Proxy Server Protection for Web Search

[1]Amrutha Ashok, [2]Ashna John, [3]Pretty Joy, [4]Reshma K Vijayan, [5]Amrutha V V, [6]Deepa K Daniel, [7]Jooby E
[1,2,3,4,5]Dept. of CSE, College of Engineering, Perumon, Panayam, Kerala
[6]Dept. of Information and Technology, College of Engineering, Perumon, Panayam, Kerala
[7]Dept. of Computer Science, College of Engineering, Perumon, Panayam, Kerala

Abstract
The web search engines has long become the most important portal for ordinary people looking for useful information on the web. But there are certain reluctance shown by the users towards disclosing their private information during searching. In this paper we present a search engine which provides privacy to the users as well as serve as a perfect search engine. We propose a framework called UPS that provides runtime generalization along with two algorithms namely GreedyDP and GreedyIL. Extensive experiments demonstrate the effectiveness of our framework.

General Terms
Runtime generalization, privacy protection, greedy alogorithms

Keywords
Generalized Profile, Online Profiler, Support, Risk

I. Introduction

Personalized web search (PWS) [1] is a general category of search techniques aiming at providing better search results, which are tailored for individual user needs. As the expense, user information has to be collected and analyzed to figure out the user intention behind the issued query.
To protect user privacy in profile-based PWS, researchers have to consider two contradicting effects during the search process. On the one hand, they attempt to improve the search quality with the personalization utility of the user profile. On the other hand, they need to hide the privacy contents existing in the user profile to place the privacy risk under control.
The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. The click-log based methods are straightforward— they simply impose bias to clicked pages in the user's query history. Although this strategy has been demonstrated to perform consistently and considerably well it can only work on repeated queries from the same user, which is a strong limitation confining its applicability. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be potentially effective for almost all sorts of queries, but are reported to be unstable under some circumstances. Although there are pros and cons for both types of PWS techniques, the profile-based PWS has demonstrated more effectiveness in improving the quality of web search recently, with increasing usage of personal and behavior information to profile its users, which is usually gathered implicitly from query history , browsing history ,click-through data ,bookmarks, user documents, and so forth.

A. Motivations

To protect user privacy in profile-based PWS, we considered two contradicting effects during the search process. On the one hand, we attempted to improve the search quality with the personalization utility of the user profile. On the other hand, there was a need to hide the private contents existing in the user profile to place the privacy risk under control. In an ideal case, it is useful to perform personalization at the expense of only a small (and less-sensitive) portion of the user profile, namely a generalized profile. Thus, user privacy can be protected without compromising the personalized search quality. In general, there is a tradeoff between the search quality and the level of privacy protection achieved from generalization.

The motivation for PWS was obtained from the current scenario, which includes:

1. The existing profile-based PWS do not support runtime profiling.
2. The existing methods do not take into account the customization of privacy requirements.
3. Many personalization techniques require iterative user interactions when creating personalized search results.

II. Working

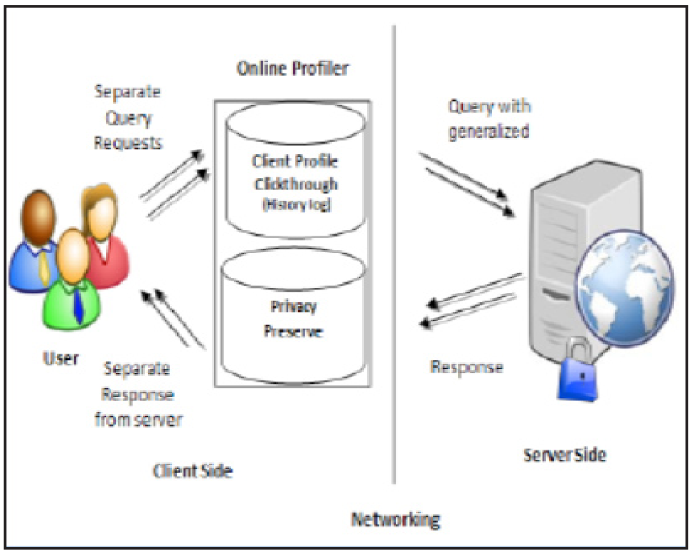


Fig. 1: System Architecture

As shown in fig. 1, UPS consists of number of clients/users and a server for fulfilling client's request. In client's machine, the online profiler is implemented as search proxy which maintains users profile in hierarchy of nodes and also maintain the user specified privacy requirement as a set of sensitive nodes. There are two phase, namely Offline and Online phase for the framework. During Offline, a hierarchical user profile is created and user specified privacy requirement is marked on it. The query fired by user is handled in the online phase as:
When user fires a query on the client, proxy generates user profile in run time. The output is generalized user profile considering the privacy requirements. Then, the query along with generalized profile of user is sent to PWS server for personalized web search. The search result is personalized and the response is sent back to

query proxy. Finally, the proxy presents the raw result or re-ranks them with user profile.

UPS is distinguished from conventional PWS in that it:

1. Provides runtime profiling, which in effect optimizes the personalization utility while respecting user's privacy requirements;
2. Allows for customization of privacy needs; and
3. Does not require iterative user interaction [2].

A. Advantages

1. PWS System increases the stability of the search quality.
2. System supports privacy by preventing unnecessary disclosure of the user profile.
3. PWS System provides runtime profiling, which in effect optimizes the personalization utility while respecting user's privacy requirements.
4. System allows for customization of privacy needs.
5. Does not require iterative user interaction.
6. System is Client side completely.

B. Limitation

1. System depends entirely upon Proxy Server.
2. Proxy Server Failure leads to the failure of the whole system.
3. System gives the results in "Text" format.
4. Only one user can Login at a time to System.

C. Modules

1. Personalization Based on Profile Module

This module personalizes digital multimedia. There is a profile generator that automatically creates user profiles. Also a content based algorithm is provided.

2. Privacy Protection Module

This module generalizes the profiles according to user-specified privacy requirements. The two generalization algorithms namely GreedyDP and Greedy IL is also presented.

3. Generalizing User Profile Module

This module is used for preprocessing the user profile. We first initialize the user profile by taking the indicated parent user profile into account. Then we add the inherited properties to the properties of the local user profile.

4. Online Decision Making Module

This module improves search quality. Online mechanism run-time profiling optimizes personalization utility. Query will be send to the server without a user profile

III. Algorithm

A. Greedy Algorithm

A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. Greedy algorithm considers easy to implement and simple approach and decides next step that provide beneficial result. In many problems, a greedy strategy does not produce an optimal solution, but a greedy heuristic yields locally optimal solutions that approximate a global optimal solution in a reasonable time.

1. Greedydp Algorithm

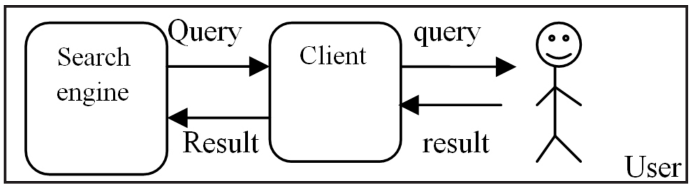
It works in a bottom up manner. Starting with the leaf node, for every iteration, it chooses leaf topic for pruning thus trying to maximize utility of output. During iteration a best profile-so- far is maintained satisfying the Risk constraint. The iteration stops when the root topic is reached. The best profile-so-far is the final result. GreedyDp algorithms require recomputation of profiles which adds up to computational cost and memory requirement.

2. Greedyil Algorithm

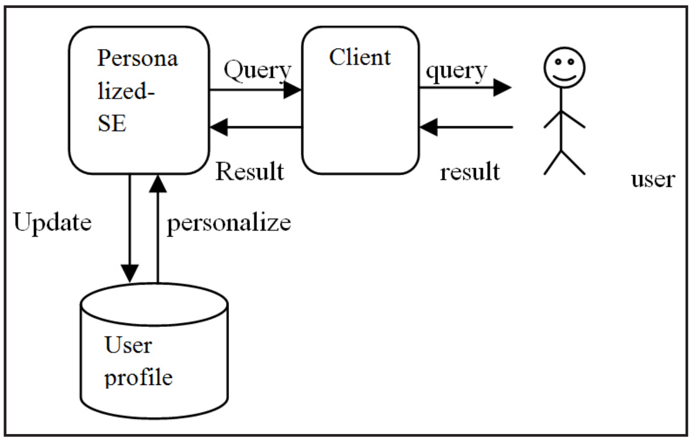
Greedy IL algorithm improves generalization efficiency. Greedy IL maintains priority queue for candidate prune leaf operator in descending order. This decreases the computational cost. Greedy IL states to terminate the iteration when Risk is satisfied or when there is a single leaf left. Since, there is less computational cost compared to GreedyDP, Greedy IL outperforms GreedyDP.

4. Software Architecture For Personalized Search

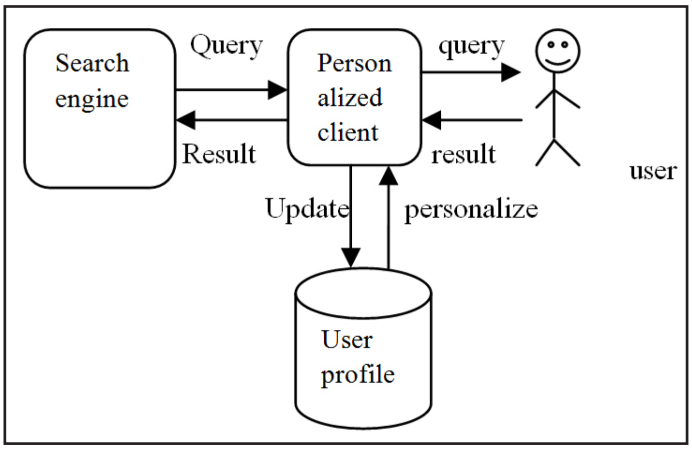
For Web search applications, server-client architecture, as shown in fig. 2 (a), is commonly adopted, where a client (often the web browser) sends queries to a server (the search engine). The search engine analyzes the user information need, looks up its index structure of documents, and returns a ranked list of search results to the client for a user to view. A search engine generally stores user search logs for various kinds of purposes including personalization and anti-spam. Thus it is to the interest of search engines not to remove the search engine logs automatically. Indeed, they tend to keep the search engine logs indefinitely. There are three kinds of software architectures that expand the basic server-client model of Web search to support personalized search. Their main differences lie in where personally identifiable information P (U) is stored and how it is exploited for personalization. In this section, we describe these three kinds of software architectures and analyze what levels of privacy preservation can be achieved with these different architectures.



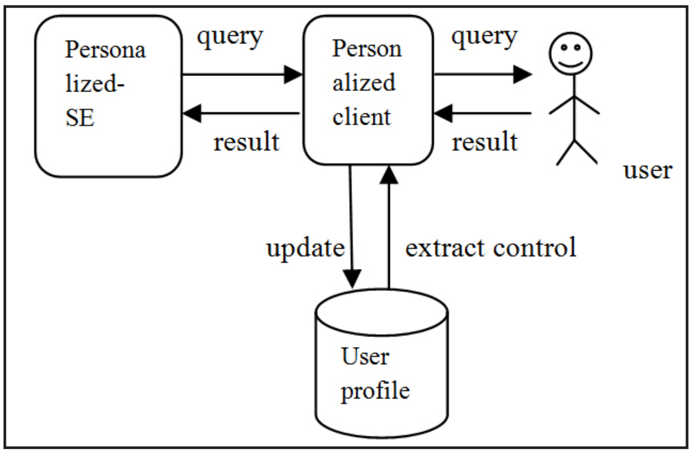
(a). No Personalization



b. Server Side Personalization



(c). Client Side Personalization



(d). Client Server Collaborative Personalization

Fig. 1: Software Architecture of Personalized Web Search

A. Server-Side Personalization

For server-side personalization as shown in fig. 2 (b), the personally identifiable information P (U) is stored on the search engine side. The search engine builds and updates the user profile either through the user's explicit input (e.g., asking the user to specify personal interests) or by collecting the user's search history implicitly (e.g., query and click through history). Both approaches require the user to create an account to identify him. But the latter approach requires no additional effort from the user and contains richer description of user information need. The advantage of this architecture is that the search engine can use all of its resources (e.g, document index, common search patterns) in its personalization algorithm. Also, the client software generally requires no changes. This architecture is adopted by some general search engines such as Google Personalized. Currently most personalized search systems with server-side personalization architecture require the user to give consent before his/her search history can be collected and used for personalization. If the user gives the permission, the search engine will hold all the personally identifiable information possibly available on the server side. Thus from the user perspective, it even does not have level I privacy protection.

Since many users fear its potential privacy infringement by search engines, this has hindered the wide adoption of personalization with this architecture. However, if the search engine decides to voluntarily replace the user identity ID (U) with a pseudo user identity IDP (U), Level I privacy protection can be achieved. When the search engines release the search engine logs to the public or a group of researchers, they generally replace user identity ID (U) by a pseudo user identity IDP (U). To the third parties receiving these search engine logs, which may use it for personalized search purpose, the user will have Level I privacy protection. If the user decides to use a proxy to communicate with the search engine, all user information going through the same proxy will be combined in a user profile. Through this method, privacy protection can be achieved. However, this method does not always work: When the search engine uses the user login ID to collect user information, this method will not achieve privacy protection; when the search engine only uses the IP address to aggregate the user information, this method works. Sometimes, search engines group users randomly or according to some criteria before they release the search engine logs. Then the user will also have privacy protection to those third parties which receive the search engine logs. It is impossible to implement privacy protection if personalization is done on the server side.

B. Client-Side Personalization

For client-side personalization as shown in fig. 2(c), the personally identifiable information is always stored on a user's personal computer. As in the case of server-side personalization, the user profile can be created from user specification explicitly or search history implicitly. The client sends queries to the search engine and receives results, which is the same as in the general web search scenario. But given a user's query, a client-side personalized search agent can do query expansion to generate a new query before sending the query to the search engine. The personalized search agent can also re- rank the search results to match the inferred user preferences after receiving the search results from the search engine. With this architecture, not only the user's search behavior but also his contextual activities(e.g., web pages viewed before) and personal information (e.g., emails, browser bookmarks) could be incorporated into the user profile, allowing for the construction of a much richer user model for personalization. The sensitive contextual information is generally not a major concern since it is strictly stored and used on the client side. Another benefit is that the overhead in computation and storage for personalization can be distributed among the clients.

A main drawback of personalization on the client side is that the personalization algorithm cannot use some knowledge that is only available on the server side (e.g., Page Rank score of a result document). UCAIR adopts the client-side personalization. With proxy functionality applied to the client side, Level II privacy protection can be achieved. If the client side uses an anonymous network such as Tor to communicate with the search engine, privacy protection can also be achieved. In order to achieve privacy protection, additional cooperation of the search engine would be needed as we described

C. Client-Server Cooperative Personalization

For the client-server cooperative personalization as shown in Fig.2 (d), it is a compromise between the previous two kinds of architectures. The user profile is still stored on the client side, but the server also participates in personalization. At query time, the client extracts contextual information from the user profile, and sends it to the search engine along with the query. The search engine then does personalization with the received context. Compared with client-side personalization, this architecture has an advantage of allowing for the use of a search engine's internal resources. The contextual information sent to the server specifies the user's search preferences (e.g., query expansion terms, topic weight vector). It is extracted from the user profile (e.g., the weight vector can be learned from search history), and is only relevant to a particular query. Therefore, it is a condensed version of the whole user profile (generally a few terms or a weight vector from a user's search history), thus the architecture can minimize the personal information obtained by the search engine.

A main drawback is that the condensed contextual information may not be as powerful as the whole user profile. We have not yet seen any personalization products in this category, probably due to the relatively complex architecture. This architecture provides the same level of privacy protection as server-side personalization. However, the personally identifiable information collectable on the server side is less than in the case of pure server-side personalization

V. Literature Survey Review

1. A LargeScale Evaluation and Analysis of Personalized Search Strategies Author: Z.Dou, R.Song, and J.-R. Wen,2007 Proc. Int"l Conf. Method: A large scale evaluation framework for personalized search based on query logs, and then evaluate five personalized search strategies (incuding two click log based and profile based)using 12-days MSR query log. Advantage: Search accuracy is evaluated by real user clicks recorded in query logs automatically. Disadvantage: Personalization may lack effectiveness on some querry
2. Implicit User Modeling for Personalized Search Author: sX. Shen,B.Tan, and C.Zhai, pro 14th ACM Int"l Conf. Information and Knowledge Management (CIKM), 2005 Method:Here present a decision theoretic framework and develop techniques for implicit user modeling in information retrieval. They develop an intelligent clientside web search agent (UCAIR) that can perform eager implicit feedback. Advantage: Search agent can improve search accuracy over the popular Google search engine. Disadvantage: They generally lack user modeling and are not adaptive to individual users.
3. Personalizing Adaptive Web Search Based on User Profile Constructed without any effort from Users. Author: K. Sugiyama, K. Hatano, and M. Yoshikawa Proc. 13th Int"l Conf., 2004. Method: Propose several approaches to adapting search results according to each user"s need for relevant information without any user effort, and then verify the effectiveness of our proposed approaches. Advantage: User"s preferences can be achieved by user profile based on modified collaborative filtering with detailed analysis of user"s browsing history in one day. Disadvantage: Each user needs different information for his/her query. Therefore, the search results should be adapted to users with different information needs.
4. Mining Long-Term Search History to Improve Search Accuracy Author: B.Tan,X.Shen, and C.Zhai,Proc. ACM SIGKDD Int"l Conf. Knowledge Discovery and Data Mining (KDD), 2006 Method: Statistical language modeling based methods to mine contextual information from long term search history. Advantage: Exploit it for a more accurate estimate of query language model. Disadvantages: The web search engines, suffer from the problem of document.

VI. Conclusion

This paper presented a framework of privacy protection called UPS. This framework could potentially be adopted by any PWS that captures user profiles in a hierarchical taxonomy. The users are allowed to specify the privacy requirements. This framework also supports online generalization to protect personal privacy without compromising the search quality.

VII. Acknowledgement

Our sincere thanks to THE ALMIGHTY GOD, and our teachers who guided us throughout our research work. Though words cannot express our gratitude completely, we would like to thank everyone who had stood as a pillar of strength and support for us. Above all we concede the surviving presence and flourishing refinement of ALMIGHTY GOD for concealed hand yet substantial supervision all through the project. We would like to thank our respected principal Dr Z A ZOYA and our respected Head of the Department Dr DHEEPA J and Assistant professor in IT Ms.DEEPA K DANIEL and Ms JOOBY E (dept of CSE) for guiding's all through our work. We sincerely thank all our staffs of the Department Of CSE for their assistance and timely advice Last but not the last; we would like to thank our parents and friends for their support.

References

[1] Bowman, M., Debray, S. K., Peterson, L. L.,"Reasoning about naming systems", 1993.

[2] Lidan Shou, He Bai, Ke Chen, Gang Chen,"IEEE transactions on knowledge and data engineering", Vol. 26, No. 2, Year 2014.

Ashna John is an Undergraduate student at College of Engineering, Perumon. Currently she is doing her final year bachelors in Computer Science and Engineering.

Reshma K Vijayan is an Undergraduate student at College of Engineering, Perumon. Currently she is doing her final year bacherlors in Computer Science and Engineering.

Pretty Joy is an Undergraduate student at College of Engineering, Perumon. Currently she is doing her final year bacherlors in Computer Science and Engineering.

Amrutha Ashok is an Undergraduate student at College of Engineering, Perumon. Currently she is doing her final year bacherlors in Computer Science and Engineering.

Amrutha V.V is an Undergraduate student at College of Engineering, Perumon. Currently she is doing her final year bacherlors in Computer Science and Engineering. Also she did her degree in Computer Applications and Management from Government Polytechnic, Malappuram.

Deepa K Daniel is currently working as Asst Prof at College of Engineering Perumon in the Information and Technology Department.