Supporting Privacy Protection in Personalized Web Search

Lidan Shou, He Bai, Ke Chen, and Gang Chen

Abstract—Personalized web search (PWS) has demonstrated its effectiveness in improving the quality of various search services on the Internet. However, evidences show that users' reluctance to disclose their private information during search has become a major barrier for the wide proliferation of PWS. We study privacy protection in PWS applications that model user preferences as hierarchical user profiles. We propose a PWS framework called UPS that can adaptively generalize profiles by queries while respecting user-specified privacy requirements. Our runtime generalization aims at striking a balance between two predictive metrics that evaluate the utility of personalization and the privacy risk of exposing the generalized profile. We present two greedy algorithms, namely GreedyDP and GreedyIL, for runtime generalization. We also provide an online prediction mechanism for deciding whether personalizing a query is beneficial. Extensive experiments demonstrate the effectiveness of our framework. The experimental results also reveal that GreedyIL significantly outperforms GreedyDP in terms of efficiency.

Index Terms—Privacy protection, personalized web search, utility, risk, profile

1 Introduction

The web search engine has long become the most important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is largely due to the enormous variety of users' contexts and backgrounds, as well as the ambiguity of texts. *Personalized web search* (PWS) 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.

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 [1], 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 [1].

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 [2], [3], [4], browsing history [5], [6], click-through data [7], [8], [1] bookmarks [9], user documents [2], [10], and so forth. Unfortunately, such implicitly collected personal data can easily reveal a gamut of user's private life. Privacy issues rising from the lack of protection for such data, for instance the AOL query logs scandal [11], not only raise panic among individual users, but also dampen the data-publisher's enthusiasm in offering personalized service. In fact, privacy concerns have become the major barrier for wide proliferation of PWS services.

1.1 Motivations

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. A few previous studies [10], [12] suggest that people are willing to compromise privacy if the personalization by supplying user profile to the search engine yields better search quality. In an ideal case, significant gain can be obtained by 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.

Unfortunately, the previous works of privacy preserving PWS are far from optimal. The problems with the existing methods are explained in the following observations:

1. *The existing profile-based PWS do not support runtime profiling.* A user profile is typically generalized for

• *The authors are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, P.R. China. E-mail: should@acm.org, bailaohe@gmail.com, {chenk, cg}@zju.edu.cn.*

Manuscript received 1 May 2012; revised 29 Aug. 2012; accepted 3 Oct. 2012; published online 11 Oct. 2012.
Recommended for acceptance by B.C. Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2012-05-0302.
Digital Object Identifier no. 10.1109/TKDE.2012.201.

only once offline, and used to personalize all queries from a same user indiscriminatingly. Such "one profile fits all" strategy certainly has drawbacks given the variety of queries. One evidence reported in [1] is that profile-based personalization may not even help to improve the search quality for some ad hoc queries, though exposing user profile to a server has put the user's privacy at risk. A better approach is to make an online decision on

a. whether to personalize the query (by exposing the profile) and
b. what to expose in the user profile at runtime.

To the best of our knowledge, no previous work has supported such feature.

2. *The existing methods do not take into account the customization of privacy requirements.* This probably makes some user privacy to be overprotected while others insufficiently protected. For example, in [10], all the sensitive topics are detected using an absolute metric called *surprisal* based on the information theory, assuming that the interests with less user document support are more sensitive. However, this assumption can be doubted with a simple counterexample: If a user has a large number of documents about "sex," the *surprisal* of this topic may lead to a conclusion that "sex" is very general and not sensitive, despite the truth which is opposite. Unfortunately, few prior work can effectively address individual privacy needs during the generalization.
3. *Many personalization techniques require iterative user interactions when creating personalized search results.* They usually refine the search results with some metrics which require multiple user interactions, such as *rank scoring* [13], *average rank* [8], and so on. This paradigm is, however, infeasible for runtime profiling, as it will not only pose too much risk of privacy breach, but also demand prohibitive processing time for profiling. Thus, we need predictive metrics to measure the search quality and breach risk after personalization, without incurring iterative user interaction.

1.2 Contributions

The above problems are addressed in our UPS (literally for User customizable Privacy-preserving Search) framework. The framework assumes that the queries do not contain any sensitive information, and aims at protecting the privacy in individual user profiles while retaining their usefulness for PWS.

As illustrated in Fig. 1, UPS consists of a nontrusty search engine server and a number of clients. Each client (user) accessing the search service trusts no one but himself/herself. The key component for privacy protection is an *online profiler* implemented as a search proxy running on the client machine itself. The proxy maintains both the complete *user profile*, in a hierarchy of nodes with semantics, and the *user-specified (customized) privacy requirements* represented as a set of *sensitive-nodes*.

The framework works in two phases, namely the *offline* and *online* phase, for each user. During the offline phase, a

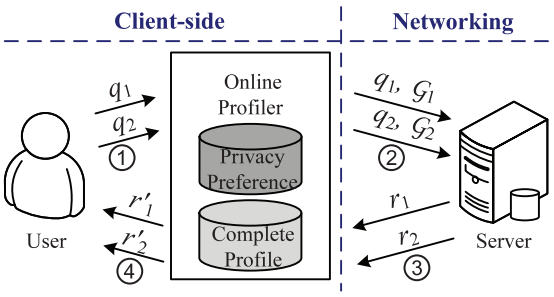

Fig. 1. System architecture of UPS.

hierarchical user profile is constructed and customized with the user-specified privacy requirements. The online phase handles queries as follows:

1. When a user issues a query q_i on the client, the proxy generates a user profile in runtime in the light of *query terms*. The output of this step is a *generalized* user profile $\mathcal{G}_i$ satisfying the privacy requirements. The generalization process is guided by considering two conflicting metrics, namely the *personalization utility* and the *privacy risk*, both defined for user profiles.
2. Subsequently, the query and the generalized user profile are sent together to the PWS server for personalized search.
3. The search results are personalized with the profile and delivered back to the query proxy.
4. Finally, the proxy either presents the raw results to the user, or reranks them with the complete 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. Our main contributions are summarized as following:

- We propose a privacy-preserving personalized web search framework UPS, which can generalize profiles for each query according to user-specified privacy requirements.
- Relying on the definition of two conflicting metrics, namely *personalization utility* and *privacy risk*, for hierarchical user profile, we formulate the problem of privacy-preserving personalized search as δ-Risk Profile Generalization, with its $\mathcal{NP}$-hardness proved.
- We develop two simple but effective generalization algorithms, GreedyDP and GreedyIL, to support runtime profiling. While the former tries to maximize the *discriminating power (DP)*, the latter attempts to minimize the *information loss (IL)*. By exploiting a number of heuristics, GreedyIL outperforms GreedyDP significantly.
- We provide an inexpensive mechanism for the client to decide whether to personalize a query in UPS. This decision can be made before each runtime profiling to enhance the stability of the search results while avoid the unnecessary exposure of the profile.
- Our extensive experiments demonstrate the efficiency and effectiveness of our UPS framework.

The rest of this paper is organized as follows: Section 2 reviews the related work, focusing on PWS and its privacy preservation. Section 3 introduces some preliminary knowledge and gives the problem statement. Section 4 presents the procedures of UPS framework. The generalization techniques used in UPS are proposed in Section 5. Section 6 further discusses some implementation issues of UPS. The experimental results and findings are reported in Section 7. Finally, Section 8 concludes the paper.

2 Related Works

In this section, we overview the related works. We focus on the literature of profile-based personalization and privacy protection in PWS system.

2.1 Profile-Based Personalization

Previous works on profile-based PWS mainly focus on improving the search utility. The basic idea of these works is to tailor the search results by referring to, often implicitly, a user profile that reveals an individual information goal. In the remainder of this section, we review the previous solutions to PWS on two aspects, namely the *representation* of profiles, and the *measure* of the effectiveness of personalization.

Many profile representations are available in the literature to facilitate different personalization strategies. Earlier techniques utilize term lists/vectors [5] or bag of words [2] to represent their profile. However, most recent works build profiles in hierarchical structures due to their stronger descriptive ability, better scalability, and higher access efficiency. The majority of the hierarchical representations are constructed with existing weighted topic hierarchy/graph, such as ODP[1] [1], [14], [3], [15], Wikipedia[2] [16], [17], and so on. Another work in [10] builds the hierarchical profile automatically via term-frequency analysis on the user data. In our proposed UPS framework, we do not focus on the implementation of the user profiles. Actually, our framework can potentially adopt any hierarchical representation based on a taxonomy of knowledge.

As for the performance measures of PWS in the literature, Normalized Discounted Cumulative Gain (nDCG) [18] is a common measure of the effectiveness of an information retrieval system. It is based on a human-graded relevance scale of item-positions in the result list, and is, therefore, known for its high cost in explicit feedback collection. To reduce the human involvement in performance measuring, researchers also propose other metrics of personalized web search that rely on clicking decisions, including *Average Precision (AP)* [19], [10], *Rank Scoring* [13], and *Average Rank* [3], [8]. We use the *Average Precision* metric, proposed by Dou et al. [1], to measure the effectiveness of the personalization in UPS. Meanwhile, our work is distinguished from previous studies as it also proposes two predictive metrics, namely *personalization utility* and *privacy risk*, on a profile instance without requesting for user feedback.

1. Open Directory Project (ODP), http://dmoz.org/.
2. Wikipedia, the Free Encyclopedia, http://www.wikipedia.org/.

2.2 Privacy Protection in PWS System

Generally there are two classes of privacy protection problems for PWS. One class includes those treat privacy as the identification of an individual, as described in [20]. The other includes those consider the sensitivity of the data, particularly the user profiles, exposed to the PWS server.

Typical works in the literature of protecting user identifications (class one) try to solve the privacy problem on different levels, including the *pseudoidentity*, the *group identity*, *no identity*, and *no personal information*. Solution to the first level is proved to fragile [11]. The third and fourth levels are impractical due to high cost in communication and cryptography. Therefore, the existing efforts focus on the second level. Both [21] and [22] provide online anonymity on user profiles by generating a group profile of k users. Using this approach, the linkage between the query and a single user is broken. In [23], the useless user profile (UUP) protocol is proposed to shuffle queries among a group of users who issue them. As a result any entity cannot profile a certain individual. These works assume the existence of a trustworthy third-party anonymizer, which is not readily available over the Internet at large. Viejo and Castellà-Roca [24] use legacy social networks instead of the third party to provide a distorted user profile to the web search engine. In the scheme, every user acts as a search agency of his or her neighbors. They can decide to submit the query on behalf of who issued it, or forward it to other neighbors. The shortcomings of current solutions in class one is the high cost introduced due to the collaboration and communication.

The solutions in class two do not require third-party assistance or collaborations between social network entries. In these solutions, users only trust themselves and cannot tolerate the exposure of their complete profiles an anonymity server. In [12], Krause and Horvitz employ statistical techniques to learn a probabilistic model, and then use this model to generate the near-optimal partial profile. One main limitation in this work is that it builds the user profile as a finite set of attributes, and the probabilistic model is trained through predefined frequent queries. These assumptions are impractical in the context of PWS. Xu et al. [10] proposed a privacy protection solution for PWS based on hierarchical profiles. Using a user-specified threshold, a generalized profile is obtained in effect as a rooted subtree of the complete profile. Unfortunately, this work does not address the query utility, which is crucial for the service quality of PWS. For comparison, our approach takes both the privacy requirement and the query utility into account.

A more important property that distinguishes our work from [10] is that we provide *personalized privacy protection* in PWS. The concept of personalized privacy protection is first introduced by Xiao and Tao [25] in Privacy-Preserving Data Publishing (PPDP). A person can specify the degree of privacy protection for her/his sensitive values by specifying "guarding nodes" in the taxonomy of the sensitive attribute. Motivate by this, we allow users to customize privacy needs in their hierarchical user profiles.

Aside from the above works, a couple of recent studies have raised an interesting question that concerns the privacy protection in PWS. The works in [1], [26] have