Asian Journal of Science and Applied Technology (AJSAT)
Vol.3.No.1 2015 pp .2-5
available at: www.goniv.com
Paper Received :08-03-2015
Paper Accepted:20-03-2015
Paper Reviewed by: 1. R. Venkatakrishnan 2. R. Marimuthu
Editor : Prof. P.Muthukumar

ENRICHING PRIVACY IN PERSONALIZED WEB SEARCH

[1]E.ARULJOTHI., [2]DR. C.RAJABHAUSHANAM, M.S, PH.D.
[1]M.E CSE (Student), [2]Professor
Tagore Engineering College, Chennai, India
aruljothi9393@gmail.com

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. I study privacy protection in PWS applications that model user preferences as hierarchical user profiles. I propose a PWS framework called UPS that can adaptively generalize profiles by queries while respecting user specified privacy requirements. My 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. I present two greedy algorithms, namely Greedy DP and Greedy IL, for runtime generalization. I 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 Greedy IL significantly outperforms Greedy DP in terms of efficiency

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

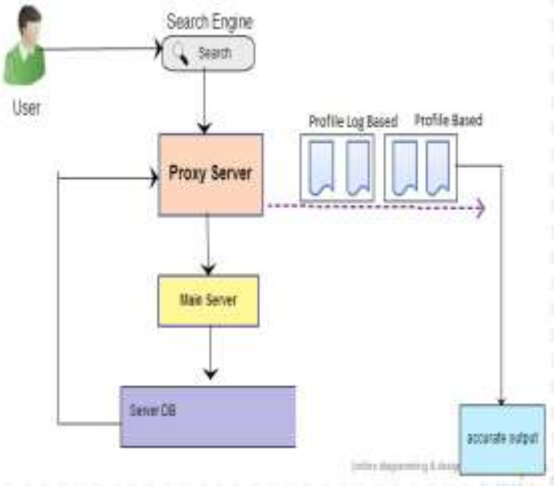
I.INTRODUCTION

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. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. In fact, privacy concerns have become the major barrier for wide proliferation of PWS services.

A.Motivations

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 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.
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.
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, average rank.



B. Contributions

The above problems are addressed in our UPS (literally for User customizable Privacy-preserving Search) framework. The framework works in two phases, namely the offline and online phase, for each user. During the offline phase, a 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 qi on the client, the proxy generates a user profile in runtime in the light of query terms.
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.

II LITERATURE SURVEY

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.

As for the performance measures of PWS in the literature, Normalized Discounted Cumulative Gain (nDCG) 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), Rank Scoring, and Average Rank. We use the Average Precision metric, proposed by Dou et al., 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.

Generally there are two classes of privacy protection problems for PWS. One class includes those treat privacy as the identification of an individual. 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 pseudo identity, the group identity, no identity, and no personal information. Solution to the first level is proved too fragile. The third and fourth levels are impractical due to high cost in communication and cryptography. Therefore, the existing efforts focus on the second level. We 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. 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 Castell_a-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.

III PROPOSED WORK

A. User Profile

Moreover, our profile is constructed based on the availability of a public accessible taxonomy, denoted as R, which satisfies the following assumption.

Assumption 1: The repository R is a huge topic hierarchy covering the entire topic domain of human knowledge. That is, given any human recognizable topic t, a corresponding node (also referred to as t) can be found in R, with the subtree subtr(t, R) as the taxonomy accompanying t.

Assumption 2: Given a taxonomy repository R, the repository support is provided by R itself for each leaf topic. In fact, Assumption 2 can be relaxed if the support values are not available. In such case, it is still possible to "simulate" these repository supports with the topological structure of R.

B. Customized Privacy Requirements

Customized privacy requirements can be specified with a number of sensitive-nodes (topics) in the user profile, whose disclosure (to the server) introduces privacy risk to the user. It must be noted that user's privacy concern differs from one sensitive topic to another.

C. Attack Model

Our work aims at providing protection against a typical model of privacy attack, namely eavesdropping. Note that in our attack model, Eve is regarded as an adversary satisfying the following assumptions:

Knowledge bounded: The background knowledge of the adversary is limited to the taxonomy repository R. Both the profile H and privacy are defined based on R.

Session bounded: None of previously captured information is available for tracing the same victim in a long duration. In other words, the eavesdropping will be started and ended within a single query session.

D. Generalizing User Profile

The generalization technique can seemingly be conducted during offline processing without involving user queries. However, it is impractical to perform offline generalization due to two reasons:

1. The output from offline generalization may contain many topic branches, which are irrelevant to a query.

2. It is important to monitor the personalization utility during the generalization. Using the running example, profiles Ga and Gb might be generalized to smaller rooted subtrees.

IV. ANALYTICAL MODEL

A. UPS Procedures- The Generalization Algorithms

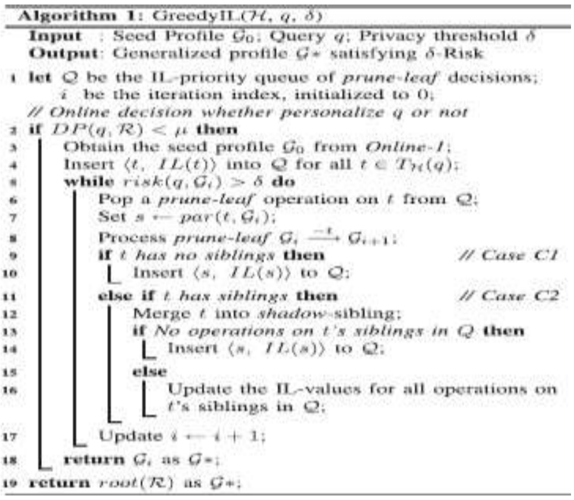
We propose two greedy algorithms, namely the GreedyDP and GreedyIL.

1. The GreedyDP Algorithm

As preliminary, we introduce an operator called prune-leaf, which indicates the removal of a leaf topic t from a profile. Formally, we denote by Gi _t _! Giþ1 the process of pruning leaf t from Gi to obtain Giþ1. The first greedy algorithm GreedyDP works in a bottomup manner. Starting from G0, in every ith iteration, GreedyDP chooses a leaf topic t 2 TGi ðqÞ for pruning, trying to maximize the utility of the output of the current iteration, namely Giþ1. During the iterations, we also maintain a bestprofile- so-far, which indicates the Giþ1 having the highest discriminating power while satisfying the _-risk constraint. The main problem of GreedyDP is that it requires recomputation of all candidate profiles generated from attempts of prune-leaf on all t 2 TGi ðqÞ. This causes significant memory requirements and computational cost.

2. The GreedyIL Algorithm

The GreedyIL algorithm improves the efficiency of the generalization using heuristics based on several findings. One important finding is that any prune-leaf operation reduces the discriminating power of the profile. In other words, the DP displays monotonicity by prune-leaf.



```
Algorithm 1: GreedyIL($\mathcal{H}$, $q$, $\delta$)
  Input  : Seed Profile $\mathcal{G}_0$; Query $q$; Privacy threshold $\delta$
  Output: Generalized profile $\mathcal{G}*$ satisfying $\delta$-Risk
let $\mathcal{Q}$ be the IL-priority queue of prune-leaf decisions;
    $i$ be the iteration index, initialized to 0;
// Online decision whether personalize $q$ or not
if $DP(q, \mathcal{R}) < \mu$ then
    Obtain the seed profile $\mathcal{G}_0$ from Online-1;
    Insert $\langle t, IL(t) \rangle$ into $\mathcal{Q}$ for all $t \in T_{\mathcal{H}}(q)$;
    while $risk(q, \mathcal{G}_i) > \delta$ do
        Pop a prune-leaf operation on $t$ from $\mathcal{Q}$;
        Set $s \leftarrow par(t, \mathcal{G}_i)$;
        Process prune-leaf $\mathcal{G}_i \xrightarrow{-t} \mathcal{G}_{i+1}$;
        if $t$ has no siblings then                          // Case C1
            Insert $\langle s, IL(s) \rangle$ to $\mathcal{Q}$;
        else if $t$ has siblings then                        // Case C2
            Merge $t$ into shadow-sibling;
            if No operations on $t$'s siblings in $\mathcal{Q}$ then
                Insert $\langle s, IL(s) \rangle$ to $\mathcal{Q}$;
            else
                Update the IL-values for all operations on
                $t$'s siblings in $\mathcal{Q}$;
        Update $i \leftarrow i + 1$;
    return $\mathcal{G}_i$ as $\mathcal{G}*$;
return $root(\mathcal{R})$ as $\mathcal{G}*$;
```

V. EXPERIMENTAL RESULTS

The UPS framework is implemented on a PC with a Pentium Dual-Core 2.50-GHz CPU and 2-GB main memory, running Microsoft Windows XP. All the algorithms are implemented in Java.

The topic repository uses the ODP web Directory. To focus on the pure English categories, we filter out taxonomies "Top/World" and "Top/Adult/World." The click logs are downloaded from the online AOL query log, which is the most recently published data we could find. The AOL query data contain over 20 million queries and 30 million clicks of 650k users over 3 months. The data format of each record is as follows: huid; query; time½; rank; url_i; where the first three fields indicate user uid issued query at timestamp time, and the last two optional fields appear when the user further clicks the url ranked at position rank in the returned results.

VI. CONCLUSION

This paper presented a client-side privacy protection framework called UPS for personalized web search. UPS could potentially be adopted by any PWS that captures user profiles in a hierarchical taxonomy. The framework allowed users to specify customized privacy requirements via the hierarchical profiles. In addition, UPS also performed online generalization on user profiles to protect the personal privacy without compromising the search quality. We proposed two greedy algorithms, namely GreedyDP and GreedyIL, for the online generalization. Our experimental results revealed that UPS could achieve quality search results while preserving user's customized privacy requirements.

VII. FUTURE WORK

For future work, we will try to resist adversaries with broader background knowledge, such as richer relationship among topics (e.g., exclusiveness, sequentiality, and so on), or capability to capture a series of queries from the victim. We will also seek more sophisticated method to build the user profile, and better metrics to predict the performance (especially the utility) of UPS.

REFERENCES

[1] *Z. Dou, R. Song, and J.-R. Wen, "A Large-Scale Evaluation and Analysis of Personalized Search Strategies," Proc. Int'l Conf. World Wide Web (WWW),pp. 581-590, 2007.*

[2] *J. Teevan, S.T. Dumais, and E. Horvitz, "Personalizing Search viaAutomated Analysis of Interests and Activities," Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 449-456, 2005.*

[3] *M. Spertta and S. Gach, "Personalizing Search Based on User Search Histories," Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence(WI),2005.*

[4] *B. Tan, X. Shen, and C. Zhai, "Mining Long-Term Search History to Improve Search Accuracy," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD),2006.*

[5] *K. Sugiyama, K. Hatano, and M. Yoshikawa, "Adaptive Web Search Based on User Profile Constructed without any Effort from Users," Proc. 13th Int'l Conf. World Wide Web (WWW),2004.* [9]

[6] *X. Shen, B. Tan, and C. Zhai, "Implicit User Modeling for Personalized Search," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM),2005.*

[7] *X. Shen, B. Tan, and C. Zhai, "Context-Sensitive Information Retrieval Using Implicit Feedback," Proc. 28th Ann. Int'l ACMSIGIR Conf. Research and Development Information Retrieval (SIGIR), 2005.*

[8] *F. Qiu and J. Cho, "Automatic Identification of User Interest for Personalized Search," Proc. 15th Int'l Conf. World Wide Web(WWW),pp. 727-736, 2006.*

J. Pitkow, H. Schu ¨tze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel, "Personalized Search," Comm.ACM, vol. 45, no. 9, pp. 50-55, 2002.