aggregation from $O(N \times E)$ to $O(N)$, where N is the number of recommended items in the final recommendation list to users and E is the number of recommendation lists maintained in the system. Consequently, the online computation complexity of Yoda remains the same even if number of recommendation lists increases.

In sum, the time complexity is reduced through a model-based technique, a clustering approach, and the optimized aggregation method. Additionally, due to the utilization of content analysis techniques, Yoda can detect the latent association between items and therefore provides better recommendations. Moreover, Yoda is able to collect information about user interests from implicit web navigation behaviors while most other recommendation systems [16, 17, 11, 9, 10] do not have this ability and therefore require explicit rating information from users. Consequently, Yoda puts less overhead on the users.

Since content analysis techniques only capture certain characteristics of products, some desired products might not be included in the recommendation lists produced by analyzing the content. For example, picking wines based on brands, years, and descriptors might not be adequate if "smell" and "taste" are more important characteristics. In order to remedy for this problem, in [2] we extended Yoda to incorporate more recommendation lists than just web navigation patterns. These recommendation lists can be obtained from various experts, such as human experts and clusters of user evaluations.

Meanwhile, because PPED is specially designed for measuring the similarity between two web navigation patterns including related data such as browsed items, view time, and sequences information, it can only be used for estimating confidence values to navigation-pattern clusters. Therefore, a learning mechanism is needed for obtaining the complete confidence values of an active user toward all experts. We proposed a learning mechanism that utilizes users' relevance feedback to improve confidence values automatically using genetic algorithms (GA) [5].

To the best of our knowledge, only a few studies [4, 3] incorporate GA for improving the user profiles. In these studies, users are directly involved in the evolution process. Because users have to enter data for each product inquiry, they are often frustrated with this method. On the contrary, in our design, users are not required to offer additional data to improve the confidence values. These confidence values are corrected by the GA-based learning mechanisms using users' future navigation behaviors. Our experimental results indicated a significant increase in the accuracy of recommendation results due to the integration of the proposed learning mechanism.

3 Personalized Web Search Systems

A variety of techniques have been proposed for personalized web search systems. These techniques, which are adopted from IR systems, face a common challenge, i.e., evaluating the accuracy of retrieved documents. The common evaluation method applied in IR systems is *precision and recall*, which usually requires relevance feedback from users. However, obtaining relevance feedback explicitly from users for personalized web search systems is extremely challenging due to the large size of WWW, which consists of billions of documents with a growth rate of 7.3 million pages per day [33]. Therefore, it is very time consuming and almost impossible to collect relevance judgments from each user for every page resulting from a query.

In order to incorporate user preferences into search engines, three major approaches are proposed: *personalized page importance*,*query refinement*, and *personalized metasearch systems*. Consider each approach in turn.

3.1 Personalized Page Importance

In addition to the traditional text matching techniques, modern web search engines also employ the importance scores of pages for ranking the search results.

The most famous example is the *PageRank* algorithm , which is the basis for all web search tools of Google [34]. By utilizing the linkage structure of the web, PageRank computes the corresponding importance score for each page. These importance scores will affect the final ranking of the search results. Therefore, by modifying the importance equations based on user preference, the PageRank algorithm can create a personalized search engine.

Basically, personalized importance scores are usually computed based on a set of favorite pages defined by users. In *topic-sensitive PageRank* [36], the system first pre-computes web pages based on the categories in *Open Directory*. Next, by using the pre-computation results and the favorite pages, the system can retrieve "topic-sensitive" pages for users. The experimental results [36] illustrated that this system could improve the search engine. However, this technique is not scalable, since the number of favorite pages is limited to 16 [35].

With the aim of constructing a scalable and personalized PageRank search engine, Jeh and Widom [35] proposed a model based on *personalized PageRank vector (PPV)*. PPV represents the distribution of selection in the model. The selection of PPV prefers pages related to input favorite pages. For example, the pages linked by the favorite pages and the pages linked to these favorite pages have higher selected possibilities. Each PPV can be considered as a personalized view of the importance of pages. Therefore, by incorporating PPV during the selection process, the search engine can retrieve pages closer to user preferences.

In general, since these techniques require direct inputs from users, the system increases the usage overhead. As a result, instead of saving time from identifying relevant web pages, users could possibly spend more time to personalize the search.

3.2 Query Refinement

Instead of modifying the algorithms of search engines, researchers [37–40] proposed assisting users with the query refinement process. Generally, the query refinement process of these systems consists of three steps.

1. **Obtaining user profiles from user**: The user profiles could be explicitly entered by users or implicitly learned from user behaviors. For example, WebMate [39] automatically learns the users' interested domains through a set of interesting examples; Persona [40] learns the taxonomy of user interests and disinterests from user's navigation history; the system proposed by Liu et al. [38] can learn user's favorite categories from his/her search history. Different from these systems, the client-side web search tool proposed by Chau et al. [37] requires direct inputs about interesting phrases from users.
2. **Query modification**: The systems first adjust the input query based on the corresponding user profile. Subsequently, the modified query is outsourced to search engines. For instance, the system proposed by Liu et al. [38] maps the input query to a set of interesting categories based on the user profile and confines the search domain to these categories. In Websifter [42], after a

user submits his/her intent , Websifter formulates the query based on user's search taxonomy and then submits the query to multiple search engines[6].

3. **Refinement**: After receiving the query results from the search engine, the systems refine the response. Occasionally, some search systems would further filter the irrelevant pages. For example, in the Persona system [40], the search results are ranked according to authoritativeness with a graph based algorithm. The returned set in Persona only contains the top n documents. Furthermore, Persona would refine the results if the user provides positive or negative feedback on the response.

In general, maintaining efficiency is the major challenge of the query refinement approach. That is, the time complexity of the proposed techniques grows with the size of user profiles, e.g., the number of interested categories, keywords, and domains.

3.3 Personalized Metasearch Systems

It has been reported [41] that the search engine coverage decreases steadily as the estimated web size increases. In 1999, no search engine can index more than 16% of the total web pages. Consequently, searching data by employing only a single search engine could result in a very low retrieval rate. To solve this problem, metasearch systems, such as MetaCrawler[7], Dogpile[8], and McFind[9], are proposed to increase the search coverage by combining several search engines.

Ideally, by merging various ranked results from multiple search engines into one final ranked list, metasearch systems could improve the retrieval rate. However, since metasearch systems expand the search coverage, the information overload problem could possibly be intensified. In order to improve the accuracy of returned results, researchers proposed different techniques for incorporating user preferences into metasearch systems.

The first type of personalized metasearch systems [37, 42, 45] adopt the query refinement approach. Typically, these metasearch systems modify the input query based on the corresponding user profile. Some systems[37, 45] can further select the outsourcing search engines based on user's intent. Since these systems exploit the query refinement approach, they also inherit the scalability problem from the query refinement approach.

The second types of personalized metasearch systems [43, 44] emphasize on the merging procedures. By considering user preferences during the merging process, the systems could retrieve different documents even with the same set of input lists from search engines. For example, in Inquirus 2 [44], users can assign (explicitly or implicitly) weights to different search engines and categories. The final rankings of results in Inquirus 2 are aggregated with a weighted average

[6] Note that aggregating the results from different search engines is the problem of *metasearch*, which is described later in Section 3.3.

[7] http://www.metacrawler.com/

[8] http://www.dogpile.com/

[9] http://www.mcfind.com/

process. For another instance, the personalized metasearch engine proposed by Zhu et al. [43] merges the lists based on explicit relevance feedback. In this system, users can assign "good" or "bad" scores to returned pages. With content-based similarity measure, the system could evaluate final scores to all pages. Note that the importance degrees of search engines are not considered in this merging technique.

In general, most metasearch systems emphasize on one-phase merging process, i.e., the system only considers the final score of each page returned from a search engine. However, the final score provided by each search engine is composed of several similarity values, where each value corresponds to a feature. For instance, the similarity values can be derived based on the corresponding titles of the pages, the URLs of the pages, or the summaries generated by the search engine. For another example, assume the query submitted by the user is "SARS WHO", the metasearch system can obtain different scores from the same search engine with similar queries (e.g., "SARS WHO", "SARS and WHO organization", "SARS on Who magazine", and "Severe Acute Respiratory Syndrome and WHO organization") that are generated by a query modification process. Therefore, merging these query scores based on user preferences should also be considered.

In our recent work [46], we introduced a new concept, *two-phase decision fusion*, where scores returned from the search engines are aggregated based upon user perceptions on both search engines and the relevant features. Our experimental results indicate that as compared to a traditional decision fusion approach, the retrieval accuracy of the two-phase decision fusion approach is significantly improved.

References

1. Shahabi, C., Banaei-Kashani, F., Chen Y.-S., McLeod, D.: Yoda: An Accurate and Scalable Web-based Recommendation System. In Proceedings of Sixth International Conference on Cooperative Information Systems (2001)
2. Shahabi, C., Chen, Y.-S.: An Adaptive Recommendation System without Explicit Acquisition of User Relevance Feedback. Distributed and Parallel Databases, Vol. 14. (2003) 173–192
3. Moukas, A.: Amalthea: Information discovery and filtering using a multiagent evolving ecosystem. In Proceedings of 1st Int. Conf. on The Practical Applications of Intelligent Agents and MultiAgent Technology (1996)
4. Sheth, B., Maes, P.: Evolving Agents for Personalized Information Filtering. *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications* (1993)
5. Holland, J.: *Adaption in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, Michigan
6. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: Applying Collaborative Filtering to Usenet News. Communications of the ACM Vol. 40 (3) (1997)
7. Shahabi, C., Zarkesh, A.M., Adibi, J., Shah, V.: Knowledge Discovery from Users Web Page Navigation. In *Proceedings of the IEEE RIDE97 Workshop* (1997)

8. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Application of Dimensionality Reduction in Recommender System – A Case Study. In *Proceedings of ACM WebKDD 2000 Web Mining for e-Commerce Workshop* (2000)
9. Kitts, B., Freed D., Vrieze, M.: Cross-sell, a fast promotion-tunable customer-item recommendation method based on conditionally independent probabilities. In*Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (2000) 437-446
10. Breese, J., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (1998) 43–52
11. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of Recommendation Algorithms for e-Commerce. In *Proceedings of ACM e-Commerce 2000 Conference* (2000)
12. Balabanovi, M., Shoham, Y.: Fab, content-based, collaborative recommendation. Communications of the ACM, Vol 40(3) (1997) 66–72
13. Balabanovi, M.: An Adaptive Web page Recommendation Service. In *Proceedings of Autonomous Agents* (1997) 378-385
14. Kohrs, A., Merialdo, B.: Using category-based collaborative filtering in the Active WebMuseum. In *Proceedings of IEEE International Conference on Multimedia and Expo*, Vol 1 (2000) 351–354
15. Lieberman, H., Dyke, N., Vivacqua, A.: Let's Browse, A Collaborative Browsing Agent. Knowledge-Based Systems, Vol 12 (1999) 427–431
16. Shardanand, U., Maes, P.: Social Information Filtering, Algorithm for automating "Word of Mouth". In *Proceedings on Human factors in computing systems*(1995) 210–217
17. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens, An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM conference on Cumputer-Supported Cooperative Work* (1994) 175–186
18. Good, N., Schafer, J., Konstan, J., Borchers, J., Sarwar, B., Herlocker, J., Riedl, J.: Combining Collaborative Filtering with Personal Agents for Better Recommendations. In *Proceedings of the 1999 Conference of the American Association of Artifical Intelligence* (1999) 439–446
19. Pazzani, M., Billsus, D.: Learning and Revising User profiles: The Indentification of Interesting Web Sites. Machine Learning, Vol 27 (1997) 313–331
20. Tan, A., Teo, C., Learning User Profiles for Personalized Information Dissemination. In *Proceedings of Int'l Joint Conf. on Neural Network* (1998) 183–188
21. Lam, W., Mukhopadhyay, S., Mostafa J., Palakal, M.: Detection of Shifts in User Interests for Personalized Information Filtering. In *Proceedings of the 19th Int'l ACM-SIGIR Conf on Research and Development in Information Retrieval* (1996) 317–325
22. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation, and Machine Learning. Addison-Wesley, Wokingham, England (1989)
23. Shahabi, C., Banaei-Kashani, F., Faruque, J., Faisal, A.: Feature Matrices: A Model for Efficient and Anonymous Web Usage Mining. In *Proceedings of EC-Web* (2001)
24. Shahabi, C., Banaei-Kashani, F., Faruque, J.: A Reliable, Efficient, and Scalable System for Web Usage Data Acquisition. In *WebKDD'01 Workshop in conjunction with the ACM-SIGKDD* (2001)
25. Fagin, R.: Combining Fuzzy Information from Multiple Systems. In *Proceedings of Fifteenth ACM Symposyum on Principles of Database Systems* (1996)

26. Hunter, A.: Sugal Programming manual. http://www.trajan-software.demon.co.uk/sugal.htm (1995)
27. Wu, L., Faloutsos, C., Sycara, K., Payne, T.: FALCON: Feedback Adaptive Loop for Content-Based Retrieval. In *Proceedings of Int'l. Conf. on Very Large Data Bases* (2000)
28. Knorr, E., Ng, R., Tucakov, V.: Distance-Based Outliers: Algorithms and Applications. *The VLDB Journal*, Vol 8(3) (2000) 237–253
29. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. *Communications of the ACM*, Vol 43(8) (2000) 142–151
30. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Web Data Mining: Effective personalization based on association rule discovery from web usage data. In *Proceeding of the Third International Workshop on Web Information and Data Management* (2001)
31. Rui, Y., Huang, T., Ortega, M., Mehrotra, S.: Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol 8(5) (1998) 644–655
32. Knuth, D. Seminumerical Algorithm. *The Art of Computer Programming Volume 2*, 1997
33. Lyman, P., Varian, H. R.: How Much Information . *Retrieved from http://www.sims.berkeley.edu/research/projects/how-much-info/internet.html* (2000)
34. Google: Google Technology. *Retrieved from http://www.google.com/technology/* (2003)
35. Jeh, G., Widom, J.: Scaling Personalized Web Search. *Proceedings of the 12th International World Wide Web Conference* (2003)
36. Haveliwala, T.H.: Topic-sensitive PageRank. *Proceedings of the 11th International World Wide Web Conference* (2002)
37. Chau, M., Zeng, D., Chen, H.: Personalized Spiders for Web Search and Analysis. *Proceedings of ACM/IEEE Joint Conference on Digital Libraries* (2001)
38. Liu, F., Yu, C.T., Meng, W.: Personalized web search by mapping user queries to categories. *Proceedings of CIKM* (2002)
39. Chen, L., Sycara, K.: WebMate : A Personal Agent for Browsing and Searching. *Proceedings of the 2nd International Conference on Autonomous Agents* (1998)
40. Tanudjaja, F., Mui, L.: Persona: a contextualized and personalized web search. *35th Annual Hawaii International Conference on System Sciences* (2002)
41. Lawrence, S., Giles, C.L.: Accessibility of Information on the Web . *Nature*, Vol 400 (1999) 107–109
42. Scime, A., Kerschberg, L.: WebSifter: An Ontology-Based Personalizable Search Agent for the Web . *Proceedings of International Conference on Digital Libraries: Research and Practice* (2000)
43. Zhu, S., Deng, X., Chen, K., Zheng, W.: Using Online Relevance Feedback to Build Effective Personalized Metasearch Engine. *Proceedings of Second International Conference on Web Information Systems Engineering* (2001)
44. Glover, E., Lawrence, S., Birmingham, W.P., Giles, C.L.: Architecture of a Metasearch Engine that Supports User Information Needs. *Proceedings of Eighth International Conference on Information and Knowledge Management* (1999)
45. Glover, E., Flake, G.W., Lawrence, S., Birmingham, W.P., Kruger, A., Giles, C.L., Pennock, D.M.: Improving Category Specific Web Search by Learning Query Modifications. *Proceedings of Symposium on Applications and the Internet* (2001)
46. Chen, Y.-S., Shahabi, C., Burns, G.: Two-Phase Decision Fusion Based On User Preferences. *submitted for reviewing* (2003)