IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Privacy Assurance In Publishing Search Logs Using Data Mining Techniques

R.Hari Krishna[1] and D.Anitha [2]

[1, 2] **Computer Science and Engineering, SRM University, Chennai, India**

Abstract
The Search engine companies collect the "database of intentions," the histories of their users' search queries. These search logs are a gold mine for researchers. Search engine companies, however, are wary of publishing search logs in order not to disclose sensitive information. The methods vary in the guarantee of disclosure limitations they provide and in the amount of useful information they retain. Existing proposals to achieve k-anonymity in search logs are insufficient in the light of attackers, who can actively influence the search logs. Then turn to Differential privacy, a much stronger privacy guarantee, it is impossible to achieve good utility. Evaluation includes applications that use search logs for improving both search experience and search performance, and our results show that ZEALOUS' output is sufficient for these applications while achieving strong formal privacy guarantees using two real applications from the information retrieval community: Index caching and query substitution, as a representative application for search quality. For both applications, the sufficient statistics are histograms of keywords, queries, or query pairs. Here, analyzing algorithms for publishing infrequent keywords, queries, and clicks of a search log, this provides a strong privacy mechanism.

Keywords: *k-anonymity, ZEALOUS, Index Caching, Probabilistic Differential privacy, Query Substitution.*

1. Introduction

Search engines play a crucial role in the navigation through the vastness of the Ib. Today's search engines do not just collect and index WebPages, they also collect and mine information about their users. They store the queries, clicks, IP-addresses and other information about the interactions with users, in what is called a search log. Search logs contain valuable information that search engines use to tailor their services better to their users' needs. They enable the discovery of trends, patterns and anomalies in the search behavior of users and they can be used in the development and testing of new algorithms to improve search performance and quality. Scientists all around the world would like to tap this gold mine for their own research. Search engine companies, however, do not release them because they contain sensitive information about their users, for example searches for diseases, lifestyle choices, personal tastes and political affiliations.

The only release of a search log happened in 2006 by AOL and it into the annals of tech history as one of the greatest debacles in the search industry. AOL published three months of search logs of 650,000 users. The only measure to protect user privacy was the replacement of user-ids with random numbers—utterly insufficient protection as the New York Times showed by identifying a user from Lilburn – Georgia, whose search queries not only contained identifying information but also sensitive information about her friends ailments.

The AOL search log release shows that simply replacing user-ids with random numbers does not prevent information disclosure. Other ad hoc methods have been studied and found to be similarly insufficient, such as the removal of names, age, zip codes and other identifiers and the replacement of keywords in search queries by random numbers.

Comparing the formal methods of disclosure, when publishing frequent keywords, queries, and clicks of a search log, the methods vary in the guarantee of disclosure limitations they provide and in the amount of useful information they retain. I first describe two negative results. I show that existing proposals to achieve *k-anonymity* in search logs are insufficient in the light of attackers who can actively influence the search log. I then turn to *Differential privacy*, a much stronger privacy guarantee. However it shows it is impossible to achieve good utility with *Differential privacy*.

The Algorithm ZEALOUS was developed independently by Korolova et al with the goal to achieve relaxations of

Differential privacy. Showed how to set the parameters of ZEALOUS to guarantee ð;Þ-*indistinguishability* and here offer a new analysis that shows how to set the parameters of ZEALOUS to guarantee ð;Þ-*probabilistic Differential privacy*, a much stronger privacy guarantee as the analytical comparison shows.

With an extensive experimental evaluation, where comparing the utility of various algorithms that guarantee anonymity or privacy in search log publishing. Evaluation includes applications that use search logs for improving both search experience and search performance and results show that ZEALOUS output is sufficient for these applications while achieving strong formal privacy guarantees.

By believing that the results of this research enable search engine companies to make their search log available to researchers without disclosing their user's sensitive information. Search engine companies can apply our algorithm to generate statistics that are ð; Þ-*probabilistic differentially privacy* while retaining good utility for the two applications tested. Beyond publishing search logs, I believe that the findings are of interest when publishing infrequent item sets, as ZEALOUS protects privacy against much stronger attackers

2. Related Work

2.1 Search Logs

Search engines such as Bing, Google, or Yahoo log interactions with their users. When a user submits a query and clicks on one or more results, a new entry is added to the search log. Without loss of generality, I assume that a search log has the following schema:

USER-ID; QUERY; TIME; CLICKS;

Where a USER-ID identifies a user, a QUERY is a set of keywords, and CLICKS is a list of url's that the user clicked on. The user-id can be determined in various ways; for example, through cookies, IP addresses, or user accounts. A user history or search history consists of all search entries from a single user. Such a history is usually partitioned into sessions containing similar queries; how this partitioning is done is orthogonal to the techniques.

A query pair consists of two subsequent queries from the same user within the same session. I say that a user history contains a keyword k if there exists a search log entry such that k is a keyword in the query of the search log. A keyword histogram of a search log S records for each keyword k the number of users ck whose search history in S contains k. A keyword histogram is thus a set of pairs ðk; ckÞ. I define the query histogram, the query pair histogram, and the click histogram analogously. I classify a keyword, query, consecutive query, clicks in a histogram to be frequent if its count exceeds some predefined threshold _; when I do not want to specify whether I count keywords, queries, etc., I also refer to these objects as items.

With this terminology, I can define our goal as publishing frequent items (utility) without disclosing sensitive information about the users (privacy). I will make both the notion of utility and simple type of disclosure is the identification of a particular user's search history (or parts of the history) in the published search log. The concept of *k-anonymity* has been introduced to avoid such identifications.

Definition 1 (k-anonymity): A search log is k-anonymous if the search history of every individual is indistinguishable from the history of at least k _ 1 other individuals in the published search logs. There are several proposals in the literature to achieve different variants of *k-anonymity* for search logs. Adar proposes to partition the search log into sessions and then to discard queries that are associated with fewer than k different user-ids. In each session, the user-id is then replaced by a random number. I call the output of Adar's Algorithm a k-query anonymous search log. Motwani and Nabar add or delete keywords from sessions until each session contains the same keywords as at least k _ 1 other session in the search log, following by a replacement of the user-id by a random number. I call the output of this algorithm a k-session anonymous search log. He and Naughton generalize keywords by taking their prefix until each keyword is part of at least k search histories and publish a histogram of the partially generalized keywords. I call the output a k-keyword anonymous search log. Efficient ways to anonymize a search log are also discussed by Yuan et al. Stronger disclosure limitations try to limit what an attacker can learn about a user. *Differential privacy* guarantees that an attacker learns roughly the same information about a user whether or not the search history of that user was included in the search log. *Differential privacy* has previously been applied to contingency tables, learning problems, synthetic data generation and more.

Definition 2 (*Differential privacy*): An algorithm A is differentially private if for all search logs S and S0 differing in the search history of a single user and for all output search logs. This definition ensures that the output of the algorithm is insensitive to changing/omitting the complete search history of a single user. I will refer to search logs that only differ in the search history of a single user as neighbouring search logs. Similar to the variants of *k-anonymity*, I could also define variants of *Differential privacy* by looking at neighbouring search logs that differ only in the content of one session, one query or one keyword. However, I chose to focus on the strongest definition in which an attacker learns roughly the same about a user even if that user's whole search history was omitted. *Differential privacy* is a very strong guarantee and in some cases it can be too strong to be practically achievable.

Definition 3 (Probabilistic Differential privacy): An algorithm guarantees that A achieves - *Differential privacy* with high probability. The set-2 contains all outputs that are considered privacy breaches according to *Differential privacy*; the probability of such an output is bounded.

Definition 4 (c-accuracy): An algorithm A is c-accurate if for any input search log S and any very frequent item d in S, the probability that A outputs d is at least in Experimental Utility Measures Traditionally, the utility of a privacy-preserving algorithm has been evaluated by comparing some statistics of the input with the output to see "how much information is lost." The choice of suitable statistics is a difficult problem as these statistics need to mirror the sufficient statistics of applications that will use the sanitized search log, and for some applications the sufficient statistics are hard to characterize. To avoid this drawback, Brickell and Shmatikov measure the utility with respect to data mining tasks and they take the actual classification error of an induced classifier as their utility metric.

Definition 5 (ðA; SÞ-inaccuracy): Given an algorithm A and an input search log S. The expectation is taken over the randomness of the algorithm. As an example, consider the simple algorithm that always outputs the empty set; I call this algorithm the baseline algorithm

2.2 Privacy Mechanism

Ignoring computational constraints, it is possible to privately release synthetic databases that are useful for large classes of queries – much larger in size than the database itself. Despite this, I give a privacy-preserving polynomial time algorithm that releases information useful for all half space queries, given a slight relaxation of the utility guarantee. This algorithm does not release synthetic data, but instead another data structure capable of representing an answer for each query. Here given an efficient algorithm for releasing synthetic data for the class of interval queries and axis-aligned rectangles of constant dimension. Finally, inspired by learning theory, I introduce a new notion of data privacy, which I call distributional privacy and show that it is strictly stronger than the prevailing privacy notion, *Differential privacy*[1].

Set-valued data, in which a set of values are associated with an individual, is common in databases ranging from market basket data, to medical databases of patients symptoms and behaviors, to query engine search logs. Anonymizing this data is important if there is to reconcile the conflicting demands arising from the desire to release the data for study and the desire to protect the privacy of individuals represented in the data. Unfortunately, the bulk of existing anonymization techniques, which Ire developed for scenarios in which each individual is associated with only one sensitive value, are not well-suited for set-valued data. Proposing a top-down, partition-based approach to anonymize set-valued data, that scales linearly with the input size and scores well on an information-loss data quality metric. Further noting that our technique can be applied to anonymize the infamous AOL query logs and discuss the merits and challenges in Anonymizing query logs using this approach [2].

Re-identification is a major privacy threat to public datasets containing individual records. Many privacy protection algorithms rely on generalization and suppression of quasi- identifier attributes such as ZIP code and birthdates. In this paper, generalization and suppression of quasi-identifiers over any benefits over trivial sanitization which simply separates quasi-identifiers from sensitive attributes. Previous work showed that k-anonymous databases can be useful for data mining, but k anonymization does not guarantee any privacy. Here results demonstrate that even modest privacy gains require almost complete destruction of the data-mining utility. In most cases, trivial sanitization provides equivalent utility and better privacy than k anonymity, diversity and similar methods based on generalization and suppression[3].

Introduce a new, generic framework for private data analysis. The goal of private data analysis is to release aggregate information about a data set while protecting the privacy of the individuals whose information the data set contains. This framework allows one to release functions f of the data with instance-based additive noise. That is, the noise magnitude is determined not only by the function required to release, but also by the database itself. One of the challenges is to ensure that the noise magnitude does not leak information about the database. This framework raises many interesting algorithmic questions. Namely, to apply the framework one must compute or approximate the smooth sensitivity of f on x. To show how to do this efficiently for several different functions, including the median and the cost of the minimum spanning tree and also give a generic procedure based on sampling that allows one to release f(x) accurately on many databases x[4].

3. Figure and Methodology

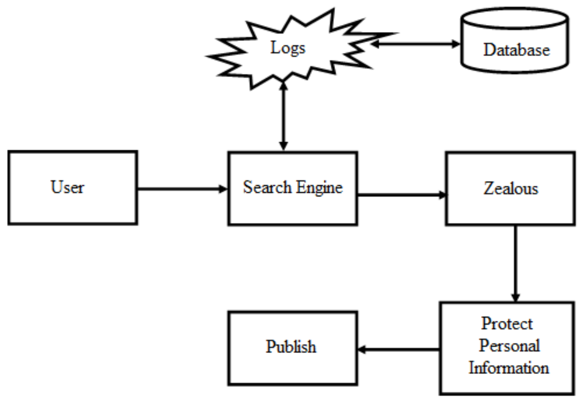

Fig. 1 Proposed Search Engine.

3.1 Search Engine Formation

Creating a search engine which scales even to today's web presents many challenges. Fast crawling technology is needed to gather the web documents and keep them up to date. Storage space must be used efficiently to store indices and optionally, the documents themselves.

The indexing system must process hundreds of gigabytes of data efficiently. Queries must be handled quickly, at a rate of hundreds to thousands per second. Search engines play a crucial role in the navigation through the vastness of the Ib. Today's search engines do not just collect and index WebPages, they also collect and mine information about their users. They store the queries, clicks, IP-addresses and other information about the interactions with users in what is called a search log.

3.2 Log Generation

When a user submits a query and clicks on one or more results, a new entry is added to the search log. It is also called search history. This will helpful to find the users behavior without loss of generality.

Traditional search log information are user-id, query, time and user clicks, where a USER-ID identifies a user, a QUERY is a set of keywords and the CLICKS is a list of URL'S that the user clicked on. The user-id can be determined in various ways for example, through cookies, IP addresses, or user accounts. The search logs are updated every time when user submits and browse a page.

3.3 Query Clustering

Query clustering is a process used to discover frequently asked questions or most popular topics on a search engine. This process is crucial for search engines based on question-answering. A query may be a well-formed natural language question, or one or more keywords or phrases. Once a user query is inputted, lists of documents are presented to the user, together with the document titles. Because the document titles in Encarta are carefully chosen, they give the user a good idea of their contents.

Therefore, if a user clicks on a document, it is likely that the document is relevant to the query, or at least related to it. After applying classification in search logs the clusters will be formed. Clusters are similar search things.

3.4 Providing Security

For providing privacy, the proposed approach introduces a search log publishing algorithm called ZEALOUS. ZEALOUS ensures probabilistic *Differential privacy* and it follows a simple two-phase framework.

In the first phase, ZEALOUS generates a histogram of items in the input search log, and then removes from the histogram the items with frequencies below a threshold.

In the second phase, ZEALOUS add noise to the histogram counts, and eliminates the items whose noisy frequencies are smaller than another threshold.

4. Conclusion

The results of this research enable search engine companies to make their search logs available to researchers without disclosing their user's sensitive information. Search engine companies can apply our algorithm to generate statistical probabilistic *Differential privacy* while retaining good utility. In proposed technique to release useful information about infrequent keywords, queries and clicks in a search log while preserving user privacy. My proposed approach provides privacy by publishing infrequent keywords and queries thus searched user queries cannot be matched to a particular user

Acknowledgment

We specially thank our Director, HOD, Guide and the staff members of SRM University for rendering us valuable information and encouraging us throughout our research.

References

[1]. A. Blum, K. Ligett, and A. Roth, “A Learning Theory Approach to Non-Interactive Database Privacy,” Proc. 40th Ann. ACM Symp.Theory of Computing (STOC), pp. 609-618, 2008.
[2]. Y. He and J.F. Naughton, “Anonymization of Set-Valued Data via Top-Down, Local Generalization,” Proc. VLDB Endowment, vol. 2,no. 1, pp. 934-945, 2009.
[3]. J. Brickell and V. Shmatikov, “The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing,” Proc. 14th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), 2008.
[4]. K. Nissim, S. Raskhodnikova, and A. Smith, “Smooth Sensitivity and Sampling in Private Data Analysis,” Proc. Ann. ACM Symp.Theory of Computing (STOC), 2007.
[5]. C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating Noise to Sensitivity in Private Data Analysis,” Proc. Theory of Cryptography Conf. (TCC), 2006.
[6]. M. Go¨ tz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke,“Privacy in Search Logs,” CoRR, abs/0904.0682v2, 2009.
[7]. J. Han and M. Kamber, Data Mining: Concepts and Techniques, first ed. Morgan Kaufmann, Sept. 2000.
[8]. Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri, “Effective Anonymization of Query Logs,” Proc. ACM Conf. Information and Knowledge Management (CIKM), 2009.
[9]. R. Jones, R. Kumar, B. Pang, and A. Tomkins, “I Know What You Did Last Summer: Query Logs and User Privacy,” Proc. ACM Conf. Information and Knowledge Management (CIKM), 2007.