International Journal of Research

International Journal of Research (IJR)

e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015

Available at http://internationaljournalofresearch.org

Enhancing Social Network Privacy in Web Browsing

T.Manjula[1] & K.G.Anitha[2]

1 Associate Professor, PBR Viswodaya Institute of Technology and Science, Kavali.

2 PG Scholar, Dept of CS(CSE) PBR Viswodaya Institute of Technology and Science, Kavali.

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 run time generalization aim sat striking a balance between wopredictive metrics that evaluate the utility of personalization and the privacy risk of exposing the generalized profile. We present two greedy algorithms, namely Greedy D P and Greedy IL, for runtime gene realization. We also provide an on line 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 out performs Greedy DP in terms of efficiency.

Keywords: Privacy protection; personalized web search; utility; risk; profile

INTRODUCTION

The web search engine has long become the mostimportant 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 toperform consistently and considerably well ,it can only work on repeated queries from the same user, which is astrong 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,user documents 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,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.

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 userprofile. On the other hand, they need to hide the privacy
contents existing in the user profile to place the privacy risk under control. A fewprevious studies suggest that people are willing to compromise privacy if the personalization by supplying user profile to the searchengine
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 foronly 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 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 all the sensitive topics are detected using an absolute metric called surprisal based on theinformation theory, assuming that the interests withless user document support are more sensitive. However, this assumption can be doubted with a simple counterexample: 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 , average rank , 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.

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

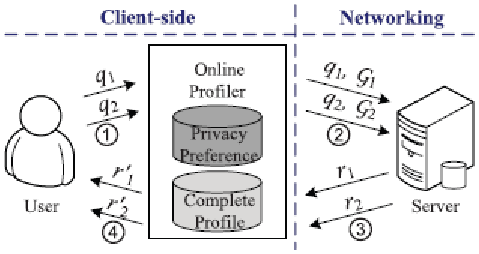


Fig. 1. System architecture of UPS.

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 qi 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 Gisatisfying 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 thepersonalization 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 websearch 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 problemof privacy-preserving personalized search as
- -RiskProfile Generalization, with itsNP-hardness proved.
- We develop two simple but effective generalizationalgorithms, GreedyDP and GreedyIL, to supportruntime profiling. While the former tries to maximizethe discriminating power (DP), the latterattempts to minimize the information loss (IL). Byexploiting a number of heuristics, GreedyIL outperformsGreedyDP significantly.
- We provide an inexpensive mechanism for the clientto 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.

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 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, 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 is a common measure of the effectiveness of an information retrieval system. It is based on a humangradedrelevance 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 [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 alsoproposes two predictive metrics, namely personalization utility and privacy risk, on a profile instance without requesting for user feedback.

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.

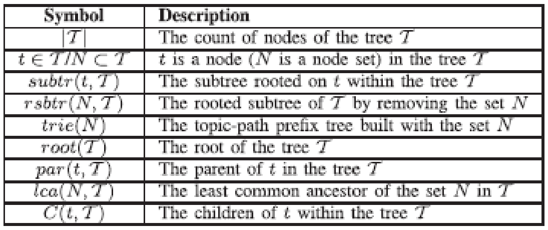
Therefore, the existing efforts focus on the second level. Both [21] and [22] provide online anonymity on user profiles by generating a group profile of k users. Using this approach, the linkage between the query and a single user is broken. In [23], the useless user profile (UUP) protocol is proposed 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.

They candecide 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 statisticaltechniques 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 subtreeof 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 studieshave raised an interesting question that concerns the privacy protection in PWS. The works in [1], [26] havefound that personalization may have different effects on different queries. Queries with smaller click-entropies, namely distinct queries, are expected to benefit more from personalization, while those with larger values (ambiguousones) are not. Moreover, the latter may even cause privacy disclosure. Therefore, the need for personalization becomes questionable for such queries. Teevan et al. [26] collect a set of features of the query to classify queries by their clickentropy.While these works are motivative in questioning whether to personalize or not to, they assume the availability of massive user query logs (on the server side) and user feedback. In our UPS framework, we differentiate distinct queries from ambiguous ones based on a client-side solution using the predictive query utility metric.

TABLE 1
Symbols and Descriptions



Symbol	Description
$\lvert\mathcal{T}\rvert$	The count of nodes of the tree $\mathcal{T}$
$t \in \mathcal{T}/N \subset \mathcal{T}$	t is a node (N is a node set) in the tree $\mathcal{T}$
$subtr(t, \mathcal{T})$	The subtree rooted on t within the tree $\mathcal{T}$
$rsbtr(N, \mathcal{T})$	The rooted subtree of $\mathcal{T}$ by removing the set N
$trie(N)$	The topic-path prefix tree built with the set N
$root(\mathcal{T})$	The root of the tree $\mathcal{T}$
$par(t, \mathcal{T})$	The parent of t in the tree $\mathcal{T}$
$lca(N, \mathcal{T})$	The least common ancestor of the set N in $\mathcal{T}$
$C(t, \mathcal{T})$	The children of t within the tree $\mathcal{T}$

PRELIMINARIES and PROBLEM DEFINITION

In this section, we first introduce the structure of userprofile in UPS. Then, we define the customized privacy requirements on a user profile. Finally, we present the attack model and formulate the problem of privacypreservingprofile generalization. For ease of presentation, Table 1 summarizes all the symbols used in this paper.

3.1 User Profile

Consistent with many previous works in personalized web services, each user profile in UPS adopts a hierarchical structure. 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 subtreesubtrðt; RÞ as the taxonomy accompanying t.

The repository is regarded as publicly available and canbe used by anyone as the background knowledge. Such
repositories do exist in the literature, for example, the ODP [1], [14], [3], [15], Wikipedia [16], [17], WordNet [22], and so
on. In addition, each topic t 2 R is associated with a repository support, denoted by supRðtÞ, which quantifies how often the respective topic is touched in human knowledge. If we consider each topic to be the result of a random walk from its parent topic in R, we have the following recursive equation:

$$sup_{\mathcal{R}}(t) = \sum_{t' \in C(t,\mathcal{R})} sup_{\mathcal{R}}(t').$$

Equation (1) can be used to calculate the repository support of all topics in R, relying on the following assumption that the support values of all leaf topics in R are available.
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. That is, supRðtÞ can be calculated as the count of leaves in subtrðt; RÞ.Based on the taxonomy repository, we define a probability model for the topic domain of the human knowledge.
In the model, the repository R can be viewed as a hierarchical partitioning of the universe

(represented by the root topic) and every topic t 2 R stands for a random event.

The conditional probability Prðtj sÞ (s is an ancestor of t) is defined as the proportion of repository support:

$$Pr(t \mid s) = \frac{sup_{\mathcal{R}}(t)}{sup_{\mathcal{R}}(s)}, \quad t \in subtr(s, \mathcal{R}). \qquad (2)$$

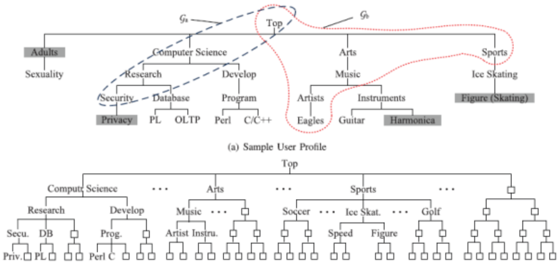
Thus, $Pr(t)$ can be further defined as

$$Pr(t) = Pr(t \mid root(\mathcal{R})), \qquad (3)$$

Definition 1 (USER PROFILE/H). A user profile H, as a
hierarchical representation of user interests, is a rootedsubtree of R. The notion rooted subtree is given in
Definition 2 (ROOTED SUBTREE). Given two trees S and T ,S is a rooted subtree of T if S can be generated from T by
removing a node set X _ T (together with subtrees) from T ,

$i.e., \mathcal{S} = rsbtr(X, \mathcal{T}).$

A diagram of a sample user profile is illustrated inFig. 2a, which is constructed based on the sample taxonomy
repository in Fig. 2b. We can observe that the owner of thisprofile is mainly interested in Computer Science and Music,because the major portion of this profile is made up of
fragments from taxonomies of these two topics in thesample repository. Some other taxonomies also serve incomprising the profile, for example, Sports and Adults.



(a) Sample User Profile
(b) Sample Taxonomy Repository

Fig. 2. Taxonomy-based user profile.

Although a user profile H inherits from R a subset oftopic nodes and their links, it does not duplicate the repository supports. Instead, each topic t 2 H is labeled with a user support, denoted by supHðtÞ, which describes the user's preference on the respective topic t. Similar to its repository counterpart, the user support can be recursively aggregated from those specified on the leaf topics:

supHðtÞ ¼
X
t02Cðt;HÞ
supHðt0Þ: ð4Þ

The user support is different from the repository support as the former describes the user's preference on t, while the latter indicates the importance of t in the entire human knowledge.

3.2 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.

Definition 3 (SENSITIVE NODES/S). Given a user profile H, the sensitive nodes are a set of user specified sensitive topics S _ H, whose subtrees are nonoverlapping, i.e., 8s1; s2 2 Sðs1 6¼ s2Þ; s2 62 subtrðs1; HÞ.

In the sample profile shown in Fig. 2a, the sensitive nodes S ¼ fAdults; Privacy;Harmonica; Figure ðSkatingÞg are shaded in gray color in H.

It must be noted that user's privacy concern differs from one sensitive topic to another. In the above example, the user may hesitate to share her personal interests (e.g., Harmonica, Figure Skating) only to avoid various advertisements.

Thus, the user might still tolerate the exposure ofsuch interests to trade for better personalization utility.

However, the user may never allow another interest in topic Adults to be disclosed. To address the difference in privacy concerns, we allow the user to specify a sensitivity for each nodes 2 S.

Definition 4 (SENSITIVITY). Given a sensitive-node s, its sensitivity, i.e., senðsÞ, is a positive value that quantifies the severity of the privacy leakage caused by disclosing s. As the sensitivity values explicitly indicate the user's privacy concerns, the most straightforward privacy preserving method is to remove subtrees rooted at all sensitive-nodes whose sensitivity values are greater than a threshold. Such method is referred to as forbidding.

However, forbidding is far from enough against a more sophisticated adversary. To clearly illustrate the limitation of forbidding, we first introduce the attack model which we aim at resisting.

Attack Model

Our work aims at providing protection against a typical model of privacy attack, namely eavesdropping. As shown in Fig. 3, to corrupt Alice's privacy, the eavesdropper Eve successfully intercepts the communication between Alice and the PWS-server via some measures, such as man-in-the middle attack, invading the server, and so on. Consequently, whenever Alice issues a query q, the entire copy of q together with a runtime profile G will be captured by Eve. Based on G, Eve will attempt to touch the sensitive nodes ofAlice by recovering the segments hidden from the originalH and computing a confidence for each recovered topic, relying on the background knowledge in the publicly available taxonomy repository R.

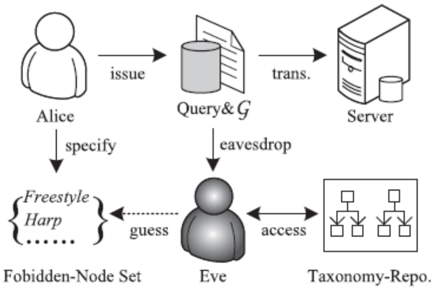


Fig. 3. Attack model of personalized web search.

In this section, we present the experimental results of UPS. We conduct four experiments on UPS. In the first

experiment, we study the detailed results of the metrics in each iteration of the proposed algorithms. Second, we look at the effectiveness of the proposed query-topic

mapping. Third, we study the scalability of the proposed algorithms in terms of response time. In the fourth experiment, we study the effectiveness of clarity prediction and the search quality of UPS.

Experimental Setup

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 thealgorithms 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 (March 1, 2006 to May 31,

2006). 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.

The profiles used in our experiment can be either synthetic or generated from real query logs:

Synthetic. We cluster all AOL queries by their DP into three groups using the 1-dimensional k-means algorithm. These three groups, namely Distinct Queries, Medium Queries, and Ambiguous Queries,

can be specified according to the following empirical rules obtained by splitting the boundaries between two neighboring clusters.

- *Distinct Queries* for $DP(q, \mathcal{R}) \in (0.82, 1]$.
- *Medium Queries* for $DP(q, \mathcal{R}) \in (0.44, 0.82)$.
- *Ambiguous Queries* for $DP(q, \mathcal{R}) \in (0, 0.44)$.

Each synthetic profile is built from the click log of three queries, with one from each group. The forbidden node set S is selected randomly from the topics associated with the clicked documents.

Real. The real user profiles are extracted from

50 distinct user click logs (with #clicks _ 2;000) from

AOL. For each user, the user profile is built with thedocuments dumped from all urls in his/her log.6

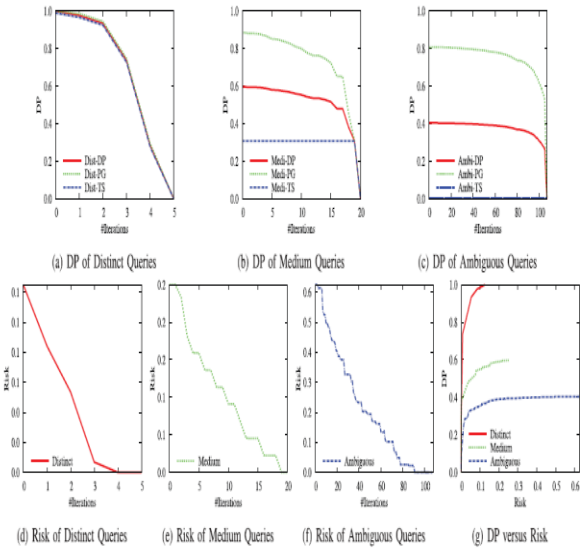
The sensitive nodes are randomly chosen from no more than five topics (with depth _ 3Þ.

Micro Results of Queries

In this experiment, we analyze and compare the effect of the generalization on queries with different discriminating power, and study the tradeoff between the utility and the

privacy risk in the GreedyDP/GreedyIL algorithm. To

clearly illustrate the difference between the three groups of queries, we use the synthetic profiles in this experiment. We perform iterative generalization on the profile using one of the original queries for creating the profile itself. The DP and risk are measured after each iteration. As the results of different profiles display similar trends, we only plot theresults of three representative queries (“Wikipedia” for distinct queries, “Freestyle” for medium queries, and Program” for ambiguous queries) in Fig. 5.



(a) DP of Distinct Queries (b) DP of Medium Queries (c) DP of Ambiguous Queries

(d) Risk of Distinct Queries (e) Risk of Medium Queries (f) Risk of Ambiguous Queries (g) DP versus Risk

Fig. 5. Results of Distinct/Medium/Ambiguous queries during each iteration in GreedyDP/GreedyIL. All results are obtained from the same profile.

As Figs. 5a, 5b, and 5c show the discriminating power ofall three sample queries displays a diminishing-returns property during generalization, especially the ambiguous one (i.e., “Program”). This indicates that the higher-level topics in the profile are more effective in improving the

search quality during the personalization, while the lowerlevelones are less. This property has also been reported in [12], [10]. In addition, we also plot the results of profilegranularity and topic similarity (TS) across iterations inthese figures. We observe that for all three samples, 1) PG

shows an exactly similar trend as that of DP, 2) TS remains unchanged until the last few iterations of generalization. In particular, the TS of the ambiguous one is

always 0. The reason of such results is that TS is fixed before the generalization reaches the least common ancestor of the related queries, which means PG shapes the overall DP more. Similarly, Figs. 5d, 5e, and 5f show the results of risk during the generalization. The value of the metric first

declines rapidly, but the decrease slows down as more specific profile information becomes hidden.

Fig. 5g illustrates the tradeoff pattern of DP versus risk of three sample queries. For all queries, we observe an apparent “knee” on their tradeoff curve. Before this turning point, small concessions on risk can bring great promotion on utility; while after that, any tiny increase of utility will lead to enormous increase in risk. Hence, the knee is a nearoptimalpoint for the tradeoff. We also find that the kneecan be reached within limited iterations for all cases (whenrisk is below 0.1).

Ex2: **Efficiency of Generalization Algorithms**

To study the efficiency of the proposed generalization algorithms, we perform GreedyDP and GreedyIL algorithms on real profiles. The queries are randomly selected

from their respective query log. We present the results in terms of average number of iterations and the response time of the generalization.

Fig. 6 shows the results of the experiment. For comparison, we also plot the theoretical number of iterations of the Optimal algorithm. It can be seen that both greedy

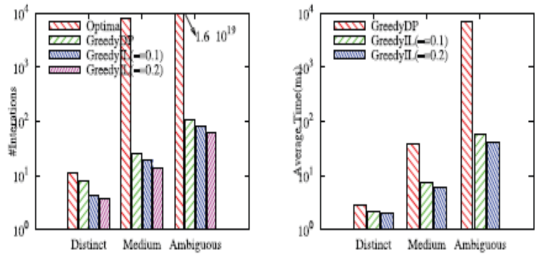
algorithm outperform Optimal. GreedyDP bounds the search space to the finite-length transitive closure of prune-leaf. GreedyIL further reduces this measure with

Heuristic 1. The greater the privacy threshold _, the fewer iterations the algorithm requires.

The advantage of GreedyIL over GreedyDP is moreobvious in terms of response time, as Fig. 6b shows. This is because GreedyDP requires much more recomputation of

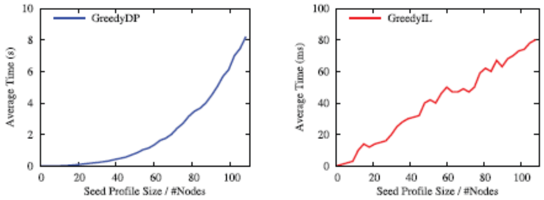
DP, which incurs lots of logarithmic operations. The problem worsens as the query becomes more ambiguous.

For instance, the average time to process GreedyDP for queries in the ambiguous group is more than 7 seconds. In contrast, GreedyIL incurs a much smaller real-time cost, and outperforms GreedyDP by two orders of magnitude.



(a) Results of Average #Iterations (b) Results of Average Response Time

Fig. 6. Efficiency of Optimal/GreedyDP/GreedyIL.



(a) Results of GreedyDP (b) Results of GreedyIL

Fig. 7. Scalability by varying profile size.

Ex3: Scalability of Generalization Algorithms

We study the scalability of the proposed algorithms byvarying 1) the seed profile size (i.e., number of nodes), and

2) the data set size (i.e., number of queries). For each possible seed profile size (ranging from 1 to 108), we randomly choose 100 queries from the AOL query log, and take their respective RðqÞ as their seed profiles. All leaf nodes in a same seed profile are given equal user preference. These queries are then processed using the GreedyDP and GreedyIL algorithms. For fair comparison,

we set the privacy threshold _ ¼ 0 for GreedyIL to make it always run the same number of iterations as GreedyDPdoes.

Fig. 7 shows the average response time of the two algorithms while varying the seed profile size. It can be seen that the cost of GreedyDP grows exponentially, and exceeds 8 seconds when the profile contains more than 100 nodes. However, GreedyIL displays near-linear scalability, and significantly outperforms GreedyDP.

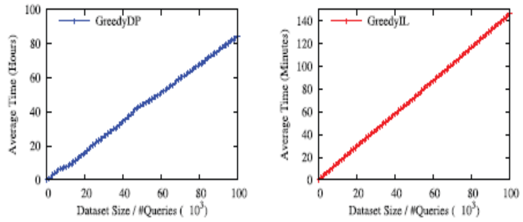
Fig. 8 illustrates the results of data sets containing different numbers of queries (from 1,000 to 100,000 queries). Apparently both algorithms have linear scalability by the data set size. For the largest data set containing 100,000 queries, it took GreedyDP 84 hours to complete all queries whileGreedyIL less than 150 minutes.

Ex4: Effective Analysis of Personalization

In this experiment, we evaluate the real search quality on commercial search engines using our UPS framework. The search results isreranked with the generalized profile output by GreedyIL over 50 target users. The final search quality is evaluated using the Average Precision of the click records of the users, which is defined as

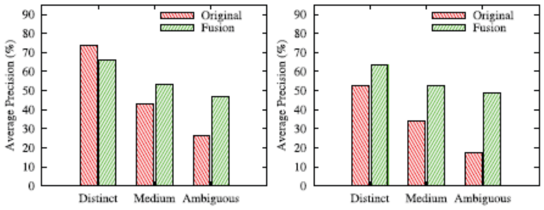
$$AP = \sum_{i=1}^{n} \frac{i}{l_i.rank} / n,$$

where li is the ith relevant link identified for a query, and n is the number of relevant links.



(a) Results of GreedyDP (b) Results of GreedyIL

Fig. 8. Scalability by varying data set size.



(a) Yahoo

(b) ODP

Fig. 9. Effectiveness of personalization on test queries.

For each test query, the framework computes the finalpersonalized rank as the Borda fusion [1] of the UPRankand the original rank, and then evaluate AP of the search results on both the fusion and the original rank. UPRank is achieved by sorting link items l in the descending order ofuscore, which is the weighted sum over related topics in profile G_, where the weight dnbðl; tÞ is the relevance quantified in (17). The uscore is given by

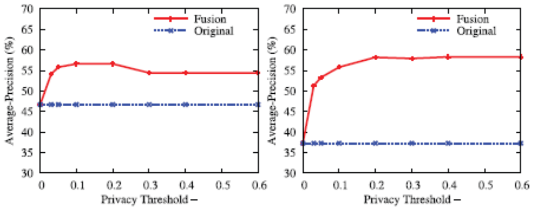
$$uscore(l) = \sum_{t \in T_{\mathcal{G}*}(q)} dnb(l,t).$$

Fig. 9 shows the average AP of the ranks before (Original) and after (Fusion) personalizing the test queries on Yahoo and ODP, respectively. The GreedyIL has a _ ¼ 0:1 and online decision mechanism disabled. From the results of both search engines, we can observe that improvements of the search quality for Medium Queries and Ambiguous Queries are much more significant than that of Distinct Queries. In particular, the personalization on Distinct Queries of Yahoo results reduces the average performance from 73.4 to 66.2 percent. This is because some irrelevant profile topics (noises) are added. The results demonstrate that profile-based personalization is more suitable for queries with small DPðq; RÞ.

Fig. 10 shows the results of search quality by varying the _ threshold. It is observed that the average precision of FusionRank increases rapidly when _ grows from 0.0 to 0.1. Then, further increasing _ (in effect exposing more specific topics) will only improve the search quality marginally.

Moreover, the AP of FusionRank based on Yahoo (Fig. 10a) has a significant drop when _ > ¼0:3.

A comparison between the personalization results of ODP and Yahoo reveal that, although the original ODPRank (AP ¼ 37:3%) is poorer than the original Yahoo-Rank (AP ¼ 46:7%), personalization on ODP will generate better ranking than that on Yahoo. The reason for this may be that the document-distribution of ODP over all the available topics is expectedly more consistent with its own taxonomy repository, which has been employed in our implementation.



(a) Yahoo

(b) ODP

Fig. 10. Effectiveness of personalization on varying δ.

CONCLUSION

This paper presented a client-side privacy protectionframework 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. The results also confirmed the effectiveness and efficiency of our solution.

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 (relaxing the second constraint of the adversary in Section 3.3) 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] B.Samhita,Y.Dilip Kumar,Dr. P.H.V Sesha Talpa Sai, "Investigation Of Heat Sink Capacity Of N-Heptane Fuel For Scramjet Application" Proc. www.ijastems.org,voi.1,Issue.2

[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.

[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 ACM SIGIR 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.

[9] 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.

[10] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-Enhancing Personalized Web Search," Proc. 16th Int'l Conf. World Wide Web (WWW), pp. 591-600, 2007.

[11] K. Hafner, Researchers Yearn to Use AOL Logs, but They Hesitate, New York Times, Aug. 2006.

[12] A. Krause and E. Horvitz, "A Utility-Theoretic Approach to Privacy in Online Services," J. Artificial Intelligence Research,
vol. 39, pp. 633-662, 2010.

[13] J.S. Breese, D. Heckerman, and C.M. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI), pp. 43-52, 1998.

[14] P.A. Chirita, W. Nejdl, R. Paiu, and C. Kohlschu¨ tter, "Using ODP Metadata to Personalize Search," Proc. 28th Ann. Int'l ACM

SIGIR Conf. Research and Development Information Retrieval (SIGIR), 2005.

[15] A. Pretschner and S. Gauch, "Ontology-Based Personalized Search and Browsing," Proc. IEEE 11th Int'l Conf. Tools with Artificial Intelligence (ICTAI '99), 1999.

[16] E. Gabrilovich and S. Markovich, "Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge," Proc. 21st Nat'l Conf. Artificial Intelligence
(AAAI), 2006.

[17] K. Ramanathan, J. Giraudi, and A. Gupta, "Creating Hierarchical User Profiles Using Wikipedia," HP Labs, 2008.

[18] K. Ja¨rvelin and J. Keka¨la¨inen, "IR Evaluation Methods for Retrieving Highly Relevant Documents," Proc. 23rd Ann. Int'l
ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), pp. 41-48, 2000.

[19] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison Wesley Longman, 1999.

[20] X. Shen, B. Tan, and C. Zhai, "Privacy Protection in Personalized Search," SIGIR Forum, vol. 41, no. 1, pp. 4-17, 2007.

[21] Y. Xu, K. Wang, G. Yang, and A.W.-C. Fu, "Online Anonymity for Personalized Web Services," Proc. 18th ACM Conf. Information and Knowledge Management (CIKM), pp. 1497-1500, 2009.

[22] Y. Zhu, L. Xiong, and C. Verdery, "Anonymizing User Profiles forPersonalized Web Search," Proc. 19th Int'l Conf. World Wide Web (WWW), pp. 1225-1226, 2010.

[23] J. Castellı´-Roca, A. Viejo, and J. Herrera-Joancomartı´, "Preserving User's Privacy in Web Search Engines," Computer Comm., vol. 32,no. 13/14, pp. 1541-1551, 2009.

[24] A. Viejo and J. Castell_a-Roca, "Using Social Networks to Distort Users' Profiles Generated by Web Search Engines," Computer Networks, vol. 54, no. 9, pp. 1343-1357, 2010.

[25] X. Xiao and Y. Tao, "Personalized Privacy Preservation," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD), 2006.