Transactions on Data Privacy 4 (2011) 31–50

Anonymous Search Histories Featuring Personalized Advertisement – Balancing Privacy with Economic Interests

Thorben Burghardt*, **Klemens Böhm***, **Achim Guttmann***, **Chris Clifton****

*Karlsruhe Institute of Technology, Am Fasanengarten 5, Karlsruhe, 76131, Germany. **Purdue University, 305 North University Street, West Lafayette; IN, 47907, USA.

E-mail: firstname.lastname@kit.edu, clifton@cs.purdue.edu

Abstract. Search engines are key to finding information on the web. Search presently is free for users – financed by targeted advertisement. Today, the current search terms determine the ad placement. In the near future, search-engine providers will make use of detailed user profiles for better ad placement. This puts user privacy at risk. Anonymizing search histories, which is a solution in principle, gives way to a trade-off between privacy and the usability of the data for ad placement. This paper studies this tradeoff systematically. To this end, we implement an algorithm for the anonymization of search histories which is flexible regarding the target function. It can retain frequent terms or terms where corresponding ads are clicked with a high probability, keep up the number of users it can derive interests for, etc. We quantify the usefulness of the anonymized log for ad placement in a broad way, e.g., by estimating the number of ad clicks or of ad impressions, based on marketing data from Yahoo! As a result, anonymized search logs are still useful for ad placement, but this very much depends on the target function.

1 Introduction

Advertising is the main source of revenue of search-engine providers (providers in what follows), so that search engines are available without charging users explicitly. More specifically, ad placement tries to predict the *user intent*, i.e., the current information need of the user, in order to display ads of interest. Revenue depends on prediction quality. To improve ad placement, providers are about to rely on search logs; their privacy policies indicate that they are ready and willing to do so.[1]

*Note: A four page abstract of this work appears in *Proceedings of CIKM 2010*. This submission contains substantial additional detail throughout, particularly in details of data analysis and experiments.

[1] http://www.google.co.uk/intl/en/privacy˙cookies.html, Nov. 2009; http://privacy.microsoft.com, Jan. 2010; http://privacy.yahoo.com, Jan. 2010.

The public release of search logs by AOL in 2006 has shown (inadvertently) that query histories, i.e., sets of queries assigned to an individual according to his search engine account, cookies or the IP address, are individually identifiable information [1, 2]. The European Commission has taken this perspective as well [3]. In other words, storing and processing personal data has the important drawback that user privacy is at risk.

When processing and storing personal information, providers face obligations from data-protection law. Most of these obligations are difficult and expensive to comply with. In cases where providers have failed to fulfill obligations, this frequently has lead to negative publicity. Such obligations include requesting consent, informing individuals, going through complex processes to delete data, etc. [3, 4].

One way to ensure user privacy is anonymization. Use of anonymized search histories would not only improve user privacy, it would also free providers from those regulatory obligations. According to [4], a data set is anonymous if a person cannot be individually identified. [5] has defined anonymity for set-valued data. We refer to this notion as (k, m)-Anonymity: each combination of m terms out of a search history of a particular user is in the history of $k - 1$ other users as well. The party anonymizing the log can decide on the number m of attributes forming a potential identifier. We use their definition in this paper.

Example 1: [6, 7] has shown that about 67% of the US citizens can be identified by {date of birth, zip, gender}. Setting $m = 3$ when anonymizing the data would prevent this re-identification, by guaranteeing that each combination of three attribute values exists at least k times.

The higher m, the larger the number of attributes an adversary would need to re-identify an individual. The higher k, the more users are indistinguishable from each other.

Using other definitions of anonymity [8–10] instead would not solve the problem, as we will explain in Section 2.3. Analogously, anonymization techniques that rely on generalization [5, 11] or insertion of queries [12] are not applicable in our setting as well, as we will show. To achieve (k, m)-Anonymity, we will delete terms from the individual histories of users until the definition of anonymity is met.

We are not aware of any study where search-log anonymization according to (k, m)-Anonymity is accomplished by deleting terms. While, on a technical level, this is relatively easy to accomplish, the characteristics of the anonymization results are much less clear: Inherent to anonymization is a tradeoff between privacy and the quality of the anonymized data [13, 14], in our case the usefulness for ad placement. There are several ways to achieve the same level of anonymity for a log. Existing work on anonymization [5, 11, 13] proposes anonymization algorithms and studies their effectiveness. Each optimizes towards a single generic target, e.g., the number of generalizations of terms required, or the log size after anonymization. This is somewhat undifferentiated. For instance, the "optimal" anonymized log could vary for different business models of providers, e.g., pay-per-impressions, pay-per-clicks. Alternatively, as advertisers may have different advertising strategies as well [15], "optimal" might refer to, e.g., bidding on specific but expensive terms, bidding on cheap general ones, etc. This indicates that targeting at different characteristics of the log during anonymization will lead to results with quite different degrees of usability for advertisement.

Example 2: Based on the data we will use in our evaluation, we have measured the correlation of (i) the frequency of a term in the search log and (ii) the maximum bid Yahoo! recommends advertisers for the term. We have not found any significant correlation. This means that trying to keep as many frequent terms as possible during optimization will result in a log that may or may not contain many terms advertisers bid highly for.

Thus, anonymization should be flexible regarding the target function. Accordingly, to anonymize data in an optimal way, it is important to study the effects of different target

functions systematically.

Problem Statement Assume we are given levels of privacy and different target functions optimizing different characteristics of the anonymization result. We ask, is there a significant difference between the utility (regarding each characteristic and different levels of privacy) of the anonymized log when using different target functions? In other terms, what is the impact of different target functions on the utility of the anonymization result?

Contributions Relying on the definition of (k, m)-Anonymity, we evaluate the impact of the target function of the anonymization on the usefulness of the resulting search logs for ad placement. We consider generic characteristics of the anonymized log like large log size, retaining many users, as well as characteristics specific for advertisement, such as terms with a high marketing value, terms leading to many ad impressions etc. Regarding these characteristics, we measure how different anonymized logs generated with different target functions are. We do so for different levels of anonymity. As mentioned above, we have implemented a heuristic algorithm to achieve (k, m)-Anonymity. Our implementation is flexible regarding the target function.

The real world search log we use for the evaluation includes 3.5 million queries of 370,585 users. To obtain real world marketing data as well we have crawled the Yahoo! marketing portal. This portal offers information to clients planning advertising campaigns like estimates of the number of ad impressions or of clicks, or like the maximal bid recommended. We have retrieved these values for all terms in the search log.

Our main results are that (k, m)-Anonymity prevents re-identification similar to that in the AOL case, while retaining information in the search log valuable for advertisement. For instance, a real-world log anonymized for $m = 3$, $k = 100$ still contains terms leading to 61% of the ad clicks, according to our estimates. Further, we show the importance of the target function when anonymizing data for ad placement. For example, having the same level of privacy ($m = 3$, $k = 100$) can lead to 40% less clicks with another target function.

Paper structure: We discuss background and related work in Sections 2 and 3. In Section 4 we present our approach and evaluate it in Section 5. Section 6 concludes.

2 Background

In this section we will look at the data currently collected by search-engine providers (Section 2.1) and the status quo of search-engine advertisement (Section 2.2). Further, we describe the challenges of search-log anonymization in Section 2.3. Last, we introduce the concept of set-valued anonymization and define (k, m)-Anonymity in Section 2.4.

2.1 Data Acquisition

Providers currently collect huge amounts of personal information. One important motivation behind this is targeted ad placement. Microsoft, Google and Yahoo! all state exactly this rationale in their relevant privacy policies[1]. A provider may obtain data from users who are registered or from users who are not. In both cases this includes cookie information, IP addresses, – this now is our topic – query terms, etc. Whereas anonymizing IP addresses is relatively straightforward, e.g., masking out the lower bits, this is more difficult with query terms. Albeit with user consent, providers further collect name, zip code, gender, and birth date from individuals with, e.g., a Windows Live ID or a Google account. Clearly, this data

can be correlated with search logs. However, correlations of knowledge beyond the search log exceed the scope of this paper.

One might expect that unregistered users have a higher expectation of privacy. Unfortunately, this may not be the case; it is relatively easy to link queries in a sequence with high probability based on time and IP as well as query terms [16]. Long search histories alone may be an effective user profile [17] and are individually identifiable [1]. Google creates search histories for unregistered users[2], and building histories takes place by default.

Providers in turn face various obligations resulting from data-protection law when acquiring and processing data. This becomes particularly complex, when the provider operates internationally. Processing only anonymized data would reduce the need for functionality required by law. For instance, a search-engine provider operating under European law would not need to provide the means for users to revoke their consent they have given for use of personalized profiles, delete data, consider storage limitation and limitations regarding the correlation of the personal data with third party knowledge [4], etc.

2.2 Search-Engine Advertisement

In the very beginning, providers charged advertisers per impression. Today, advertisers mostly pay per click on their ads. Thus, the key to search-engine advertisement is to show those ads that are of interest to the user. To achieve this, providers offer platforms where advertisers can participate in ad-word auctions, e.g., Google AdWords[3]. The auction type most widely used is the generalized second price auction [15, 18]. Further, providers use some additional criteria to decide which ad to show, e.g., if the page referred to in the ad fits the content of the ad. With pay-per-click, providers improve revenue by placing ads matching the intent of the user, i.e., increase click probability. However, matching ads and intentions is difficult, as the following example shows.

Example 3: The query "Golf London" at `www.google.co.uk` returns five ads for golf courses, one for golfing vacations, and two "generic" ads. Starting with the sequence of searches "compact cars" and "corolla golf focus comparison" (with several clicks on car-related results), the query "golf london" brought up – the same golf course ads.

There obviously is room to improve estimating the user intent. The expectation behind storing search histories is that the utilization of (a) nearby queries will lead to more effective ad placement, and (b) when queries do not contain good keywords, longer histories allow ads of at least some interest to the query issuer based on past queries.

2.3 Challenges of Search Log Anonymization

Anonymizing search histories is difficult. *First*, inherent to anonymization is the trade-off between privacy and the quality of the data set after anonymization [13], such as usefulness for ad placement. There are different business models of providers, e.g., pay-per-click or pay-per-impression, as well as different advertising strategies, e.g., get many impressions but few clicks only, few impressions but a high click rate, etc. [15]. Thus, providers and advertisers tend to have different interests. The provider is interested in a high revenue, while the advertiser is interested in many impressions or clicks. It is unclear which target function to use when it comes to anonymization.

[2] `http://www.google.com/support/accounts/bin/answer.py?answer=54048`, Dec. 2009
[3] `https://adwords.google.com`

Second, the party anonymizing the log has no a priori knowledge about which query term or term combination is sensitive or might lead to an identification. For instance, one cannot readily use known techniques for health records. With health records, the party anonymizing the log knows which values identify an individual, e.g., the attributes name or insurance number; further, it knows which values are sensitive, e.g., the disease. Thus, commonly known anonymization techniques for health data are not applicable [8–10, 19].

Third, using generalization hierarchies, as done in [5, 11], is hardly feasible for search logs, since such hierarchies are not available for the search context with hundred thousands of distinct terms. For example, a join of WordNet [20] with the Altavista search log removes 85% of distinct terms from the log.

Fourth, inserting queries into the log to disguise the user intent [12] seems to be another way to achieve anonymity. However, this only prevents re-identification if identifying terms from one user are placed into logs of other users *and* are not distinguishable from the real queries of that user. Further, this obviously makes placing ads in a personalized way impossible.

2.4 (k,m)-Anonymity

In this section we first say why set-valued anonymization, i.e., (k, m)-Anonymity, fits our scenario. Next, we illustrate how to identify individuals in non-anonymized logs. We introduce the definition of (k, m)-Anonymity and its relationship to the AOL case.

Deciding for (k, m)-Anonymity, instead of the variants of anonymity proposed in, e.g., [12, 13, 21, 22], is due to the following observations and assumptions: First, when considering search logs, one can observe that combinations of terms of different queries of the same user, not only combinations within the same query, threaten user privacy. Example 4 will illustrate this. Further, we assume terms disclosed once may form a quasi-identifier, as do terms disclosed several times. For instance, a credit card number one can find in many logs, poses a similar threat when found once as when found several times. In other words, we consider the privacy threat of a term to be independent from its frequency in the log. While infrequent terms may be less indicative of individual preferences than frequent terms, and thus less revealing if disclosed, our goal is to *de-identify* search logs and free providers of the constraints of privacy law. The anonymized search log retains the structure of the original log, but only including terms that meet the (k, m)-Anonymity standard; this provides information such as term frequency for profiling while protecting against identifying the individual associated with that profile. In Section 5 we measure retention of both distinct and total terms, as use of frequent terms may help to build a profile of the (unidentified) user to better support search and target advertising. Next, we cannot anticipate which term combination will lead to an identification or is sensitive. Accordingly, the definition of (k, m)-Anonymity treats all terms in the same way, i.e., guarantees anonymity for any term combination possible.

To illustrate the issues just raised, we will now look at the AOL disclosure of search logs [23].

Example 4: In 2006, AOL disclosed millions of queries including 454 queries with 356 distinct terms of user 4417749. The log of this user contained searches for several individuals named 'Arnold' as well as shops and landscapers in 'Lilburn', 'Georgia'. 166 users other than user 4417749 had searched for 'Arnold', 5 others for 'Lilburn' and 1379 others for 'Georgia' (Table 1, $m = 1$). Thus, this data set is $(k, 1)$-anonymous for $k = 6$. However, as can be seen in the table for $m = 2$, 'Arnold & Lilburn' in combination form a quasi-identifier, i.e., identify one person. This (and a phone book) let reporters find out that Thelma Arnold is user 4417749.

term	m	frq.
Georgia	1	1380
Arnold	1	167
Lilburn	1	6
Arnold & Georgia	2	15
Lilburn & Georgia	2	5
Arnold & Lilburn	2	1
Arnold & Lilburn & Georgia	3	1

Table 1: re-identifying 4417749

As the example implies, we have to choose a definition of anonymity considering combinations of terms. [5] has introduced the following definition for the market-basket scenario. We have slightly modified it for search logs.

Definition 1 ((k, m)-Anonymity)**.** A search log is (k, m)-anonymous if any combination of m or fewer terms out of the history of a user appears in at least $k - 1$ histories of other users. Further, we say that (k, m) is the level of privacy of the search log.

Looking at the frequency of term combinations of Example 4 implies that 'Arnold & Lilburn' form a quasi-identifier, for combinations of length $m = 2$ already, i.e., the AOL search log is not anonymous for $m > 1$. (k, m)-Anonymity would have assured that this combination exists at least k times, i.e., for $k = 2$, $m = 2$, Thelma Arnold would not have been identified.

Note that k and m are independent. However, if a search log is anonymous regarding (k, m), it is for $(k - 1, m)$ as well. The same holds for the parameter m, i.e., a log that is (k, m)-anonymous is $(k, m - 1)$-anonymous.

3 Related Work

We now give an overview of anonymization techniques, with a focus on search logs.

Many existing techniques require a priori knowledge about which attributes are sensitive, i.e., threaten the privacy of the individual. In health contexts, for instance, there are attributes potentially identifying an individual, like 'name' or 'insurance id', and sensitive attributes, like 'disease'. Prominent examples of anonymization techniques where identifying attributes can clearly be distinguished from sensitive ones are k-anonymity [8], l-diversity [9], t-closeness [10] and anatomy [24]. These approaches leave the sensitive attribute untouched, however, they (mostly) generalize the identifying attributes, here given by an insurance code <68766, diabetes>, <68799, cancer> to <687**, diabetes>, <687**, cancer>. Thus, for this 2-anonymous example, an adversary cannot allot the disease to one individual.

With search logs, any term combination can be sensitive as well as identifying. [17] demonstrates how to map sequences of queries to the gender, age, and location of the issuer. Such a mapping reduces the set of potential issuers by a factor of 300–600, compared to random selection.

Anonymization techniques specific to search logs exist. [13] proposes a two-step approach: The first one is to remove infrequent queries. The second one is to create for each topic a user has searched for a virtual identity. Afterwards, depending on the topic, the authors assign each query of that user to the corresponding virtual identity. To illustrate the second

step, they distinguish between queries of user u regarding, say, soccer, and those regarding cooking. Then they assign the queries to virtual user u_1 for soccer and to u_2 for cooking. However, the first step removes 97% of the distinct queries and 64% of the log size – a significant loss. Further, distributing queries among several users reduces the usefulness of the log, e.g., for ad placement. [21] analyzes another approach called 'token-based hashing' and shows its ineffectiveness as an anonymization technique. [22] studies the release of query click graphs where nodes are both queries and urls and edges correspond to clicks on URLs based on a query issued. They define a query to be anonymous if the frequency plus some noise is above a predefined threshold.

[5,11] are most closely related to our work. They do without the assumption that one knows which attributes are identifying or sensitive. [5] is first to state the problem and to propose the solution of anonymization of set-valued data, for market-basket analysis. Attacks may be successful if the attacker knows some products purchased. To cope with this privacy threat, they propose (k, m)-Anonymity (Section 2.4). Next, they propose three algorithms, all based on generalization. However, they only consider short transaction logs (2k–15k) and item domains of a size of 40. Search logs tend to consist of several millions of queries and more than 300k distinct terms. Further, [5] requires a generalization hierarchy. For instance, a term like 'bread' is replaced by the more general term 'food' if 'bread' is infrequent. This is not practical for search logs. Last, their algorithms feature global recoding, i.e., when a term is generalized, this takes place for all of its occurrences.

[11] is an extension of [5]. Based on the same definition of anonymity, they tackle the problem of finding an appropriate m by applying k-anonymity to entire queries. In other words, they look at (k, m)-Anonymity for $m = max$(transaction-size). They also generalize data and propose an algorithm. They use a top-down approach instead of a bottom-up approach and use local recoding, as we do. They show that their algorithm generalizes terms more efficiently than [5]. They also apply their approach to query logs. However, in lieu of a real term hierarchy, they simply use an alphabetically ordered generalization tree and a tree with generalization using the limited number of nouns from WordNet. They themselves state that using WordNet is 'by no means a perfect approach', as many popular made-up words, names and terms like 'myspace', 'amazon', and 'facebook' are missing.

Related work described analyzes the anonymization result based on generic characteristics, e.g., log size. We in turn are first giving a comprehensive utility analysis with realistic marketing and search data. Our approach deletes terms, and we do so locally for the history of each user. Due to the reasons mentioned above we cannot deploy the algorithms from [5, 11] in our context and cannot compare our algorithm to them.

4 Approach

In this section we first describe target functions relevant in the advertisement context (Section 4.1). Next, we give the problem statement (Section 4.3) and the realization of our approach (Section 4.4).

4.1 Information Loss and Usability

We now describe different target functions. We have derived the ones specific for advertisement by analyzing the Yahoo! marketing platform, i.e., information Yahoo! offers to advertisers to optimize their campaigns.

term	frequency	users	max bid ($)	ad impressions	clicks	revenue ($)
or	**142041**	2170	0.42	56574	28	12
com	74453	**31974**	0.69	203681	549	380
MacMall	4	4	**919**	5775	786	722334
ATT	270	146	0.5	**5009356**	3747	1874
Lowes	173	101	0.05	491515	**211489**	10575
AdultFriendFinder	6	3	334	10020	2742	**915828**

Table 2: Illustration of Utility Criteria

 # distinct terms. Providers may want to keep a maximum number of distinct terms in the anonymized log to understand the interests and intentions of their users well. Further, this objective gives way to broad advertisement.
log size. Providers may want to keep a log file of maximum size, e.g., prefer frequent occurrences of a term over distinct, but infrequent terms. This could be of interest in the case of several advertisers willing to place ads on the same (frequent) term.
 # users. Deleting terms from the history can eliminate all terms of a user. Providers may want to have some information on many (non-identifiable) users, even if it is vague.
bid. Most providers are paid per ad click. Thus, they may want to keep those terms advertisers issue high bids for. We calculate the utility as $\mathsf{bid} \cdot \mathsf{term\_frequency}$, i.e., weight bid with the term frequency. We will do so for the following target functions as well.
estimated clicks. Depending on the business model (pay-per click) providers might prefer having ads clicked frequently over terms with high bids but only few ad clicks.
ad impressions. An alternative business model is pay-per ad impression. Impressions means how often a specific ad is shown, independently from being clicked.
revenue. This criterion keeps those terms in the logs where $\sum_{term} \mathsf{clicks(term)} \cdot \mathsf{bid(term)}$ is maximal.

Example 5: For the search log we will use in the evaluation, Table 2 lists the term with the highest frequency, the one most users have issued, the one whose bid recommended is maximal and the one whose expected number of impressions/clicks/revenue is maximal. The two highest frequency terms (or and com) are not very interesting, and would often be considered candidates for a stopword list.

Note that frequent terms may be of low value, but are also unlikely to be identifying. We do not make such an apriori judgement, leaving it instead to the target function. Thus, different criteria lead to different rankings of the terms and thus influence the anonymization.

In fact, 'bids' and 'revenue' as introduced do not represent the real revenue of the provider. This is because providers mostly use second price auctions (or related), place several ads in parallel etc. However, to our knowledge, this is the first study assigning to an anonymized log real-world marketing data as indicators for its value for ad placement.

4.2 Correlation Analysis

So far, we have identified relevant characteristics of anonymized search logs, i.e., characteristics one might target on during anonymization. To answer our research question regarding the impact of the target function, it may be insightful to analyze the characteristics of the input data, i.e., of search logs and marketing data.

Example 6: Consider again Example 2: A term that is frequent is not necessarily one Yahoo! recommends high bids for. When trying to keep frequent terms during anonymization, this does not mean that the resulting log will contain many expensive terms. On the other hand, suppose that those two criteria were highly correlated. In this case, it would not make much of a difference

		revenue	impressions	clicks	max bid	user frq.
term frq.	corr.	.002	.111*	.024*	.001	.761*
	sig	.326	.000	.000	.348	.000
	N	64,694	117,916	64,694	117,916	117,916
user frq.	corr.	.003	.151*	.031*	.002	
	sig	.259	.000	.000	.254	
	N	64,694	117,916	64,694	117,916	
max bid	corr.	.579*	.002	-.003		
	sig	.000	.296	.225		
	N	64,694	117,916	64,694		
clicks	corr.	.040*	.400*			
	sig	.000	.000			
	N	64,694	64,694			
impressions	corr.	.020*				
	sig	.000				
	N	64,694				

Table 3: Pearson Correlation Analysis of Input Data

which target function to use during anonymization: The anonymization result would contain both many frequent and many expensive terms.

We compute the pairwise correlation between all relevant characteristics we have identified in Section 4.1, i.e., for each term we compare the number of users having used the term to the maximal bid Yahoo! recommends, to the frequency of the term, etc. Table 3 shows the Pearson correlation, the significance of the correlation, and the number of terms used for the correlation analysis (N) for each combination. As Yahoo! gives no estimate of the number of clicks for 45% of the terms, correlations including the variable clicks have a smaller N. Results marked with '*' are significant at the 0.01 level (1-tailed). For 9 combinations we see a significant correlation, for 6 combinations in turn we do not find a correlation. All combinations except one (clicks, max bid) have a positive correlation. Thus, we expect that in many cases targeting at one variable when anonymizing the log will have positive effects on the other characteristics as well. However, this strongly depends on the variable. For instance, for 'max bid' the only significant correlation is with the variable 'revenue'. This is not surprising since 'max bid' is the calculation basis for 'revenue'. The variable 'clicks' in turn correlates with four others. We will analyze the implications of the correlations on the anonymized log in Section 5.3.

We will focus on the utility of the anonymized log. However, as we know from [10], knowing the distribution of sensitive information within a data set, e.g., computed by correlation analyses as we have done here, might lead to privacy threats as well. It is an open question how to handle this in the context of set-valued anonymization. This is not the focus of the current article – we look at the level of privacy defined by k and m only.

4.3 Formal Problem Statement

It is an open question if an anonymized search log is of value for ad placement/marketing purposes. To obtain a (k, m)-anonymous log S' from an initial log S, we delete terms from the query histories of the users, as we will explain. The log S is a set of search histories of the users. Here, a history is the set of terms of queries a user has issued. The choice

of the term to delete from the history of user i (H_i) is crucial and depends on the target function (ta). A target function takes a set of terms and returns the term to delete that has the lowest utility regarding ta. As there are different target functions, there can be different anonymized variants S' of a log S with the same level of privacy. In what follows, U_{ta} is the utility of the anonymized log. It is the sum of the utilities (with respect to the target function considered) of each term in the anonymized log S'.

Example 7: Think of a search log S with three users and individual histories $H\_1 = \{a, b, c\}$, $H\_2 = \{a, b, d\}$, and $H\_3 = \{b, c, d\}$. $a, \ldots, d$ are terms. ta, here, targets at terms with high bids. Let $m = 2$, $k = 2$ be the requested level of anonymity. We can see that $H\_1' = H\_2' = \{a, b\}$ and $H\_3' = \{b\}$ is $(2, 2)$-anonymous, as is $H\_1' = H\_3' = \{b, c\}$ and $H\_2' = \{b\}$. However, if advertisers bid highly for a and not for c, the first variant would have a higher utility.

Problem statement: Given different levels of privacy and different target functions ta and $ta*$, is there a significant difference between the utility of the log anonymized using ta and the one using $ta*$? In other terms, what is the impact of different target functions on the utility of the anonymization result?

We can expect the best utility with respect to a target ta^ϕ (U_{ta^ϕ}) when using the target function ta^ϕ in our anonymization algorithm, and not another one. However, it is important to notice that greedy heuristics do not necessarily lead to the optimal result.

Example 8: We now extend Example 7 with a fourth history H_4={a,c,e}. Then, H_1'={c}, H_2'= {b,d}, H_3'={b,d} and H_4'={c} form a (2,2)-anonymous log (S'), as do H_1"={a,b,c}, H_2"={a,b}, H_3"={b,c} and H_4"={a,c}, subsequently referred to as S". Suppose that the utility of a term is $u = \text{bid} \cdot \text{clicks}$, and u_a=\$1, u_b=\$1.1, u_c=\$1.2, u_d=\$1.3, u_e=\$1.4. For the first variant S', we have removed a instead of d, and a instead of b, as a has the lowest utility compared to all others. Thus, the first variant has utility $U_{\text{bid}*\text{click}}(S') =$ \$7.2. In the second variant we have removed d, which has a high utility. However, this results in $U_{\text{bid}*\text{click}}(S'')$ =\$9.9, i.e., a higher overall utility.

4.4 (k,m)-Anonymity Algorithm

In this section we describe our anonymization algorithm, which is heuristic in nature. It is different from existing ones which are not applicable for search logs (cf. Section 2.3). However, existing algorithms could be adapted to work in our context as well without difficulty, and we do not claim the algorithm is a significant contribution.

In our algorithm (Algorithm 1) S is the search log, H_i is the history of user i, and $S = \bigcup H_i$. $|N|$ is the number of users. fis is the set of all term combinations that are frequent, i.e., that have occurred in queries issued by k users. List C_i contains all combinations of terms in H_i of a maximal size m. Building term combinations of size smaller than m is due to our assumption that privacy protection is optimal if the entire history of a user exists k times. Thus, if there is a history of size smaller than m, we keep it in the log if the terms appear in at least $k - 1$ other histories. This only ensures (k, 'size of history')-Anonymity for this user. c_{ij} stands for the j-th combination in C_i. We denote set-valued assignments by $\Leftarrow$.

Example 9: Continuing Example 7 with $k = 2$ and $m = 2$, fis would contain $\{a, b\}$, C_1 is $[\{a, b\}, \{a, c\}, \{b, c\}]$, $c_{1,1}$ is $\{a, b\}$.

Initially, we use the fp-growth implementation of [25] to find frequent combinations of terms (Line 3). The next step is to create for each user $n_i \in N$ all possible combinations of terms of size m using only terms in H_i and all subsets of it. Afterwards, we test for each of these

Algorithm 1 (k, m)-Anonymity - Greedy Heuristic

```
1: S ⇐ Original search log, S' ⇐ {};
2: N := Total number of users, k, m := anonymity parameters from Definition 1
3: fis := fpgrowth();
4: while S != S' do
5:    S' ⇐ S
6:    for i = 1 to |N| do
7:       H_i ⇐ all Terms of user n_i
8:       C_i ⇐ combinations of terms in H_i of size ≤ m
9:       for j = 1 to |C_i| do
10:         c_ij := C_i[j]
11:         if support(fis, c_ij) < k then
12:            r := target_fct(c_ij, ta)
13:            H_i ⇐ H_i \ r
14:            fis ⇐ update_fis(H_i)
15:         end if
16:      end for
17:   end for
18:   S ⇐ ⋃ H_i
19: end while
```

iteration	term	m	frequency	state
1.	*eugene*	3	4	keep term
1.	*chicago*	3	2	candidate
1.	*smith*	3	2	candidate
2.	*chicago*	2	4	candidate
2.	*smith*	2	4	candidate
3.	*chicago*	1	6	keep term
3.	*smith*	1	4	delete term

Table 4: Deciding for a term to delete

combinations if they are element of fis, the set of frequent term combinations. If this is the case with a specific term combination, it is (k, m)-anonymous. However, if $support(c_{ij}) < k$ we have to delete one term in the combination from H_i.

Our heuristic is greedy in the sense that it deletes the term that fits the target function in the currently best way. We compute the term to delete in Line 12, by function *target_fct*. *target_fct* can behave in three different manners, namely 'random', 'fis', and focusing on the generic/specific target functions. The first alternative, 'random', means that a term to remove from c_{ij} is chosen randomly. We use this as the baseline when evaluating the different target functions. The second alternative 'fis' is to operate on frequent itemsets only. This behavior is most related to existing work on set-valued data. We delete the term that is part of the fewest term combinations in fis. If more than one term has minimal frequency, we look at the frequency of these terms for $m' = m - 1$. We repeat this until there is only one term with the minimal frequency, or $k = 0$. In the latter case we randomly choose one of the remaining terms.

Example 10: Suppose that there is the combination of terms {eugene, chicago, smith} $\notin fis$

as given in Table 4. Thus, we have to delete one of these terms. We calculate how often each term occurs in the frequent item set. Here 'eugene' appears 4 times, 'chicago' and 'smith' both two times (Column 'frequency'). Accordingly, we keep 'eugene' and delete one of the remaining terms. We decrease m, and, again, 'chicago' and 'smith' both occur equally often (four times). We decrease m once more. 'smith' is rarer, i.e., will be deleted.

Algorithm 2 target_fct(c,ta)

```
1: c := term combination of size m
2: v ⇐ {}; /*term utility*/
3: for i = 1 to m do
4:    v ⇐ get_term_utility(i, ta);
5: end for
6: return  return a random element with utility = v.min();
```

Finally, Algorithm 2 describes the third alternative. Here, the utility of a term is according to the target function ta. We return the term whose utility is minimal (Line 6) with respect to ta. If two or more terms have the same utility, we pick one randomly. Our implementation supports $ta \in$ {random, logsize, users, bid, clicks, impressions revenue}.

Having deleted a term, we have to update fis (Alg. 1, Line 14), i.e., we decrease the support of all combinations containing this term. It is important that only those combinations must be considered that one can build by combining terms $t \in H_i$. The combinations of terms with new support $\leq k$ are deleted from fis.

Our algorithm processes one user after another. Thus, removing terms and (former) frequent term combinations can affect histories of users already processed. So we apply our heuristic repeatedly until no more terms need to be deleted.

5 Evaluation

In this section we answer our research question. We first describe the data we use in the evaluation (Section 5.1). Next, even though efficiency of our heuristic is not our primary focus, we give an intuition on the runtime behavior of our algorithm (Section 5.2). We detail the measures used to study the usefulness of anonymized logs for advertisement (Section 5.3). We present results comparing different target functions and values of k in Section 5.4. Section 5.5 focuses on the impact of the parameter m, i.e., the size of the term combinations deemed potentially identifying.

5.1 Search and Marketing Data

In this section we briefly introduce the data used in our evaluation, namely the search log from Altavista and the marketing data from Yahoo!.

Altavista Log. The Altavista search log was published in 2002, covering one day of queries[4]. It contains 3.5 million queries of 370,585 users. The log is of the form `Q=⟨user_id, query, timestamp⟩`. We have applied the following preprocessing steps: (i) Our objective is to anonymize personally identifiable information, not data from metasearch engines, crawlers etc. If an adversary can identify such a service in an anonymized log, this would

[4]Altavista transaction logs from 2002, provided by Jim Jansen (jjansen@acm.org)

reduce the indistinguishability of k by one, with every service identified. Thus, we have removed queries issued by bots according to the (simplistic) method from [26] by removing users with sessions containing more than 100 queries. (ii) We treat all characters except [A-Za-z0-9À-ÖÙ-Ýß-öù-ž] as whitespace. For example, the two queries 'ad–placement' and 'ad placement' only differ in one special character and are normalized to be the same term sequence. (iii) Privacy threats can result from combining terms of different queries of the same person. We extract the terms from each query and store the result in a relation with schema `QT=⟨user_id, query_id, term_id⟩`. Our preprocessing leads to 1,846,134 queries of 367,803 users with 251,115 distinct terms and 5,501,825 occurrences of these terms. Note that the number of terms is above the size of any conventional dictionary, e.g., English or German, due to names ('myspace'), error codes ('oraXXXX'), etc.

Yahoo! Search Marketing Data. We are interested in the value of anonymized log data for advertisement. To this end, we assign a value to each term. The Yahoo! marketing site[5] provides estimates for (i) the number of `impressions` of an ad for a given term, (ii) the number of times the ad will be `clicked`, and (iii) the maximum `max_bid` Yahoo! recommends advertisers use for the term (based on a given total budget). We have crawled the site and collected this marketing information for the approximately 250,000 distinct terms contained in the Altavista search log.

5.2 Runtime Behavior

Our focus is to compare the impact of different target functions rather than achieving an optimal computation time. Accordingly, the following evaluation will give an intuition on the runtime behavior of our algorithm for different k and m. For the reasons already described, we will not compare it to other algorithms.

We use standard hardware (Dual-Core AMD Opteron 2218), 8GB RAM, Java, and a single threaded implementation. Further, we use a relational database to store the search histories during the anonymization process, e.g., we track each deletion of a term required for any target function, each level of privacy and iteration of the algorithm. The interaction with the database has the most significant impact on the runtime behavior.

Comparing the time required for the computation of the anonymized log and each target function shows that the target function requiring the most computation time is operating on frequent term combinations only (the second case in Section 4.4). The time for the worst case, i.e., small k, is particularly interesting. The larger k, the smaller the vocabulary and, thus, the time needed.

The first iteration for each k requires to delete a large number of infrequent terms and term combinations. Accordingly, the first iteration requires most time, e.g., for $k = 2$ 02h:55m:58s. With the second iteration, the time already drops to 09m:25s, i.e., a factor of around 17. For $k > 2$, the time for the first iteration is significantly reduced, e.g., to 7m:41s for $k = 5$. For $m = 2$, the average time per iteration is 4 minutes, the average number of iterations 6.7.

In qualitative terms, this observation also holds for $(k, 3)$-Anonymity. Nevertheless, with 90h:08m:24s for $k = 2$ when starting from the initial log file, the first iteration is significantly more cost intensive than for $m = 2$ (factor 31). This is due to the number of possible term combinations. For example, the largest T_i contains 495 terms. According to Algorithm 1, we have to test $\binom{493}{3} + \binom{493}{2} + \binom{493}{1} = 19,970,937$ combinations to be frequent for this user

[5] `http://sem.smallbusiness.yahoo.com/searchenginemarketing/marketingcost.php`

m	k	dist. terms %	logsize %	users %	bid %	clicks %	impressions %	revenue %
2	2	10.88	45	95	51	85	85	61
	10	2.12	29	87	33	75	71	41
	20	1.09	26	83	29	71	67	36
	40	0.56	23	78	25	68	63	33
	100	0.18	19	70	20	61	58	27
3	2	10.86	37	95	42	78	76	55
	10	2.12	25	87	31	71	67	43
	20	1.11	23	83	28	68	64	39
	40	0.56	21	78	24	65	62	35
	100	0.18	18	70	19	61	57	30

Table 5: Quality of anonymized search log

only. However, starting from $(2, 2)$-Anonymity, we can calculate $(2, 3)$-Anonymity within 06h:35m:22s, i.e., in 7% of the time required. Again, for $k > 2$, the duration of the first iteration is reduced. For $k = 5$ the first iteration lasts 48m:45s. The average time per iteration has been 13 minutes with 4.8 iterations on average.

5.3 Usability for Ad Placement

In this section we will first quantify generic characteristics of the anonymized log, i.e., the number of distinct terms in it, its size, and the number of users with queries remaining. Afterwards, we focus on the characteristics specific for advertisement, namely the maximal bid Yahoo! recommends, the number of clicks, the number of ad impressions estimated and potential revenue. Note that we have specific target functions for log size, the users with anonymous queries remaining, bid, clicks, impressions and revenue. For each target function, in combination with values $m = 2$ and $m = 3$, and samples of k between 2 and 100, we compute the anonymized log. Table 5 shows the impact of anonymization in percent.

In all cases, each target function also leads to the best result with respect to its focus. For example, targeting on the log size also leads to the largest log compared to all other target functions. We do not have a target function for the number of distinct terms, i.e., here, we only present the best outcome of any target function. Our results are given in Table 5. The upper half are results for $m = 2$, $k = 2, 10, 20, 40, 100$, the lower half the ones for $m = 3$.

distinct terms. Column 'dist. terms' shows that even for the lowest level of anonymity, i.e., $m = 2$, $k = 2$, the number of distinct terms is significantly reduced, to only 10.88%. For the highest level of anonymity ($m = 3$, $k = 100$), only 0.18% distinct terms remain in the anonymized log. The major reason for this is the long tail effect of search logs, i.e., there are many terms that are infrequent and few that are frequent. However, there still are about 500 distinct terms, allowing for a relatively broad set of ad topics.

log size. With $m = 2$, $k = 2$ there is a significant drop off of the log size, compared to the initial size (55%). This, again, is due to the long tail effect of search logs. For $m = 2$, $k = 20$, we maintain 26% of the log size, for $m = 3$, $k = 100$ this is 18%. This result means that, even if we can keep only between 11% and 0.18% of distinct terms, the remaining terms are frequently used, i.e., terms an advertiser might be interested in as well.

 # users. Independently of m, and for small k, the anomymization result contains histories for 95% of the users, 5% of the users 'are lost', as we delete their entire histories to achieve (k, m)-Anonymity. Even for large k, e.g., $k = 100$, we maintain some data for 70% of all users. We deem these numbers promising to derive general interests of the users.

Next to the generic measures we apply measures specific for the advertisement context.

bid. Our results (Column 'bid') show that after the anonymization for $m = 2$, $k = 2$ the sum of the bids for terms included in the anonymized log still is 51%, compared to the

m	k	dist. terms %			users %			logsize %		
		R	W	fis	R	W	fis	R	W	fis
2	2	0.00	4.16	4.08	13	2	0	2	0	0
	10	0.00	0.92	0.90	11	2	0	12	1	0
	20	0.00	0.40	0.39	11	2	1	17	1	0
	40	0.00	0.16	0.15	11	2	1	21	2	0
	100	0.00	0.00	0.14	11	2	1	28	2	0
3	2	0.00	4.18	4.07	10	3	0	2	0	0
	10	0.00	0.88	0.77	10	2	0	12	1	0
	20	0.00	0.31	0.27	10	2	1	17	1	0
	40	0.00	0.16	0.14	10	2	1	21	2	0
	100	0.00	0.02	0.00	10	2	1	28	2	0

Table 6: Impact of generic target functions

m	k	bid %			clicks %			impressions %			revenue %		
		R	W	fis	R	W	fis	R	W	fis	R	W	fis
2	2	20	13	12	28	15	8	31	9	6	20	5	5
	10	17	13	11	34	28	13	32	14	8	16	5	5
	20	16	12	11	35	33	15	32	15	9	16	5	5
	40	14	11	9	38	36	18	31	16	9	15	5	5
	100	13	9	7	37	37	19	30	16	10	15	5	5
3	2	15	14	11	32	31	12	29	16	7	21	12	10
	10	16	14	11	39	40	21	34	19	11	22	12	12
	20	15	13	11	40	41	24	34	19	12	21	12	12
	40	14	11	9	40	41	25	33	20	12	19	11	11
	100	13	9	7	39	40	25	32	19	12	19	10	10

Table 7: Impact of specific target functions

original one. For $m = 3$, $k = 100$ it is still 20%.

clicks. For large $k = 100$ and $m = 3$, the terms still in the log account for 61% of the clicks estimated. For $m = 2$, $k = 2$, the rate even is 85%.

impressions. Our results (Column 'impressions') are very similar to those for clicks. We can maintain those terms in the log leading to 57% to 85% of all ad impressions, depending on the level of privacy.

revenue. The revenue combines the bid of the advertiser, the number of clicks and the term frequency. Our results (Column 'revenue') are that revenues vary between 30% and 61%, again depending on the level of privacy.

5.4 Impact of the Target Functions

In this section we compare the different target functions. Table 6 contains our results for the number of distinct terms, log size and users, i.e., the generic target functions. Table 7 gives the results for the bid recommended, estimated clicks, ad impressions and revenue, i.e., our specific target functions. Again, we give results relative to the original log. For each measure, we give the difference of the best result achieved compared to random (Column 'R'), to the worst target function other than random ('W') and to that function working on

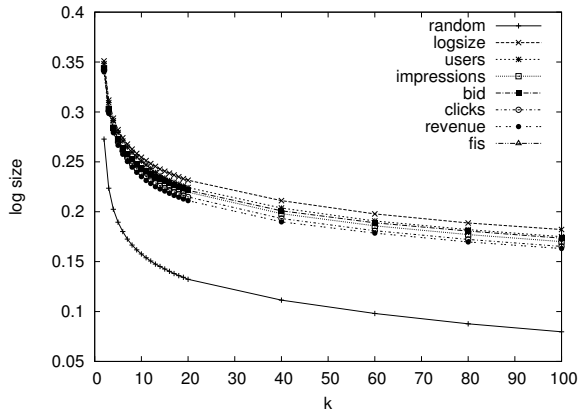

Figure 1: log size per target function

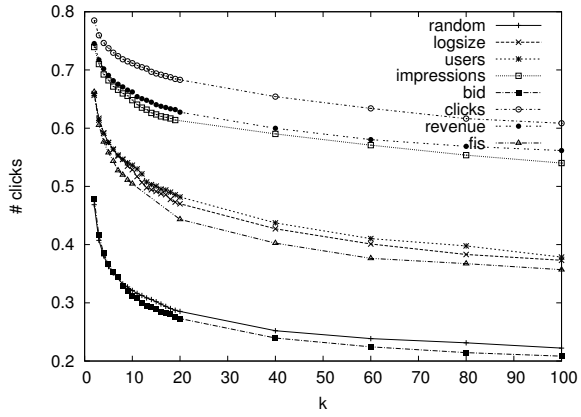

Figure 2: estimated clicks per target function

frequent term combination only (fis). Random is the average of two runs.

Our first result is that, for the generic characteristics 'log size', 'distinct terms' and 'number of remaining users in the anonymized log', the difference between the best target function and the worst one is less than 4.2% in all cases. Figure 1 shows the resulting log size for the different target functions and $m = 3$, $k = 2 \ldots 100$, to give an example. Compared to random, the difference of the log of up to 28% is relatively high. Thus, even though the impact of the target function is relatively small for the generic characteristics, it clearly is different from random. Targeting on many users also leads to the second largest log size. This is due to the high correlation between the users having searched for a term and the frequency of the term (.761, Table 3).

Our second result is that, for those characteristics specific to advertisement, the difference between the best and the worst target function is significant. For example, for the number of clicks estimated (Figure 2), the best target function (clicks) and the worst one (bids) have a difference of more than 40%. The reason for this large gap is that clicks and max bid are the only variables having (even if not significant) a negative correlation (Table 3). Interestingly, for the distinct terms for which we do not have a specific target function,

randomly deleting terms outperforms all others. Thus, the target function has a significant impact on the usefulness of the anonymized log.

5.5 Impact of the Term-Combinations Size

An important decision for the party anonymizing the log is the choice of the value of m, i.e., the size of the term combinations potentially leading to a re-identification. Recall that $m = 2$ would have prevented the re-identification in the AOL example (Section 2.4). We have evaluated our results for $m = 3$ as well. Surprisingly, the effect of m is minimal, compared to that of k. For $k \geq 10$, the difference of the outcomes for any target function and $m = 2$ and $m = 3$ is less than 4%. For $k = 100$ the values are mostly equal. This observation also holds for the different target functions. For the generic ones, the differences are under 1% in all cases. The largest difference of at most 7% is for revenue. We conclude that, for large k, a value of $m < 3$ would yield an unnecessarily low level of anonymity, without increasing the usefulness of the log.

5.6 Discussion

We have shown in some detail the impact of changing the anonymity factor k. We have also shown that for small values of m, changes have little impact. But what about other factors?

History size: The length of history we use to profile a user can be varied. While increasing the number of queries used to determine user interests could improve ad placement, it also results in more possible term combinations. This is not only a computational issue, it can reduce the terms kept. For example, suppose several users have issued queries for a and b; and several other users have issued queries for c and d. A query history containing $\{a, b, c, d\}$ would be able to keep only $\{a, b\}$ or $\{c, d\}$, as no combination involving both a and c, a and d, b and c, or c and d is frequent. But if that user's history was split into two (and the linkage between the two eliminated), the first $\{a, b\}$ and the second $\{c, d\}$, both could be kept in their entirety.

However, the number of combinations that are frequent would drop as well (given the shorter histories), so it is also possible that the suppression could increase.

Log size: Increasing the overall log size increases the number of frequent term combinations, although at some expense to run time. A more important issue is that some term combinations may be frequent in a narrow time range (e.g., names of candidates during an election); searching for those same terms at a different time may be highly identifying even though the (k, m)-anonymity definition is met globally. We therefore believe that the log size should be somewhat limited, and anonymization performed across a limited time (and possibly geographic) range, to ensure privacy protection.

k: Increasing k forces more terms to be deleted. This has a positive effect on the runtime behavior, as our algorithm has to test fewer term combinations. The effect on the utility of the anonymized log is negative. However, the long-tail effect inherent to search logs will cushion this, i.e., the impact of each increase in k is less than the previous increase.

m: While increasing m reduces the possibility of re-identification, it will force our algorithm to delete more terms (affecting utility). We have seen that for small values of m, this effect

is not that great – and examples of re-identification we have seen are prevented with small values of m. With the algorithm presented here, particularly Lines 8 and 11, and histories that are large, high m lead to hundred millions of tests for the frequency of term combinations $(\binom{T_i}{m}+\binom{T_i}{m-1}+\ldots+\binom{T_i}{1})$. As m increases, the result becomes closer to standard k-anonymity (which we have shown results in very high levels of suppression). Efficient algorithms for optimizing to a particular utility target with higher values of m remains an open challenge.

5.7 Summary

We have shown that, even though the number of distinct terms is quite small after anonymization, the remaining frequent terms lead to a log size between 18% and 45% of the original, depending on the level of anonymization. More importantly, the number of users remaining in the log is quite high. Further, the large share of clicks and ad impressions that are still feasible with the anonymized log (between 57% and 85%) is promising.

6 Conclusions

Targeted advertising is the primary source of revenue of search engines. Further refinements of ad placement are likely to rely on individually identifiable data. However, storing such data puts user privacy at risk, and its use by search engines raises various compliance issues. We have shown that, by means of anonymization, search engines can avoid using individually identifiable information for ad placement while still maintaining high effectiveness.

In this work we make use of the notion of (k, m)-Anonymity for set-valued data. We have implemented an algorithm that is flexible regarding the target function, e.g., to retain a large log size, to retain a log with terms leading to many ad clicks etc. With extensive evaluations on real world data we have shown that anonymized search logs contain valuable information for providers and advertisers. For instance, anonymization retains data of 70% to 95% of all users, depending on the level of anonymization. We can retain 61% to 85% of these clicks, based on retaining keywords with high click rate expectations, given the data available from Yahoo!. Choosing $m = 2$ or $m = 3$ turns out to have only a small impact. Further, the target functions are important. For instance, with a target function poorly chosen, the number of clicks on an ad may be 40% lower than necessary.

Thus, to preserve user privacy and free search-engine providers from compliance issues, anonymized logs to improve advertising could be a feasible approach.

References

[1] Michael Barbaro and Jr. Tom Zeller. A face is exposed for AOL searcher no. 4417749. *New York Times*, 2006.

[2] Andrew Kantor. AOL search data release reveals a great deal. *USA Today*, August 17 2006.

[3] Article 29 Data Protection Working Party. Opinion 1/2008 on data protection issues related to search engines. Technical report, B-1049 Brussels, Belgium, Office No LX-46 01/43, 4 April 2008.

[4] Directive 95/46/EC of the European Parliament and of the Council. *Official Journal of the European Communities*, 1995.

[5] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Privacy-preserving anonymization of set-valued data. *VLDB*, 1(1):115–125, 2008.

[6] Latanya Sweeney. Uniqueness of simple demographics in the US population. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, 2000. LIDAP-WP4.

[7] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. In *WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 77–80, New York, NY, USA, 2006. ACM.

[8] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[9] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd IEEE International Conference on Data Engineering*, 2006.

[10] Ninghui Li, Tiancheng Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. pages 106–115, April 2007.

[11] Yeye He and Jeff Naughton. Anonymization of set-valued data via top-down, local generalization. In *VLDB '09: Proceedings of the Thirtieth international conference on Very large data bases*. VLDB, 2009.

[12] Mummoorthy Murugesan and Chris Clifton. Providing privacy through plausibly deniable search. In *SIAM International Conference on Data Mining*, Sparks, Nevada, April 30-May 2 2009.

[13] Eytan Adar. User 4xxxxx9: Anonymizing query logs. In *Workshop on Query Log Analysis at WWW*, 2007.

[14] Tiancheng Li and Ninghui Li. On the tradeoff between privacy and utility in data publishing. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009.

[15] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007.

[16] Rosi Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. Vanity fair: Privacy in querylog bundles. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, page 853Ű862, Napa Valley, California, October 26-30 2008.

[17] Rosi Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. "i know what you did last summer": query logs and user privacy. In *Proceedings of the sixteenth ACM conference on information and knowledge management*, pages 909–914, Lisbon, Portugal, November 6-9 2007.

[18] Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, 2007.

[19] Yabo Xu, Benjamin C. M. Fung, Ke Wang, Ada W. C. Fu, and Jian Pei. Publishing sensitive transactions for itemset utility. In *IEEE International Conference on Data Mining*, 2008.

[20] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[21] Ravi Kumar, Jasmine Novak, Bo Pang, and Andrew Tomkins. On anonymizing query logs via token-based hashing. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 629–638, New York, NY, USA, 2007. ACM.

[22] Aleksandra Korolova et al. Releasing search queries and clicks privately. In *WWW*, 2009.

[23] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, page 1, New York, NY, USA, 2006. ACM.

[24] Xiaokui Xiao and Yufei Tao. Anatomy: simple and effective privacy preservation. In *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, pages 139–150. VLDB, 2006.

[25] Christian Borgelt. An implementation of the fp-growth algorithm. In *International Workshop on Open Source Data Mining*, 2005.

[26] Bernard J. Jansen, Amanda Spink, and Jan Pedersen. A temporal comparison of AltaVista web searching: Research articles. *J. Am. Soc. Inf. Sci. Technol.*, 56(6):559–570, 2005.