Privacy through Solidarity: A User-Utility-Preserving Framework to Counter Profiling

Asia J. Biega
Max Planck Institute for Informatics
Saarland Informatics Campus
jbiega@mpi-inf.mpg.de

Rishiraj Saha Roy
Max Planck Institute for Informatics
Saarland Informatics Campus
rishiraj@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Saarland Informatics Campus
weikum@mpi-inf.mpg.de

ABSTRACT

Online service providers gather vast amounts of data to build user profiles. Such profiles improve service quality through personalization, but may also intrude on user privacy and incur discrimination risks. In this work, we propose a framework which leverages solidarity in a large community to scramble user interaction histories. While this is beneficial for anti-profiling, the potential downside is that individual user utility, in terms of the quality of search results or recommendations, may severely degrade. To reconcile privacy and user utility and control their trade-off, we develop quantitative models for these dimensions and effective strategies for assigning user interactions to Mediator Accounts. We demonstrate the viability of our framework by experiments in two different application areas (search and recommender systems), using two large datasets.

KEYWORDS

Anti-Profiling, Privacy, User Utility, Personalization, Mediator Accounts, Profile Scrambling, Search Engines, Recommender Systems

1 INTRODUCTION

Motivation: Users are profiled and targeted in virtually every aspect of their digital lives: when searching, browsing, shopping, or posting on social media. The gathered information is used by service providers to personalize search results, customize ads, provide differential pricing, and more [19, 42]. Since such practices can greatly intrude on an individual's privacy, the goal of our research is to devise a mechanism to *counter* such extensive *profiling*.

A careful user can largely preserve her privacy by taking measures like anonymizing communication or using online services only in a non-linkable manner (for instance, by changing accounts or pseudonyms on a regular basis). However, this comes at the cost of greatly reducing utility, both for the service providers and the user. On the one hand, the service provider will miss out on learning from the same user's long-term behavior, which may result in less effective systems. This issue of system-level utility has been studied in the past research on privacy [20, 23]. On the other hand, the individual user will experience degraded service quality, such as poor search results, as the service provider would not understand the user's interests and intentions. This notion of user-level utility has not been extensively explored in prior work. Our paper formalizes the *trade-off* between a user's *profiling privacy* and her *individual utility*.

State of the art and its limitations: Research in privacy has primarily addressed the disclosure of critical properties in data publishing [5, 7, 13]. Common techniques include coarsening the data so that different users become indistinguishable (e.g., k-anonymity [41], l-diversity [26], and t-closeness [25]), or perturbing the answers of an algorithm so that the absence or presence of any record does not significantly influence the output – the principle of differential privacy [11]. These methods consider notions of utility that reflect a system-level error in an analytical task, such as classification. In contrast, our goal is to prevent detailed profiling and targeting while keeping the *individual user utility* as high as possible, for example, in terms of the quality of personalized search results or product recommendations.

For privacy-preserving search, many approaches have been proposed based on *query obfuscation* [14, 33]. In these solutions, queries are generalized to hide their actual intentions, or additional dummy queries are generated to prevent accurate profiling. Both techniques come at the cost of largely reducing user utility. Similar obfuscation-based techniques have been explored for recommenders [17, 27]. However, none of the prior work addressed the trade-off between privacy and user utility in a quantitative manner. A few methods [8, 33] have considered an entire user community as a means for query obfuscation. This idea is related to our approach in this paper – we generalize it and make it applicable in the context of anti-profiling.

Approach and contribution: Our approach to reconcile privacy and user utility builds on the following observation: service providers often do not need a complete and accurate user profile to return personalized results. Thus, in accordance with the need-to-know principle, we assign user requests to *Mediator Accounts* (MA) mimicking real users, such that (i) individual user profiles are scrambled across MAs to counter profiling, while (ii) coherent fragments of a user's profile are kept intact in the MAs to keep user utility high. We call this paradigm *privacy through solidarity*. Specifically, MAs are constructed by *split-merge assignment* strategies: splitting the interaction history of a user and merging pieces of different users together. Mediator Accounts are meant as an intermediate layer between users and the service provider, so that the provider *only sees MAs* instead of the real users.

Ideas along these lines have been around in the prior literature [15, 34–36, 44], but the formalization of the privacy - user-utility trade-off has never been worked out. In particular, to make

SIGIR '17, August 07-11, 2017, Shinjuku, Tokyo, Japan

DOI: http://dx.doi.org/10.1145/3077136.3080830

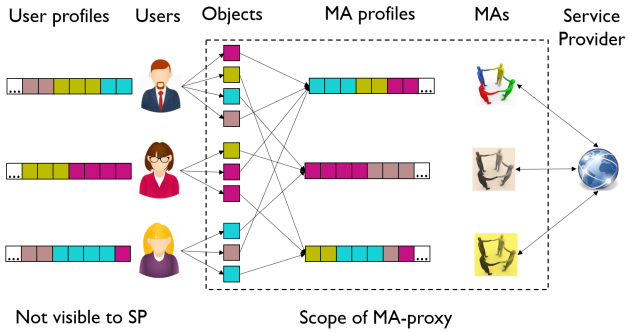

Figure 1: Overview of the MA framework.

this idea viable, one needs to devise quantitative measures for the effects of Mediator Accounts on privacy and utility. In addition, a strategy is needed for assigning user requests to such accounts. The simplest approach of uniform randomization would be ideal for privacy but could prove disastrous for user utility. This paper addresses these challenges within a framework of Mediator Accounts. Our ideas are general enough to be applied to search engines, recommender systems, and other online services where personalization is based on the user interaction history. Our salient **contributions** are:

- a model with measures for quantifying the trade-off between profiling privacy and user utility;
- the Mediator Accounts framework together with strategies for assigning user interactions to MAs;
- comprehensive experiments with two large datasets: search logs derived from the StackExchange Q&A community, and Amazon product ratings.

2 FRAMEWORK OVERVIEW

2.1 Architecture

The architecture of the Mediator Accounts framework is shown in Fig. 1. It consists of three parties: users, a service provider (SP), and a Mediator Accounts proxy (MA-proxy). A *user profile* consists of a set of *objects*, such as queries, product ratings or other forms of user interactions with the SP. Instead of issuing objects directly to the SP, users pass them on to the MA-proxy together with some context information. The goal of the MA-proxy is to redistribute the incoming objects on to mediator profiles mimicking real users. The MA-proxy assigns each incoming object to a Mediator Account offering the right context for the current object and user, and issues the object to the SP from the chosen MA. Upon receiving a response (for example, a result page or a product recommendation) from the SP, the MA-proxy passes it back to the user. When an interaction is over, the MA-proxy discards all linking information about the original user and the object and remembers only the association between the mediator account and the object. As a result, the original user profiles are scrambled across multiple MAs, and each MA consists of data from multiple users.

2.2 Incentives of participating parties

Users. The goal of a user participating in an MA system is to be able to get high-quality personalized results, while not letting any online provider (neither SPs nor the MA-proxy) keep her interaction history and link it to her as an individual. The MA-proxy has the user interaction history scrambled across multiple accounts, and no links between the objects and the real users are stored.

Users of anonymous services that do not offer topical personalization, such as the DuckDuckGo, Startpage or Qwant search engines, may be open to trading off some privacy for enhanced results through the MA framework.

Non-profiling service providers. The incentive of a non-profiling service provider would be to enhance personalization in the results, without compromising on the non-profiling principle.

Profiling service providers. A big question is whether profiling service providers would allow a third-party like an MA-proxy to mediate between them and the users. While examples of such third-parties already exist (the Startpage search engine uses Google as a source of search results), we believe that (i) an MA-proxy being able to group objects into realistic profiles that yield similar analytics results for the SP, and (ii) an MA-proxy being able to attract privacy-wary users who would not otherwise use the profiling SP, would be viable incentives for an SP not to block an MA service.

MA-proxy. An MA-proxy could be set up by individuals, or cooperatives of non-profiling SPs (to provide personalization without accumulating real user profiles), or by non-governmental organizations that promote online privacy. The Electronic Frontier Foundation is such an organization – a non-profit organization that has built privacy-preserving solutions like Privacy Badger.

2.3 Trusted and adversarial parties

MA-proxy. Users opting for an MA service would need to trust that it scrambles their profiles across mediator accounts, and discards the original profiles as well as any identifying information once an interaction (a single request or a session) is complete. A standard approach to gain such trust would be to make the MA solution open-source, enabling the code to be vetted by the community. A real implementation of an MA framework would have to take into account secure end-to-end communication channels between users and SPs via the MA-proxy. These issues may be resolved using encryption and security techniques (e.g., secure browser, onion routing, etc.), and are outside the scope of this paper.

Provider. The service provider is not exactly distrusted, but there have been cases where user-related information has been leaked or passed on beyond the original intentions – by sabotage, acquisition by other companies, or enforcement by government agencies. By detaching users from profiles and limiting their accuracy, the potential damage is bounded.

Other risks might result from service providers displaying privacy-sensitive personalized ads, such as ads related to pregnancy or health issues, especially when observed by others on a user's screen. The architecture would allow an MA-proxy to support filtering ads and adjusting them to users' topical interest. Such a configuration has indeed been found to be a preferable ad-serving setup in a user study [1]. Ad filtering, however, is orthogonal to this research.

Third parties. Profiling companies that operate outside the user-provider connections are considered untrusted. The same holds for agglomerates of providers that aggregate and exchange user data. A conceivable attack could be to guess a user's attribute (e.g., whether she is pregnant) by combining (i) observations on the MAs and (ii) observations on a set of accounts in a social network, using statistical inference methods. The MA framework aims to keep such risks low by breaking observable associations between MAs and real users, and limiting the profiling accuracy of the split-merge superpositions of different users that cannot be easily disentangled.

3 ASSIGNMENT MODEL

The core of the MA framework is an algorithm for assigning user objects to Mediator Accounts. To guide it on the privacy-utility trade-offs and to assess the quality of the output, we need measures for quantifying the effect of an assignment on privacy and user utility. This section presents such measures, and the algorithm for object assignment based on the split-merge concept.

3.1 Concepts and notation

We use the following notations:

- A set U of users $u_1 \dots u_p$.
- A set O of objects $o_1 \dots o_s$ issued by users; the objects are treated as unique even if they represent the same content. For instance, a query `folk music` issued by user u_i is treated as an object distinct from the same query issued by u_j. Analogously, a product rating for a book (`Folk Music History`, 3.0) by u_i is distinct from the rating by u_j for the same book, irrespective of the rating value.
- A set M of mediator accounts $m_1 \dots m_t$ to which objects are assigned by the MA-proxy.

We reserve the symbols i, j, k for subscripts of users, objects, and MAs. If user u_i issues object o_j, we write $o_j \in u_i$. Similarly, if o_j is assigned to MA m_k, we write $o_j \in m_k$.

Assignments. An *assignment* of objects on to MAs can be denoted as an $s \times t$ matrix A of 0-1 values, where $A_{ij} = 1$ means that o_i is assigned to m_j. If we think of the Cartesian product $O \times M$ as a bipartite graph, then the assignment can be conceptualized as a subgraph $S \subseteq O \times M$ where each node of type O has exactly one edge with one of the M nodes.

3.2 Objective

In a real application, an MA-proxy has to assign objects to accounts in an online manner, one object at a time as input arrives. In this paper, we focus on analyzing the model and assignments in an offline setting, although the algorithm we devise can be applied in both offline and online scenarios. The offline case is useful for two reasons. First, it is a foundation for understanding the underlying privacy-utility trade-offs. Second, performing offline assignment on a set of initial user profiles can address the cold-start problem that a new MA-proxy would face. Using the notation from Sec. 3.1, the *MA offline assignment problem* can be defined as follows:

- Given a set of objects O belonging to a set of users U, and the set of mediator accounts M, compute an assignment matrix A that optimizes a desired objective function for the privacy-utility trade-off.

The *MA online assignment problem* is:

- Given an assignment A of past user objects to MAs and a newly arriving object o of user u, find the best MA to which o should be assigned with regard to a desired goal for the privacy-utility trade-off.

3.3 Measuring privacy gain

An ideal situation from the perspective of privacy is when the objects from a user profile are spread across MAs uniformly at random – this minimizes the object-level similarity of any MA to the original profile. We thus measure privacy as the entropy of the user distribution over MAs, formalizing these notions as follows.

Entropy. We introduce for each user u_i an *MA-per-user vector* $\vec{mu}_i \in (\mathcal{N}_0)^t$ with one counter (≥ 0) per MA, written as $\vec{mu}_i = \langle x_{i1} \dots x_{ij} \dots x_{it} \rangle$ where x_{ij} is the number of objects by user u_i in account m_j (such that $\sum_{j=1}^{t} x_{ij} = |u_i|$). We can cast this into an *MA-per-user probability distribution* $\mathbf{\Phi_i} = \langle \phi_{i1} \dots \phi_{ij} \dots \phi_{it} \rangle$ by setting $\phi_{ij} = x_{ij}/|u_i|$ followed by smoothing (e.g., Laplace smoothing) so that $\phi_{ij} > 0$ for each j and $\sum_{j=1}^{t} \phi_{ij} = 1$.

The degree of u_i's profile fragmentation can be captured by the entropy of the distribution $\mathbf{\Phi_i}$. We can define the *MA-per-user entropy* as a measure of *privacy gain* (gain over having each user exhibit her full individual profile):

$$privacy{-}gain(u_i) = H_i = -\sum_{j=1}^{t} \phi_{ij} \log \phi_{ij} \tag{1}$$

This quantifies the spread of the user's objects across accounts. The higher the entropy value, the higher the gain in profiling privacy.

Profile overlap. If a use-case requires a more user-interpretable measure of privacy, an alternative is to minimize the *maximum profile overlap*. For a user u_i, this measure can be expressed as:

$$O_i = \max_{j=1}^{t} \frac{|\{o \in u_i \cap m_j\}|}{|u_i|} \tag{2}$$

This measure of overlap can directly tell a user how much "error" could be made by an adversary, who assumes one of the MAs is the user's profile. The optimum for this measure, as with entropy, is achieved when the objects are uniformly spread across accounts. Thus, in the following, we use entropy as our privacy measure, and leave maximum profile overlap as a design alternative.

3.4 Measuring user utility loss

User utility loss measures to what extent an object o_k of user u_i is placed *out of context* by mapping it to account m_j. We define a real-valued function $sim(\cdot,\cdot)$ to measure the coherence of user and MA profiles: $sim(o_i, o_j) \in [0,1]$ is a symmetric measure of the relatedness between objects represented by o_i and o_j. In practice, different notions of relatedness can be used, based on object properties or usage. In settings where labels for topics or categories are available, we can set $sim(o_i, o_j) = 1$ if o_i and o_j are issued by the same user and have the same topic/category label, and 0 otherwise. Generally, we assume that sim measures are normalized with values between 0 and 1.

The objects of user u_i form a *context*, typically with high pairwise relatedness among the objects. When considering sets of objects as a whole (rather than time-ordered sequences of object posts), we can measure the *normalized context coherence* of an object o_k in the profile of user u_i by:

$$coh(o_k, u_i) = \frac{\sum_{o_l \in u_i, k \neq l} sim(o_k, o_l)}{|u_i| - 1} \quad (3)$$

When o_k is placed in MA m_j, we analogously define:

$$coh(o_k, m_j) = \frac{\sum_{o_l \in m_j, k \neq l} sim(o_k, o_l)}{|m_j| - 1} \quad (4)$$

The utility loss of u_i in a given MA assignment is then measured as an average coherence loss over all user objects:

$$utility{-}loss(u_i) = \frac{\sum_{o_k \in u_i} [coh(o_k, u_i) - coh(o_k, m_j)]}{|u_i|} \quad (5)$$

where m_j is the account containing o_k in the given assignment.

The normalization helps to account for varying sizes of user profiles. As a result, coherence values are always between 0 and 1, and utility loss is normalized to take values between -1 and 1. Note that our utility measure assumes that the context coherence can increase if an object is assigned to an MA with more similar objects. Coherence increase will result in negative utility loss.

3.5 Assignment algorithms

The role of an assignment algorithm is to scramble user objects across accounts so as to satisfy a desired privacy-utility tradeoff or optimize a corresponding objective function. In this paper, we experiment with a number of assignment algorithms and study their output quality.

3.5.1 Optimal assignment (Offline). The trade-off can be expressed as a joint non-linear optimization problem as follows:

$$\max_{A} \min_{u} [\alpha \cdot privacy\_gain(u) - (1 - \alpha) \cdot utility\_loss(u)] \quad (6)$$

Alternatively, one could optimize one of the two measures with a constraint on the other. Solving this problem exactly, however, is computationally expensive. If we use the less complex overlap privacy measure, we could cast the problem into a Quadratic Integer Program. However, this would have millions ($|M| \cdot |O|$) of variables; so it would remain intractable in practice. We thus do not pursue this direction in this paper and instead consider a number of heuristics. The following are also suitable for the *online case*.

3.5.2 Profiling-tradeoff assignment. We aim to approximate the combined objective function as follows. Let o be an object we want to assign to one of the accounts m_j. If we want to optimize for privacy (i.e., entropy), we should choose an MA at random from a uniform distribution over MAs:

$$P_{priv}(m_j|o) = \frac{1}{|M|} \quad (7)$$

If we want to optimize for utility, we could choose an MA that offers the best coherence:

$$P_{util}(m_j|o) = \begin{cases} 1, & \text{if } m_j = m_{max} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $m_{max} = \arg\max_{m_k} coh(o, m_k)$.

Let α be a parameter that controls the trade-off between privacy and utility. We sample an MA according to the distribution:

$$P(m_j|o) = \alpha \cdot P_{priv}(m_j|o) + (1 - \alpha) \cdot P_{util}(m_j|o) \quad (9)$$

In the offline case, we may choose an arbitrary order of objects to feed into this assignment heuristic. In the online case, we process objects ordered by the timestamps in which they are issued to the MA-proxy. It is also worth noting that in an online setting users could choose different α for each object, deciding that some should be assigned randomly, and some with the best possible context.

3.5.3 Random assignment. In this assignment, objects are assigned to accounts uniformly at random. This is a special case of the Profiling-tradeoff algorithm with $\alpha = 1$. This assignment maximizes privacy.

3.5.4 Coherent assignment. Personalization is usually based on semantically coherent parts of user profiles. If we retain such coherent fragments of a profile within the accounts, individual utility should be preserved better than in a completely random assignment. The mode in which we assign an object to the account that offers the best coherence is a special case of the Profiling-tradeoff algorithm, in which we set $\alpha = 0$. We refer to this method as Coherent. This assignment explicitly aims for the best utility only, yet some privacy is gained as chunks of user profiles get assigned to MAs randomly.

4 MA IN SEARCH SYSTEMS

By analyzing query-and-click logs, search engines can customize results to individual users. Such user profiling, however, may reveal a detailed picture of a person's life, posing potential privacy risks. At the same time, personalization of a single query is often based on only a subset of a user's history. Thus, as a first use case, we apply the MA framework in a search engine setting, scrambling the query histories of different users across accounts.

4.1 Framework elements

In the search scenario, the elements of the framework described in Sec. 3 are instantiated as follows. The objects are *keyword queries*, and user profiles consist of sets (or sequences) of queries, possibly with timestamps. Accounts contain re-assigned queries of different users. Object similarity can be understood as topical similarity between queries, with topics being either explicit such as categories or classifier labels, or latent, based on embeddings. As a query is characterized by a set (or weight vector) of topics, the similarity can be computed, for instance, using (weighted) Jaccard overlap or vector cosine. The service provider in this setting is a search engine, which, upon receiving a query from a given user profile, returns a ranked list of documents personalized for that user. User utility is measured by the quality of the result list.

4.2 Service provider model

The ability of the MA framework to preserve utility while splitting user profiles across accounts depends on a retrieval model for ranking query answers. We use the language-model-based retrieval technique [10], as described below.

Let $o \in u$ be a query of user u consisting of a number of words $w \in o$, and D be the document collection. The model retrieves the results in two steps. First, it fetches a set of top-k documents $D_o \subseteq D$, each document $d \in D$ being scored by the query-likelihood model with Dirichlet smoothing (parameter μ_D [10]):

$$score(o,d) = \log P(o|d) = \sum_{w \in o} \log \left(\frac{tf_{w,d} + \mu_D \cdot P(w|D)}{|V_d| + \mu_D} \right) \quad (10)$$

where $tf_{w,d}$ is the count of w in d, $P(w|D)$ is the probability that w occurs in D, and $|V_d|$ is the count of all words in d. For every user u, we compute a personalization score as the log-probability of the document d being generated from the user language model using Dirichlet smoothing with parameter μ_U, where U is the set of all users (or equivalently, the collection of their search histories):

$$score(d,u) = \log P(d|u) = \sum_{w \in d} \log \left(\frac{tf_{w,u} + \mu_U \cdot P(w|d)}{|V_u| + \mu_U} \right) \quad (11)$$

where $tf_{w,u}$ is the count of w in the search history of u, $P(w|d)$ is the probability that w occurs in d, and $|V_u|$ is the count of all words in the search history of u.

In the second step, documents $d \in D_o$ are re-ranked using a linear combination of the two scores:

$$score_u(o,d) = \gamma \cdot score(o,d) + (1-\gamma) \cdot score(d,u) \quad (12)$$

In practice, γ would be set to a low value to put more importance on personalization.

When we use the MA framework, the computations are similar. The notion of a user is simply replaced by an account m. The personalization stage is adjusted as follows: we compute $score(d,m)$ using $P(d|m)$, which in turn is computed using $tf_{w,m}$, μ_M and $|V_m|$ with Eq. 11. Definitions of these quantities are analogous to their user counterparts.

5 MA IN RECOMMENDER SYSTEMS

Recommendation platforms like online shops, movie review forums, and music streaming sites, aggregate user interaction histories over time. As in the search setting, there is a direct correlation between the accuracy and completeness of user profiles and the quality of the obtained recommendations. Thus, as a second use case, we apply the MA framework to recommender systems.

5.1 Framework elements

Users of a recommendation platform rate different *items* like movies, books, and hotels. These *item-rating pairs* are the *objects* in the *profile* of a user. An item belongs to a set of *categories* (e.g., movie genres) from a taxonomy defined by the service provider. The *object similarity function* can be defined by topical similarity using such categories or tags. Additionally or alternatively, similarity can consider the ownership of objects, that is, whether two ratings are by the same user or different ones.

User utility here refers to the quality of recommendations. Rating predictions for items unseen by a user are made based on the past ratings of that user, as well as ratings of similar items by similar users. Thus, scrambling a user's ratings across accounts, if not done in a principled fashion, can potentially destroy user-item preference patterns, and hence degrade the quality of rating prediction.

5.2 Service provider model

A widely deployed rating prediction algorithm is *collaborative filtering* (CF) [22], which has been used by providers like Netflix, Amazon and Google [49]. In this model, the values of user-item ratings are stored as a matrix, where rows represent users, and columns represent items. CF maps both users and items into a low dimensional (latent) space using matrix factorization, such that user-item interactions (ratings) are modeled as inner products in that space, $\hat{r}(u_i,o_k) = \boldsymbol{q_{o_k}}^T \boldsymbol{p_{u_i}}$, where $\boldsymbol{p_{u_i}} \in \mathbb{R}^f$ and $\boldsymbol{q_{o_k}} \in \mathbb{R}^f$ are vector representations of the user u_i and item o_k in the latent space with dimensionality f. CF avoids overfitting by adding a regularization term to the objective function of the matrix factorization. To learn the low-dimensional vectors ($\boldsymbol{p_{u_i}}$ and $\boldsymbol{q_{o_k}}$), the system minimizes the regularized squared error on the set of known ratings (Eq. 13) [22]:

$$\min_{q^*,p^*} \sum_{(u_i,o_k) \in R} (r(u_i,o_k) - \boldsymbol{q_{o_k}}^T \boldsymbol{p_{u_i}})^2 + \lambda(||\boldsymbol{q_{o_k}}||^2 + ||\boldsymbol{p_{u_i}}||^2) \quad (13)$$

where R is the set of (u_i,o_k) pairs for which $r(u_i,o_k)$ is known (the training set of gold ratings), and λ is the regularization parameter. The trained model can then be used to predict ratings for unseen (user, item) pairs using the expression for $\hat{r}(u_i,o_k)$.

In the MA framework, the service provider no longer predicts the ratings for users, but for mediator accounts. Specifically, it learns a CF model from the MA-ratings data (triples of MA-id, item-id, rating), and then makes predictions for the unseen (m_j,o_k) pairs. As the ratings of an individual user are spread across different MAs, we need a method for *propagating the predicted ratings* from the MAs back to the individual users. Assume that we need to predict the rating $\hat{r}(u_i,o_k)$ when the MAs are present. Under a given assignment, u_i's items are split across MAs $\{m_j \in M\}$ with a distribution $\boldsymbol{\Phi_i}$ (Sec. 3.3), where the fraction of u_i's mass in some m_j is given by ϕ_{ij}. We assume that the MA-proxy has access to the predicted ratings $\hat{r}(m_j,o_k)$ from the service provider. We compute the propagated rating $\hat{r}^m(u_i,o_k)$ as the weighted sum of $\hat{r}(m_j,o_k)$:

$$\hat{r}^m(u_i,o_k) = \sum_{j=1}^{|M|} \phi_{ij} \hat{r}(m_j,o_k) \quad (14)$$

6 EXPERIMENTS ON SEARCH

6.1 Experimental setup

6.1.1 Dataset. For lack of publicly available query logs with user profiles, we created a query log and a document collection using the data from the Stack Exchange Q&A community (dump as of 13-06-2016). We excluded the large software subforums from outside the Stack Exchange web domain (such as StackOverflow), as they would dominate and drastically reduce the topical diversity. The final dataset consists of ca. $6M$ posts of type 'Question' or 'Answer' in 142 diverse subforums (e.g., Astronomy, Security, Christianity, Politics, Parenting, and Travel).

Document collection. We use all posts of type 'Answer' as our collection. The resulting corpus contains $3.9M$ documents.

User query histories. We construct a query log from posts of type 'Question', as these reflect users' information needs. Each question is cast into a keyword query selecting the top-l question

words with the highest TF-IDF scores, where l is a random integer between 1 and 5. We consider only users with at least 150 questions, which yields a total of 975 users and $253K$ queries. Each query is assigned a topical label, used for object similarity. We set this label to the *subforum* where the original question was posted.

6.1.2 Service provider. For reproducible experiments, we base our search engine model on the open-source IR system Indri [40]. Indri ranks query answers based on state-of-the-art statistical language models with Dirichlet smoothing [10]. We use Indri to retrieve the top-100 results for every query from the entire corpus, and implement user-personalized re-ranking ourselves (see Sec. 4.2). We compute per-user language models from the original questions to tackle sparsity. The Dirichlet smoothing parameter is set to the average document length (56 words), and γ is set to 0.1.

6.1.3 Empirical measures.

Privacy Gain. The model entropy reflects how scrambled the user profiles are. Yet from the perspective of a profiling adversary it is rather the distribution over semantic topics that matters. Empirically, a proper way to measure privacy then is to compare the original topic distribution per user against the topic distributions of the MAs. The minimum KL-divergence between pairs of these distributions signifies the privacy level:

$$emp{-}priv{-}gain(u_i) = \min_{m_j \in M} D_{KL}(P^{u_i} \parallel Q^{m_j}) \quad (15)$$

where P^{u_i} and Q^{m_j} refer to the user and MA profile distributions over topics with add-one Laplace smoothing. We use subforums as explicit labels for topics.

Utility Loss. Rankings of documents d for a query are derived from $score_u(o,d)$ and $score_m(o,d)$ (Eq. 12), respectively, where the former refers to the query being issued by user u and the latter to the query being issued by the mediator account m (see Sec. 4.2). We quantify the empirical utility loss as the divergence between the two rankings. We compute two measures: the loss in Kendall's Tau over the top-100 document rankings: $1 - KTau@100$ (as the personalization step considers the top-100 documents), and the loss in Jaccard similarity coefficient over the first 20 ranking positions: $1 - Jaccard@20$ (as end-users typically care only about a short prefix of ranked results). For each user, we average these scores over all queries.

6.1.4 Assignment methods.

Object similarity. We set $sim(o_i, o_j) = 1$ if both o_i and o_j belong to the same user and to the same topic, and 0 otherwise. During the assignment, this measure helps to keep related parts of a user profile together.

Assignment algorithms. We run the Profiling-Tradeoff algorithm varying α between 0 and 1 with a 0.1 increment, and setting the number of MAs to be the number of users (975). With the chosen object similarity, the special case of $\alpha = 0$, i.e. the Coherent assignment, results in splitting user profiles into subforum chunks and assigning each chunk to a randomly chosen account.

6.2 Results and insights

Aggregate trends. Table 1 presents the results on the model measures and empirical measures for different values of the assignment trade-off parameter, macro-averaged over users. Recall that $\alpha = 0.0$ and $\alpha = 1.0$ correspond to the special cases of Coherent and Random assignments, respectively. These results need to be contrasted with the baseline, denoted *Original* in the table, where each original user forms exactly one account (i.e., no scrambling at all). Compared to the baseline, all numbers are statistically significant by

Table 1: Search results with trade-off parameter α for the model (M) and empirical (E) measures.

α	M-Priv-Gain (Entropy)	M-Util-Loss (Coherence Loss)	E-Priv-Gain (Min. KL-div.)	E-Util-Loss (1 - KTau@100)
Original	0.000	0.000	0.000	0.000
0.0 (Coh)	1.180	0.178	0.320	0.170
0.2	2.208	0.293	0.319	0.203
0.4	3.130	0.389	0.346	0.228
0.6	3.975	0.463	0.389	0.246
0.8	4.731	0.515	0.494	0.260
1.0 (Rand)	5.287	0.535	0.863	0.266

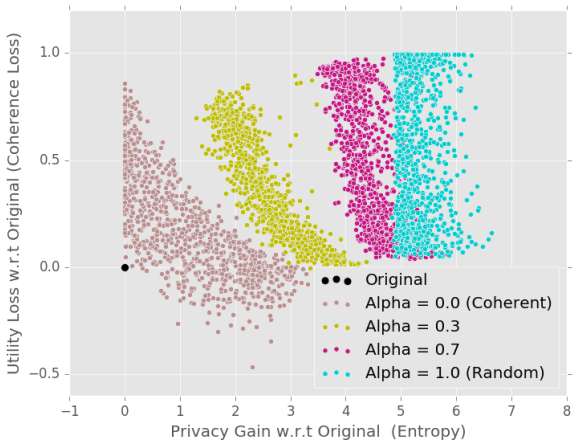

Figure 2: Model measures per user (search).

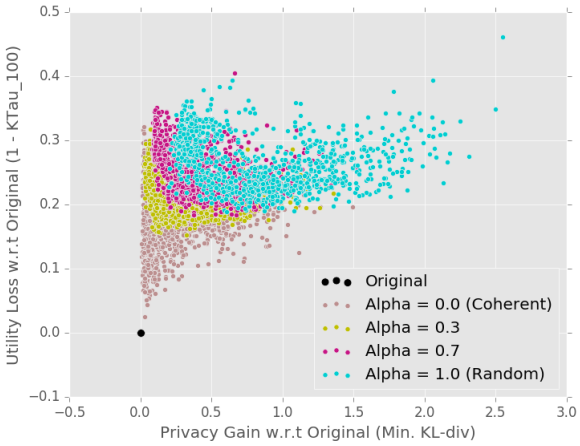

Figure 3: Empirical measures per user (search).

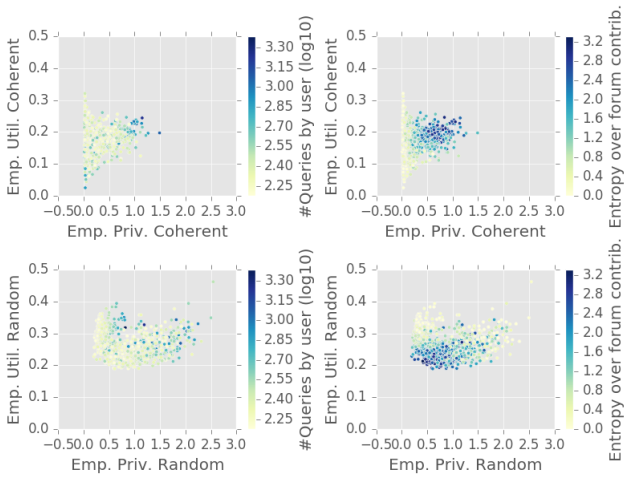

Figure 4: Effect of profile size and diversity (search).

paired t-tests with $p < 0.01$. For empirical utility loss, we report Kendall's Tau; the results for the Jaccard coefficient are similar.

The results show that the Profiling-Tradeoff assignments improve privacy over the *Original* baseline (the topical KL-div. between original users and MAs is increased) while keeping the utility loss low. This is largely true regardless of the exact choice of α. So the MA framework provides a fairly robust solution to reconciling privacy and utility, supporting the observation that high-quality topical personalization does not require complete user profiles.

With the α increasing, assignments become more random, so the privacy increases and utility is reduced (but with a low gradient). In this regard, the empirical measures reflect the expected behavior according to the model measures well.

Results per user. Figs. 2 and 3 show privacy and utility values of each user, for the model and empirical measures, respectively. Different colors represent different assignments, and each dot represents a user, with measures averaged over the user's queries. We have several observations:

- Higher privacy gain is correlated with higher utility loss. The Original assignment maps each user to the origin (0 utility loss, but also 0 privacy gain). No assignment reaches the bottom-right area of the chart – which would be an ideal.
- Varying α not only tunes the privacy-utility tradeoff at the community aggregate level, but also affects the variance over individual user scores. This suggests that we should further explore choosing α on an individual per-user basis (which is easily feasible in our framework, but is not studied in this paper).
- Even the Random assignment ($\alpha = 1.0$) keeps utility reasonably high. This is due to the fact that random MAs – sampled from queries in the community – end up being averaged rather than random profiles.
- Some users achieve high privacy gains without losing hardly any utility, and vice versa. We investigate this further below.

Effect of profile size and diversity. We analyze how different user profile characteristics affect the assignment results. Figure 4 presents the empirical trade-offs for the Coherent (top row) and Random (bottom row) assignments, where each dot is a user and the dot color represents (i) the logarithm of the number of queries in the user profile (left column), or (ii) the diversity of the profile measured by the entropy of the distribution of queries across topics (right column). We make the following observations:

- Users with more queries (darker dots) in the Coherent assignment clearly gain privacy at the cost of losing utility, whereas for the smaller profiles (lighter dots), the trade-off is not as pronounced. In the Random assignment this trade-off is less pronounced irrespective of the size of the profile.
- In the right column, one can see the lighter dots (profiles with little diversity) moving from the bottom-left for the Coherent assignment (little privacy gain, little utility loss) to the top-right for the Random assignment (higher privacy gain, higher utility loss). This suggests that our framework does not offer much help to the users with uniform and focused interests. This is an inherent limitation, regardless of which privacy protection is chosen. Such homogeneous users cannot hide their specific interests, unless they give up on personalization utility.
- Our split-merge assignments offer good results for users with high diversity. As suggested by the darker dots, the Coherent assignment leads to a lower utility loss and higher privacy gain for users with diverse profiles, when compared to the Random assignment. This is because such users have more independent and internally coherent chunks that can be split without affecting utility. This class of users is exactly where the right balance of utility and privacy matters most, and where we can indeed reconcile the two dimensions to a fair degree.

7 EXPERIMENTS ON RECOMMENDERS

7.1 Experimental setup

7.1.1 Dataset. We use the Amazon product rating data collected by Mukherjee et al. [30] which contains user-item ratings, along with review text and metadata. We extract (user, item, rating) triples from this data with associated timestamps, and restrict product type to `music`, one of the most frequent product types. We identify active users who rated between 50 and 1000 items. Within `music` there are 22 categories, as defined by Amazon: `rock`, `jazz`, `country`, etc. An item is originally associated with 2.12 categories on average, but we assign it uniquely to *one* category by selecting the most frequent category. The final data had 1,719 users, 72,464 items and 197,215 ratings (ratings are between 1 and 5).

7.1.2 Service provider. We used a parallelized implementation of a collaborative filtering model based on matrix factorization: ALS-WR [22, 49]. This is available in Apache Spark (https://goo.gl/X33DN9, Accessed 23 Jan 2017), and widely used. The model has three parameters – number of latent features f, the regularization parameter λ, and the number of iterations n_i to run. f was set to 22, which is the number of categories in the data. λ and n_i were learnt to be 0.3 and 10 respectively, using grid search on a separate development set of $60K$ ratings.

Table 2: Recommender results with parameter α for the model (M) and empirical (E) measures.

α	M-Priv-Gain (Entropy)	M-Util-Loss (Coh. Loss)	E-Priv-Gain (Min. KL-Div.)	E-Util-Loss (MSE)
Original	0.000	0.000	0.000	0.000
0.0 (Coh)	0.826	0.094	0.073	0.501
0.2	1.778	0.086	0.064	0.576
0.4	2.573	0.106	0.075	0.623
0.6	3.339	0.122	0.093	0.664
0.8	4.009	0.133	0.130	0.689
1.0 (Rand)	4.495	0.137	0.262	0.690

7.1.3 Empirical measures.

Privacy Gain. We use the same measure of per-user empirical privacy as in the search scenario, which is the KL-divergence between the topical distributions representing the user profile and the "closest" MA profile (Eq. 15). The distributions are computed over the 22 Amazon Music categories (with add-one Laplace smoothing).

Utility Loss. Empirical utility for user u_i is measured in terms of mean squared error (MSE). MSE is computed for $R_{u_i}^{test}$, where the system needs to generate recommendations for each of 40 given items, for each user u_i. These test cases are not part of the training data. The error is computed between predictions made by the user model and the model propagated through the MAs, i.e., $\hat{r}(u_i, o_k)$ and $\hat{r}^m(u_i, o_k)$ (Sec. 5.2). We thus have in Eq. 16:

$$emp\text{–}util\text{–}loss(u_i) = \frac{\sum_{y=1}^{|R_{u_i}^{test}|}(\hat{r_y}(u_i, o_k) - \hat{r_y}^m(u_i, o_k))^2}{|R_{u_i}^{test}|} \quad (16)$$

The set of 40 unseen items for a user to make predictions on is picked at random from the set of 72,464 items. Our complete test set thus contained 68,760 (= 1,719 × 40) (user, item) pairs.

Under the MA model, each user-item rating prediction requires multiple MA-item predictions. For the assignment methods presented, we had to make up to $7.5M$ predictions in total from the MA model, for the 68,760 test ratings. However, with the parallelized Spark implementation, this took only a few minutes on a single machine with 8 GB RAM.

7.1.4 Assignment methods. We create 1,719 MA profiles, equal to the number of users. The object similarity is set to 1 if two objects belong to the same user and the same category, and 0 otherwise. The tradeoff parameter α was varied from 0.0 through 1.0 in steps of 0.1. Rating timestamps were used to determine the order of object assignments.

7.2 Results and insights

Aggregate trends. Table 2 shows how the model and empirical measures of privacy and utility vary with tradeoff parameter α. Each value is an average over all 1719 users. α = 0.0 and 1.0 correspond to Coherent and Random assignments respectively. *Original* again denotes the baseline where each original user forms her own account. Note that *higher values for the privacy measures* of entropy and KL-divergence indicate better privacy gain, and *lower values for the utility measures* of coherence loss and MSE imply better user utility loss.

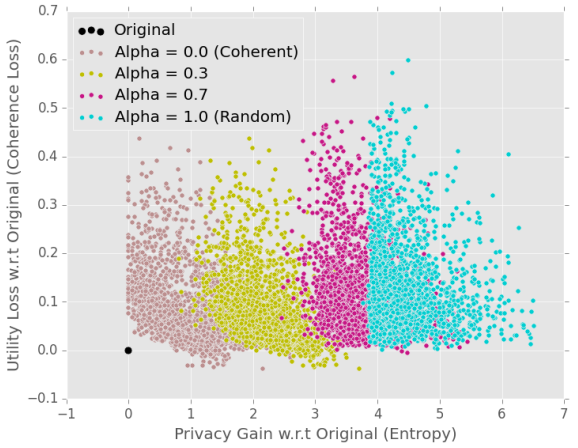

Figure 5: Model measures per user (recommenders).

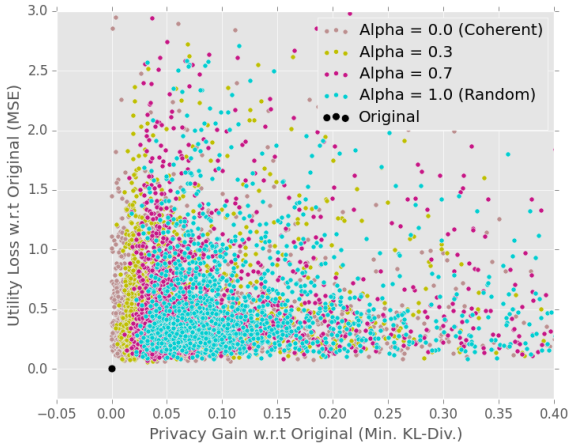

Figure 6: Empirical measures per user (recommenders).

The major insight here is the following. For the *Coherent assignment*, we observe good empirical utility (a very low MSE of 0.501), while providing a substantial improvement in empirical privacy (0.073 from 0). Thus, this assignment is a good candidate for practical deployment. Generally, the Profiling-Tradeoff assignment works well for all α, demonstrating the robustness of our approach.

The trends in empirical privacy and utility mimic those of the model measures. As α is increased from 0.0 to 1.0, there is gradual improvement in empirical privacy and a monotonic degradation in user utility, both with low gradients.

Results per user. Figs. 5 and 6 visualize privacy versus utility for model and empirical measures, where each point denotes an individual user. Different colors represent different tradeoff settings, as shown in the legend. There are two notable observations:

- Users form different clusters in the privacy-utility space as α is varied. The clusters are softer (more overlapping) in the empirical case than in the model. This shows that analytically

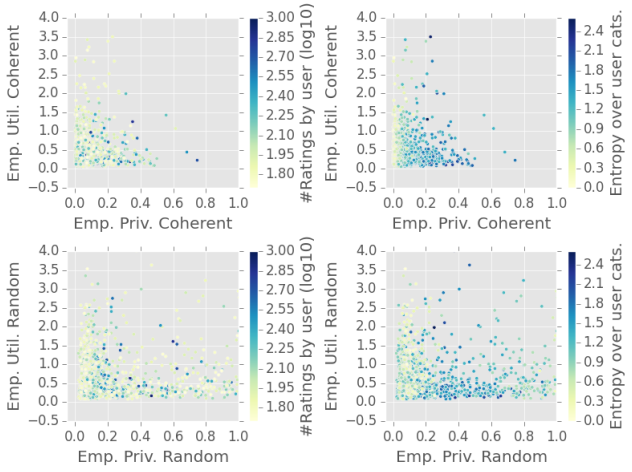

Figure 7: Effect of profile size and diversity (recommenders).

predicting these tradeoff statistics is not easy; hence our experimental approach.

- Fig. 6 indicates that, for the Coherent assignment, a fraction of users have both *high privacy gain* and *low utility loss* (the brown dots in the lower right area of the chart), while most users exhibit the expected privacy-utility tradeoff. This suggests further optimizations by tuning assignment parameters for each individual user separately.

Effect of profile size and diversity. As in the search setting, we report on the variation of the empirical measures with respect to changes in user profile size (number of items rated) and diversity (entropy of the distribution of item categories in the user profile) in Fig. 7. The two plots on the left refer to size and the ones on the right to diversity. The upper and lower plots correspond to the Coherent and Random assignments, respectively. We observe:

- Users with diverse interests do well on both dimensions (more dark dots towards lower right), and thus benefit the most from our framework. This represents a significant fraction of all users, as seen by the numerous dark dots in the diversity graphs.
- A higher profile size yields lower utility loss, for the Coherent assignment. So while a bigger profile typically entails better prediction accuracy (our measure of empirical utility), we can see that this is true only if related parts of the user profile are preserved in the MAs. This comes at the cost of somewhat reduced privacy gains. However, the Coherent assignment does fairly well in both dimensions. For Random assignment, profile size does not have a notable effect on privacy, and utility is often completely lost.
- Overall, users with larger profiles *and* varied interests have the best chance of preserving utility while having substantial privacy gains under the Coherent assignment. Thus, we can derive guidelines for tuning α (Table 2): α may be kept high in the beginning for objects of all users, and it may gradually be decreased when users show diversity in their interests as their profiles grow bigger (cf. Sec. 3.5.2).

8 RELATED WORK

Grouping for privacy: The idea of masking the traces of individual users by combining them into groups has been around since the *Crowds* proposal by [35]. However, this early work solely focused on anonymity of web-server requests. [31] devised an abstract framework for group privacy over obfuscated databases, but did not address utility. For search engines specifically, [21] proposed a notion of query bundles as an implicit grouping of users, but focused on countering de-anonymization in the presence of so-called vanity queries. The short paper [50] sketches a preliminary approach where semantically similar queries by different users are grouped for enhancing privacy. Aggregation of users' website-specific privacy preferences through a centralized server [47], can also be perceived as a type of *privacy through solidarity*. The principle of solidarity has moreover been explored through a game-theoretic framework over recommender systems [18].

Tracking and profiling: A good body of work investigates to what extent and how users are tracked by third parties in web browsers [24, 29, 47], or through mobile apps [28]. These are primarily empirical studies with an emphasis on identifying the tracking mechanisms. The interactions with service providers, where users log in and leave extensive traces, have been largely disregarded. In contrast, our framework helps counter both tracking and individual profiling by detaching users from online accounts.

To reduce the scale of profiling, a model called stochastic privacy has been proposed to selectively sample user profiles for use by personalizing algorithms [39]. To counter profiling by search engines in particular, [45] has proposed to issue queries anonymously, but provide the engine with a coarse topical profile for answer quality. On the tracking front, the Non-Tracking Web Analytics system reconciles users' need of privacy and online providers' need of accurate analytics [3]. Although these various works address the privacy-utility trade-off, no explicit control mechanism has been proposed for user utility.

Privacy-preserving IR: The intersection of privacy and IR has received some attention in the past years [46]. One of the key problems studied in the field is that of post-hoc log sanitization for data publishing [9, 16, 48]. Online sanitization, on the other hand, aims at proactively perturbing and blurring user profiles. Techniques along these lines typically include query broadening or dummy query generation (e.g., [4, 33, 37, 43]). It has also been proposed to perturb user profiles by making users swap queries and execute them on behalf of each other [34]. Very few of these prior works consider the adverse impact that obfuscation has on utility, and the usual focus is on the utility of single query results. To the best of our knowledge, none of them focuses on personalization utility or offers quantitative measures for the trade-off.

Another privacy concept studied in IR is that of exposure. Recently, the notions of R-Susceptibility and topical sensitivity have been proposed to quantify user exposure in sensitive contexts within a given community [6].

Privacy-preserving data mining: There is a vast body of literature on preserving privacy in mining data for rules and patterns and learning classifiers and clustering models [2, 12]. In this context, utility is measured from the provider's perspective, typically an error measure of the mining task at hand (e.g., classification error)

[5]. In the specific context of recommender systems, rating prediction accuracy [27, 32] and category aggregates [38] are typically used as proxies for utility. Techniques for user profile perturbation have also been studied for utility-preserving differentially-private recommeders [17].

9 CONCLUSIONS

We presented Mediator Accounts (MAs): a framework to counter user profiling while preserving individual user utility as much as possible. The framework enables decoupling users from accounts, making direct targeting impossible, and profile reconstruction or de-anonymization much harder. At the same time, users are still able to benefit from personalization by service providers. The versatility of the framework has been demonstrated in two different application scenarios. While our model allows for flexible trade-offs between privacy and utility, a key question in our two empirical studies has been to understand how well the MAs can preserve the utility in terms of high-quality search results and recommendations. The experiments show that the split-merge approach with Coherent assignment improves the privacy, while incurring little user utility loss. These benefits are most pronounced for users with larger profiles (i.e., more activity) and higher diversity of interests.

Open issues for future work include practical deployment, handling of other personalization features, and exploring the options for tuning assignments and framework parameters to the specific needs of individual users. On top of that, analyzing the three-dimensional trade-off between user privacy, user utility and the traditional service provider utility could help ensure that the resulting mediator profiles are a useful source for user analytics, making an MA proxy a tolerable component of the online landscape.

Finally, we would hope that the MA proposal stirs up the investigation of how the need-to-know principle could be implemented in case of personalized online services.

Acknowledgements. We would like to thank Subhabrata Mukherjee from MPII for useful discussions at various stages of this work.

REFERENCES

[1] L. Agarwal, N. Shrivastava, S. Jaiswal, and S. Panjwani. Do Not Embarrass: Re-examining User Concerns for Online Tracking and Advertising. In *SOUPS '13*.
[2] C. Aggarwal. *Data mining: The textbook*. Springer, 2015.
[3] I. E. Akkus, R. Chen, M. Hardt, P. Francis, and J. Gehrke. Non-tracking web analytics. In *CCS '12*.
[4] E. Balsa, C. Troncoso, and C. Diaz. Ob-pws: Obfuscation-based private web search. In *S&P '12*.
[5] E. Bertino, D. Lin, and W. Jiang. A survey of quantification of privacy preserving data mining algorithms. In *Privacy-preserving data mining '08*.
[6] J. A. Biega, K. P. Gummadi, I. Mele, D. Milchevski, C. Tryfonopoulos, and G. Weikum. R-Susceptibility: An IR-centric approach to assessing privacy risks for users in online communities. In *SIGIR '16*.
[7] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala. Privacy preserving data publishing. *Foundations and Trends in Databases '09*.
[8] G. Chen, H. Bai, L. Shou, K. Chen, and Y. Gao. UPS: efficient privacy protection in personalized web search. In *SIGIR '11*.
[9] A. Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. *TWeb '08*.
[10] W. B. Croft, D. Metzler, and T. Strohmann. *Search engines*. Pearson Education, Inc., 2010.
[11] C. Dwork. Differential Privacy: A Survey of Results. In *TAMC '08*.
[12] L. Fan, L. Bonomi, L. Xiong, and V. Sunderam. Monitoring Web browsing behavior with differential privacy. In *WWW '14*.
[13] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving Data Publishing: A Survey of Recent Developments. *ACM Computing Surveys '10*.
[14] A. Gervais, R. Shokri, A. Singla, S. Capkun, and V. Lenders. Quantifying web-search privacy. In *CCS '14*.
[15] M. T. Goodrich, M. Mitzenmacher, O. Ohrimenko, and R. Tamassia. Privacy-preserving group data access via stateless oblivious RAM simulation. In *SODA '12*.
[16] M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Publishing search logs: A comparative study of privacy guarantees. *TKDE '14*.
[17] R. Guerraoui, A.-M. Kermarrec, R. Patra, and M. Taziki. D2P: distance-based differential privacy in recommenders. *VLDB '15*.
[18] M. Halkidi and I. Koutsopoulos. A game theoretic framework for data privacy preservation in recommender systems. In *ECML PKDD'11*.
[19] A. Hannak, P. Sapiezynski, A. Molavi Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson. Measuring personalization of web search. In *WWW '13*.
[20] X. He, A. Machanavajjhala, and B. Ding. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *SIGMOD '14*.
[21] R. Jones, R. Kumar, B. Pang, and A. Tomkins. Vanity fair: Privacy in querylog bundles. In *CIKM '08*.
[22] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer '09*.
[23] A. Krause and E. Horvitz. A utility-theoretic approach to privacy in online services. *JAIR '10*.
[24] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *USENIX Security '16*.
[25] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. *ICDE '07*.
[26] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *TKDD '07*.
[27] F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the Netflix prize contenders. In *KDD '09*.
[28] W. Meng, R. Ding, S. P. Chung, S. Han, and W. Lee. The Price of Free: Privacy Leakage in Personalized Mobile In-App Ads. In *NDSS '16*.
[29] W. Meng, B. Lee, X. Xing, and W. Lee. TrackMeOrNot: Enabling Flexible Control on Web Tracking. In *WWW '16*.
[30] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *WWW '12*.
[31] A. Narayanan and V. Shmatikov. Obfuscated databases and group privacy. In *CCS '05*.
[32] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh. Privacy-preserving matrix factorization. In *CCS '13*.
[33] S. T. Peddinti and N. Saxena. Web search query privacy: Evaluating query obfuscation and anonymizing networks. *JCS '14*.
[34] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer. Query profile obfuscation by means of optimal query exchange between users. *TDSC '12*.
[35] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web transactions. *TISSEC '98*.
[36] N. Santos, A. Mislove, M. Dischinger, and K. Gummadi. Anonymity in the personalized web. In *NSDI Posters '08*, 2008.
[37] X. Shen, B. Tan, and C. Zhai. Privacy protection in personalized search. In *SIGIR Forum '07*.
[38] Y. Shen and H. Jin. Epicrec: Towards practical differentially private framework for personalized recommendation. In *CCS '16*.
[39] A. Singla, E. Horvitz, E. Kamar, and R. White. Stochastic privacy. In *AAAI '14*.
[40] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *International Conference on Intelligent Analysis '05*.
[41] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems '02*.
[42] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05*.
[43] P. Wang and C. V. Ravishankar. On masking topical intent in keyword search. In *ICDE '14*.
[44] Y. Xu, K. Wang, G. Yang, and A. W. Fu. Online anonymity for personalized web services. In *CIKM '09*.
[45] Y. Xu, K. Wang, B. Zhang, and Z. Chen. Privacy-enhancing personalized web search. In *WWW '07*.
[46] H. Yang, I. Soboroff, L. Xiong, C. L. Clarke, and S. L. Garfinkel. Privacy-Preserving IR 2016: Differential Privacy, Search, and Social Media. In *SIGIR '16*.
[47] Z. Yu, S. Macbeth, K. Modi, and J. M. Pujol. Tracking the trackers. In *WWW '16*.
[48] S. Zhang, G. H. Yang, and L. Singh. Anonymizing query logs by differential privacy. In *SIGIR'16*.
[49] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the Netflix prize. In *AAIM '08*.
[50] Y. Zhu, L. Xiong, and C. Verdery. Anonymizing user profiles for personalized web search. In *WWW '10*.