



Providing Privacy Protection in Personalized Web Search and Efficient Browsing

Manali Wadnerkar

Department of Computer Engineering,
BVCOE, University of Mumbai
Mumbai, Maharashtra, India

Abstract—Personalized web search (PWS) has illustrated its effectiveness by improving the quality of search services on the Internet. But, evidence shows that users' hesitation to disclose their private information during search has become a major barrier for PWS. Privacy protection in PWS applications can be adopted that model user preferences as hierarchical user profiles by studying a PWS framework called UPS that adaptively generalizes profiles by queries while keeping in mind user-specified privacy requirements. The method of query clustering that uses historical preferences of the user. This helps to achieve relevance criterion. Also, analysing the user's decision process on a given query with implicit feedback.

Keywords —personalized web search, privacy, query-clustering, user profile.

I. INTRODUCTION

The web search engine has become an important doorway for people for finding useful and necessary information. Nonetheless a user might come across failure when these search engines return unrelated results that do not meet their requirements. This happens due to enormous data, users' background and knowledge and ambiguity of texts. Personalized web search (PWS) is a search technique which aims at providing more efficient results, according to the users' needs. This requires user information to figure out the actual intention behind the requested query.

There are two solutions to PWS, click-log-based methods and profile-based-methods. The former is bias to clicked URLs or pages in the particular user's history and can work only on repeated queries. In contrast to this, the latter improves the search experience with user-interest models [1]. These user interest models are generated from users' profiles. PWS has illustrated more effectiveness in improving the quality of web data search. For this, implicit user data has to be collected which can be collected from query history [2], browsing history, bookmarks [1], and click-through data [3]. This raises privacy issues due to the lack of protection of user's private data. This may raise panic among the users and can also smother the publisher's enthusiasm for offering such services.

For protecting user privacy in profile-based PWS, developers have to consider two contradicting effects while performing the search process. They have to make an attempt to improve the search quality with the personalization utility and on the other hand they need to hide the privacy contents existing in the user profile for keeping the privacy risk under control [1]. People are willing to compromise their private data if this will help in an easy access to required information and an efficient search quality. A significant amount of gain can be obtained by personalizing users' information at the cost of a small information, a generalized profile. Hence, without compromising the search quality if the web user privacy can be protected. The previous works showing privacy preservation are not optimal. There are following concerns with the existing methods which can be explained as below:

1. The existing methods do not perform customization of privacy requirements. This makes some user privacy to be insufficiently protected and some over-protected. The sensitive topics are detected using the absolute metric called *surprisal* [1]. The topics, which are sensitive and the user wants to hide them may not be well protected. This increases the risk of losing a sensitive data.
2. The existing profile-based PWS are unable to support runtime profiling. A user, when searches the web engine, his profile is generalized only once. This strategy has certain drawbacks since it uses one profile for all the queries. A better approach can be to make an online decision for whether to personalize the query and at runtime what to expose in a user profile.
3. Iterative user interactions are needed when creating personalized search outcome. Predictive metrics unlike, *average rank*, are required to measure the search quality.

A key factor for today's popular search engines is that they provide a user-friendly interface. The topics which are displayed on the web page related to a particular query are in the form of list of keywords entered by the user in the search bar, ranked according to their relevance with the original query. Ranking has become a central research problem for

informational retrieval and Web data search, as it directly influences the relevance of the search results, the quality of a search system and users' search experience. Given a query, the deployed ranking function measures the relevance of each document to the query, sorts all the relevant documents and presents a list of top-ranked ones to the user. Despite of the simple interaction which proved to be successful, a list of keywords is not a good descriptor of the required information by the users. Users can not always formulate an efficient query to these search engines. One reason for this is the ambiguous data which is entered by the user. Often, users try different queries till get satisfied with the appropriate results. If users are familiar with the specific terminologies required, effective formulation can be achieved. But this may not be the case always. Users may have a little knowledge about what they are searching or even worse they do not what they are searching at all. An example explained in [2], a tourist is searching for summer rentals ad in Chile may not know that most of such ads appearing on the web are for apartments in *Vina del Mar* which is a popular beach in Chile. But local users are well aware of such facts. Hence, the idea is to use these expert queries for helping the non-expert users. So, to overcome this problem some search engines help the users to specify alternative queries related to the original query in their search process.

For example, electronic commerce (e-commerce) which is the use of computers and telecommunication technologies to share business information, maintain business relationships, and conduct business transactions. Today, e-commerce depends mostly on the Internet as the basic platform. With the increase in growth of Websites and increase in e-commerce systems and trade, web log data maintenance has become very important for many of services. Hundreds and Thousands of people may visit a particular website in a given period of time and then these visits are stored as web log data. Therefore, web log data provides important information about users' behaviors.

Further, the queries which are similar are put together in a cluster. So that whenever a user fires a query, the search will be done within clusters. This increases the efficiency of the search. Clustering analysis is widely used for establishing object profiles based on object's variables these objects can be considered to be customers, web users or any documents. Clustering analyzes data objects without consulting a certain known class label unlike classification that analyzes class-labeled data objects. These objects are clustered on the basis of minimizing inter-class similarity and maximizing the intra-class similarity which means that these clusters are created for the objects within a cluster. These objects have much more similarity when compared to the ones within the same cluster and are very much dissimilar to the ones in other clusters. Like query clustering, document clustering is also considered when it comes to better and faster search. It is the organization of a set of text documents on the basis of similarity into clusters. Obviously, these text documents which are contained in a cluster are similar than the ones which are present in different cluster.

Different algorithms can be considered while performing clusters. These algorithms are able to discover clusters which are highly coherent. These algorithms do not rely at all on the contents of the pages but uses repeated information across multiple transactions instead, for guiding the clustering. The clustering described follows two observations. One is the fact that the users may phrase their queries in a different manner though their information need relies with the same information. The other one is that the users may visit two altogether different websites. Web search services such as AltaVista, were introduced to help people find information on the web. Most of these systems return a ranked list of web pages in response to a user's search request. Web pages on different topics or different aspects of the same topic are mixed together in the returned list. The manual nature of the directory compiling process makes it impossible to have as broad coverage as the search engines, or to apply the same structure to intranet or local files without additional manual effort. To combine the advantage of structured topic information in directories and broad coverage in search engines, we built a system that takes the web pages returned by a search engine and classifies them into a known hierarchical structure. The search interface accepted query keywords, passed them to a search engine selected by the user, and parsed the returned pages. Clicking on the title hyperlink brought up the full content of the web page in another browser window, so that the category structure and the full-text of pages were simultaneously visible.

II. RELATED WORKS

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

Previous works shows only the search quality improvement. The basic idea is to search by referencing user profile which reveals individual's goal behind the query [1]. The representation of profiles are available to facilitate various personalization strategies. Earlier techniques used terms lists or vectors to represent profiles. But most recently it has been seen that the profiles are structured in the hierarchical form due to their better scalability and higher access rate like, Wikipedia. In the proposed framework User customizable Privacy preserving Search (UPS) (Fig. 1), the importance to the representation is not much since it adopts any hierarchical structure potentially. Privacy protection problems can be studied by classifying them into two classes. One class includes those who treat privacy as their identification and the other includes those consider sensitivity of data, user profiles, getting exposed to the server.

It can be tried to solve this problems on different levels like pseudo identity, no identity, and group identity. This will keep the user anonymous and their identification will remain unexposed. Limitations that can be considered are a finite set of attributes for building a user profile and introduces high cost due to communication required. These can be avoided by introducing privacy protection measures while building the user profile. A person can specify the degree of privacy protection for the values which he/she considers to be sensitive and does not want to get public. The works described in [1] show that personalization may have various effects on different queries. Distinct queries are expected to benefit more than the ambiguous ones. Hence, these queries should to relevant to the information needed by the user. Previous methods suggest to cluster similar queries to recommend URLs to frequently asked queries. Also, this can be done by

calculating the distance between any two clicked documents in some pre-defined hierarchy [2]. These notions are difficult to deal with in practice, because distance matrices between queries generated by them from real query logs are very sparse. Many queries with semantic connections appear as orthogonal objects in such matrices. So, for dealing with such problems, Ad-hoc clustering algorithms are needed. Another notion suggests an approach to query expansion. The idea here is to reformulate the query such that it gets closer to the term-weight vector space of the documents the user is looking for.

There are many clustering algorithms, which can be classified into several categories. A partitioning method classifies objects into several one-level clusters, wherein each object belongs to exactly one cluster, and each cluster has at least one object. A hierarchical method creates hierarchical decomposition of objects. Based on how the hierarchy is formed, hierarchical methods can be classified into agglomerative approaches and divisive approaches. A density-based method is used to discover clusters with arbitrary shape, based on the number of objects in neighborhood density. This method typically regards clusters as dense regions of objects in the data space that are separated by regions of low density.

Researchers have been investigating the more general problems of document clustering that would be a subset of the internet. One very popular technique is *k-means*, whose running time is linear in the terms of number of documents but it very effective. Also, hierarchical agglomerative clustering (HAC) technique for quadratic values of documents, which iteratively find two closest documents and merge them. Both these methods are prone to undesirable behavior when faced with outsiders.

In the proposed system, the distance between two documents can be evaluated without any evaluation of the documents. This is contrast to the traditional clustering algorithms that typically use a distance between the documents as some function of the fraction of the tokens which they have in common. When the web pages are clustered by contents, they may require storing and manipulating large amount of data. By estimations shown in [5], in 2000 there were at least a billion of pages over the internet which are accessible by search engines. Hence, content if ignored can be a valuable property. Content-ignorance is also applicable on text-free pages, pages having dynamic content or pages with restricted-access.

The most important advantage is that it can be implemented more efficiently than any other agglomerative techniques. Another technique was introduced which located groups of nearly equivalent documents in a vastly large database like internet. These techniques involved calculating “finger-print” of a web page based on contiguous subsequences of token in the web pages.

Earlier in search systems sessions were easy to determine. The historical session was a set of queries to satisfy a single information need, a short time period of contiguous time spent examining and querying results and a series of successive queries. The term session is split between different meanings. These may include identifiers such as IP address and cookies shared by multi-users. Several attempts were made to define time-out. Many used the sole idea of a “time-out” cutoff. A timeout is the time between two successive activities, and it is used as a session boundary when it exceeds a certain threshold [5]. But studies show that choice of cutoff does not matter. Sessions were better identified by a single word common between two queries. Few researchers have worked on automatically detecting the session boundaries. They relied on the sole feature of words which were common in queries, and rewrite classes like specification and generalization. They used contiguous values for predicting rewrite classes defined in the terms of insertion and deletion.

For searching the information and observing the URLs clicked by the users, eye-tracking is a concept which can be explored. The previous studies have used this concept loosely and they actually use users’ patterns of navigation across a general web page content and not the results displayed by the search engine. These studies are general in nature which depict the eye movement and navigation on the web page, and also assessing how the color of the link influences the decision of clicking links.

Another measure which has been used is the pupil dilation for inferring the relevance of the abstracts. But this research was a bit weak and showed no other measures.

III. SYSTEM ARCHITECTURE

A. Online profile

The proposed idea also suggests that the queries issued are recommended that are related to the input query and also search for different issues. This redirects the search process to related information of interest to the users searching previously and also keeping track of the related queries issued by other users. The key component for privacy protection is an online profiler implemented as a search proxy that runs on client side. This proxy maintains both the complete user profile in a hierarchical structure with semantics, and the user-specified privacy requirements i.e. sensitive nodes. It works in two phases, namely the offline phase and the online phase. In the offline phase, hierarchical profile is constructed and then customized with the user-specified privacy requirements [1]. The online phase can be conducted as follows:

1. When query is generated the proxy generates a runtime user profile. This process is guided by considering two conflicting metrics, personalization utility and privacy risk.
2. Then, the query and the generalized profile are sent together to the server.
3. These results are then personalized with the profile and delivered back to the query proxy.
4. Finally, the proxy sends back the results to the client or re-ranks them with the complete user profile.

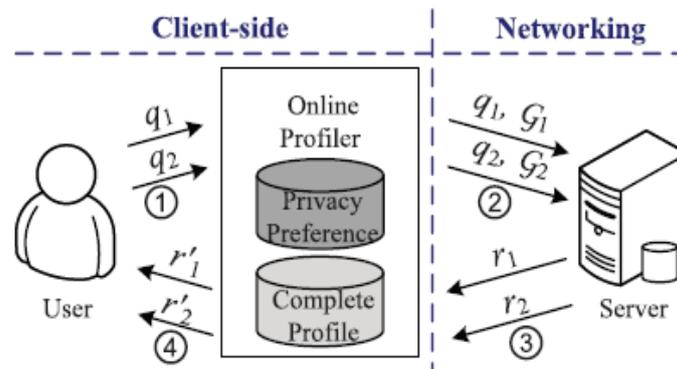


Fig. 1. System structure of UPS.

UPS differs from the conventional PWS since it provides runtime profiling which optimizes personalization utility, which performs customization on the sensitive data defined by the users, and does not require iterative user interaction.

Again, for efficient browsing, it is required to find the ranks of the related queries and cluster them. Queries along with the text of their clicked URLs extracted from the web log are clustered. This is done on the basis of two notions:

1. Similarity of the query. The similarity of the query to the input query.
2. Support of the query. This is a measure of how relevant is the query in the cluster. It is measured with the support of the query as the fraction of the documents returned by the query that captured the attention of users (clicked documents). It is estimated from the query log as well.

The quality of service can be improved when the location of the users are closer [4]. So, if the users share more data with each other the services provided by the web will be accurate. The studies show that the user is biased when it comes to searching information on the web. It can be trusts-biased or quality-biased [3]. This shows that clicks should be interpreted relative to the order of abstracts and presentation. Some attempts are made to use implicit feedback [4]. The reading time is indicative of interest while reading new stories. The reading time as well as number of times the user scrolls page can predict the relevance in browsing web. But it is generally considered that reading time varies between subjects and tasks, which makes it difficult to interpret. This difficulty can be resolved by the concept of eye-tracking. A general user approaches the results from top to bottom. It appears that users scan the viewable results before heading to scrolling. It gives evidences about users' decision making and indicates that users' clicking decisions are influenced by relevant results.

B. Session time-out

An experiment can be conducted where the users are observed with their clicked URL and session lengths and then can be re-enacted. For further help, clicks can be observed and assessment of the user's objectives can be done to label each session. Each query and clicked URL are assigned with ID number. A strength of this approach is that data is recorded without having an intervention and additionally we can observe large amount of users. There is a chance that the observer is biased to the user's goals but preliminary results show that the results achieved are reliable. The utility of adopting a hierarchical model for the grouping of user queries will allow us to more easily model what type of task the user may be doing when querying.

C. Attack Model

The user profile should be protected from adversaries which try to hamper the privacy and sensitive nodes defined by the user by a typical attack, namely eavesdropping. As shown in the Fig. 2 the eavesdropper intercepts successfully the communication happening between the server and the user by a measure, such as man-in-the-middle attack, invading the server. Accordingly, whenever the user issues any query q , the entire copy of q along with the runtime profile of the user will be seized the attacker.

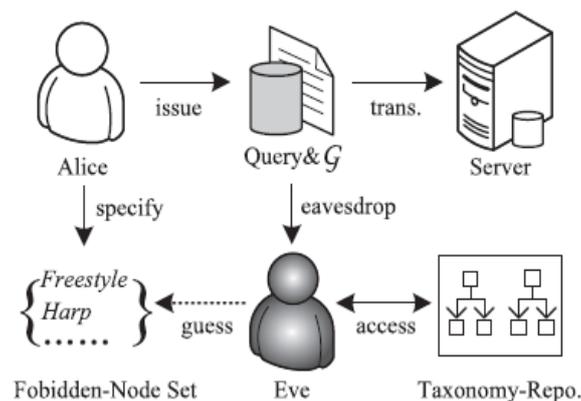


Fig. 2. Attack Model.

The attacker will then try to recover the hidden segments defined as private by the user. Now, the adversary is considered to satisfy the following assumptions:

Knowledge Bound. The background knowledge of the adversary is limited to the entire information available on the web. Both the original user profile and the privacy are defined within this information.

Session Bound. Previously captured information is not available for tracing the same victim. The eavesdropping will be started and ended within a single session.

These assumptions are strong but are reasonable in practice. This is considered since majority of attacks across the web happen by some automatic programs that sends advertisements (spam) to a wide range of users. An approach can be made to keep this privacy risk under control.

D. Generalizing user profile

This technique can be considered during the offline phase processing without involving any of the user's queries. But however it is impractical to perform this in offline phase because:

1. This output from the offline phase may contain many topics that are completely irrelevant to the particular query. This can be solve if profile is generalized in the online phase.
2. It avoids unnecessary privacy disclosure to the adversaries and also avoids noisy topics which are irrelevant to the query.
3. It is very important to monitor the personalization factor during generalizing. But overgeneralization may cause ambiguity.

There are four phases in [1] which are used in generalization of the user profile. They can be explained in the following manner:

1. *Offline profile construction:* This is the first step of the offline processing wherein the original user profile is built in a topic hierarchy which reveals the interest of the user.
2. *Offline privacy requirement customization:* This phase requests the user to specify sensitive nodes which the user considers to remain hidden from the world. When any query q is issued, this customized user profile goes through the online phases.
3. *Online query topic mapping:* There are two purposes for a query q , namely to compute a profile, *seed*, so that all the topics will be relevant to the query q . and to obtain the preference values i.e. values which are preferred to be present in the relevant topics of the query.
4. *Online profile generalization:* This process generalizes the *seed* profile which relies on the privacy requirements of the user.

E. User behavior

User behavior is to be determined by observing the clicks made by the user. But before analyzing the strategies for user feedback. It is necessary to analyze how users scan the results page. Users do not click on links at random, but make an informed choice. While click-through data is typically noisy and clicks are not "perfect" relevance judgments, the clicks are likely to convey some information. Clicks can be interpreted with respect to the parts of the results which users observe and evaluate. The users often click on the first link than the second one. But they view the corresponding abstract with almost same frequency.

It is interesting to observe that around 6/7 line, both the view behavior and the number of clicks change [3]. The links below receive less attention substantially than those which are present earlier. As the user starts scrolling the page attention starts reducing. As there are ten results on a single web page, there is a sharp drop of attention after 10th link.

Again there is an observation that the users generally scan from top to bottom. It appears that the users first scan the viewable results thoroughly and then starts scrolling the web page. The lower the clicks are made, the more abstracts are viewed above the click than the ones which are displayed below. Users' behavior depends on the order of the links displayed and the clicking decisions made by the user are influenced by the relevance of the queries.

1) *Trust bias:* The URLs ordered or ranked first receives more clicks than the second one. This happens because the search engines return rankings where the first link shows more relevance than the second link. The users make decisions based on their relevance assessment regarding the abstract. The other explanation can be made that the users prefer the first link than the second one due to some level of trust in the search engine. Here, user gets influenced by the presentation. The users have substantial trust in the search engine's ability to calculate the relevance of a page, which influences their clicking behavior.

2) *Quality bias:* The quality of the ranking influences the decision of the users. If a user has made a particular query before and went through all the documents, URLs or abstracts in his/her search and is making a similar query again, he/she knows what kind of URLs are trustworthy or show relevant results. The users will then search accordingly. The users' behavior depends on the quality of the URLs due to trust issues. A user always wants to search in a safe environment and hence trusts on the links which were visited previously for any similar search.

F. Online decision

When the profile is sent to the server the decision is made online. Here the user can decide whether the profile should be personalized or not. This depends on the number of distinct or similar queries. The similar queries are clustered together so that whenever the search is made the similar queries can be found in one place. This improves and enhances the search quality of the search engine.

The profile-based personalization contributes a little and even reduces the quality of search when there is a large amount of distinct queries. This may expose the profile to the server and will risk the privacy of the user. There is a solution to this problem. The decision to personalize the users' profile or not can be made in the online phase. The idea behind this phase is very simple, if a query issued is a distinct query during generalization the complete runtime profiling will then be aborted. Then the query will be sent to the server without any user profile. This enhances the stability of the quality of the search and also avoids the unnecessary exposure of the users' profile.

G. Using structure to support search

A number of web search services use category information to organize the search results. Many search engines show the category label associated with each retrieved page. Results are still shown as a ranked list with grouping occurring only at the lowest level of the hierarchy. There is, for example, no way to know that 70% of the matches fell into a single top-level category. In addition, these systems require pre-tagged content. Before any new content can be used, it must be categorized by hand. Custom Folders in which the retrieved documents are organized hierarchically. These folders are organized according to several dimensions such as, source (sites, domains), type (personal page, product review), language, and subject. Individual categories can be explored one at a time. But, again no global information is provided about the distribution of search results across categories.

IV. CONCLUSION

A client side privacy protection framework called UPS is proposed for personalized web search. This framework can be adopted by any PWS which seizes user profiles in hierarchy. This framework allows customers to customize their own privacy. It also performs online generalization on user profiles and that too without compromising search quality.

Users clicking decisions are influenced by the relevance of the results but they are biased either due to trust factor or quality factor. Personalizing websites can attract new customers and retain existing customers. Web personalization technologies also benefit e-commerce applications and allow users to see and receive information based on the knowledge acquired from the users' previous actions.

Within a search engine clustering organizes all web pages and objects into groups and also cluster just those pages or objects suggested to a user in response to the user's query. This improves browsing and search results. The utility of adopting a hierarchical model for the grouping of user queries will allow us to more easily model what type of task the user may be doing when querying. Further, adversaries can be resisted with more background knowledge by having more capability to capture a series of queries from the victim user.

REFERENCES

- [1] Lidan Shou, He Bai, Ke Chen, and Gang Chen, "Supporting Privacy Protection in Personalized Web Search," Proc. IEEE Transactions on knowledge and data engineering vol:26 no:2 year 2014.
- [2] Ricardo Baeza-Yates¹, Carlos Hurtado¹, and Marcelo Mendoza, "Query Recommendation using Query Logs in Search Engines," Proc. Millennium Nucleus, Center for Web Research (P01-029-F), Mideplan, Chile.
- [3] Thorsten Joachims, Laura Granka, Bing Pan, "Accurately Interpreting Clickthrough Data as Implicit Feedback," Proc. SIGIR'05, August 15–19, 2005, Salvador, Brazil..
- [4] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, Hang Li, "Context-Aware Query Suggestion by Mining Click-Through and Session Data," KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.
- [5] S. Taherizadeh N. Moghadam, "Integrating Web Content Mining into Web Usage Mining for Finding Patterns and Predicting Users' Behaviors," International journal of information science and management year 2012.
- [6] Rosie Jones, Kristina Lisa Klinkner, "Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs," CIKM'08, October 26–30, 2008, Napa Valley, California, USA.