

Where you are is who you are: User identification by matching statistics

U. Kiran Kumar¹, M. Padmavathamma²

¹Student, dept. of computer science, SVU College of cm & cs, SV University, Tirupati

²Professor & BOS Chair person, dept. of computer science, SVU College of cm & cs, SV University, Tirupati

Abstract- Most users of online services have unique behavioral or usage patterns. These behavioral patterns can be exploited to identify and track users by using only the observed patterns in the behavior. We study the task of identifying users from statistics of their behavioral patterns. Specifically, we focus on the setting in which we are given histograms of users' data collected during two different experiments. We assume that, in the first dataset, the users' identities are anonymized or hidden and that, in the second dataset, their identities are known. We study the task of identifying the users by matching the histograms of their data in the first dataset with the histograms from the second dataset. In recent works [1], [2] the optimal algorithm for this user identification task is introduced. In this paper, we evaluate the effectiveness of this method on three different types of datasets with up to 50,000 users, and in multiple scenarios. Using datasets such as call data records, web browsing histories, and GPS trajectories, we demonstrate that a large fraction of users can be easily identified given only histograms of their data; hence these histograms can act as users' fingerprints. We also verify that simultaneous identification of users achieves better performance compared to one-by-one user identification. Furthermore, we show that using the optimal method for identification indeed gives higher identification accuracy than heuristics-based approaches in practical scenarios. The accuracy obtained under this optimal method can thus be used to quantify the maximum level of user identification that is possible in such settings. We show that the key factors affecting the accuracy of the optimal identification algorithm are the duration of the data collection, the number of users in the anonymized dataset, and the resolution of the dataset. We also analyze the effectiveness of k -anonymization in resisting user identification attacks on these datasets.

A common task in data analysis is to identify users by exploiting statistics of their data. In many applications, we have access to some information about a set of users from one source, and some other information about the set of users from another source, and the task is to match pieces of information from the first source to pieces of information from the second source that correspond to the same underlying user. If the identities of the users in the two sets are known, then this is a trivial task. However, in many practical applications, the identities of the users are unknown either in the first set or in the second set or in both; therefore, in such cases, the task becomes non-trivial. For example, the two datasets might contain information about location statistics of users in a city measured over distinct time periods.

The problem of matching users is also relevant in the context of privacy of an anonymized database. In recent years, many datasets containing information about individuals have been released into the public domain in order to provide open access to statistics or to facilitate data-mining research. Often these databases are anonymized by suppressing identifiers that reveal the identities of the users, such as names or social security numbers. Nevertheless, recent research has revealed that the privacy offered by such anonymized databases could be compromised, if an adversary correlates the revealed information with auxiliary information about the users from publicly available databases. A famous example of such a de-anonymization attack was shown in [4], in which anonymous movie ratings released during the Netflix Prize contest were de-anonymized by using public user reviews from the Internet Movie Database (IMDB). In such attacks, the adversary's task of de-

I. INTRODUCTION

anonymization is essentially a matching task. The objective is to identify users in the anonymized dataset by matching their data to the publicly available auxiliary information.

As the question of matching users is relevant in many applications, this problem has been studied by many authors in different fields, including database management [5], information retrieval [6], natural language processing [7], author identification [8], [9], and privacy [4]. Nevertheless, most solutions to the matching problem rely on heuristics that are relevant for specific applications, but not for other applications. In this paper, we present a systematic study of the matching problem under a general setting. The problem we study differs

(a) Unlabeled histograms (Day 1)				(b) Labeled histograms (Day 2)			
User	Location			User	Location		
	Dorm.	Rest.	Lib.		Dorm.	Rest.	Lib.
?	75%	15%	10%	John	33%	33%	34%
?	31%	30%	39%	Jill	70%	20%	10%
?	15%	15%	70%	Mary	15%	60%	25%
?	15%	65%	20%	Mike	15%	20%	65%

TABLE I

AN ILLUSTRATIVE EXAMPLE OF USER IDENTIFICATION TASK ON HISTOGRAMS. LOCATION STATISTICS IN THE FORM OF HISTOGRAMS OF SOME USERS ARE RELEASED (IN (A)), WHERE THE USER IDENTITIES ARE REMOVED. AN ADVERSARY HAS ACCESS TO SOME AUXILIARY HISTOGRAMS (IN (B)) ABOUT THE SAME USERS WHERE THE USER IDENTITIES ARE KNOWN. THE TIME PERIOD DURING WHICH THE HISTOGRAMS IN (A) ARE COLLECTED DOES NOT OVERLAP WITH THE TIME PERIOD DURING WHICH THE HISTOGRAMS IN (B) ARE COLLECTED. THE OBJECTIVE OF THE ADVERSARY IS TO MATCH THE USERS (I.E., ROWS) ACROSS THE TWO TABLES.

from typical approaches in data analysis in that we focus on the setting in which the available information about a user's data is in the form of histograms of the user's data. The histograms capture the habits of the users. In the case of mobility traces, such histograms could be the average time spent by each user at different locations during a day, or during different time intervals. In some applications, such as urban planning, the data collected naturally contains only the statistics of the data, as they are sufficient for such applications. In other applications, the data is intentionally stripped of timing information to enhance the privacy of the users; in which case, all that remains are histograms. We study the problem of matching histograms of users' data measured in two independent experiments as a hypothesis testing problem. This novel formulation has the advantage of making it possible to rigorously define the accuracy of a matching scheme and to identify an algorithm that is provably more accurate than other schemes.

An example of a user identification task, is to consider a dataset comprised of unlabeled location histograms, given in Table I(a), where the user identities are removed. Now consider an adversary who has access to the labeled location histograms of the same users in an independent experiment where the user identities are known (refer to Table I(b)). This information could be obtained, for instance, by tracking the users during a different time-period compared to those in Table I(a). The histograms corresponding to each user in the two tables are expected to be similar, as the habits of the user are expected to remain the same across the two datasets; but they might not be exactly identical due to the inherent randomness in the user's behavior. The objective of the adversary is to match the user identities (i.e., the rows) across the two tables.

In the next section, we provide a detailed comparison of this problem with existing literature on user identification and highlight the new contributions of this work. We state the problem in mathematical form and propose our solution in Section III. We experimentally evaluate our solution by using three different datasets in Section IV. In Section V, we analyze the efficacy of our algorithm if additional privacy enhancing techniques, such as k-anonymization, are applied to histograms of users' data. We conclude the paper with some discussions in Section VI.

II. RELATED WORK AND CONTRIBUTIONS

The user matching studied in this paper is closely related to several problems that have been studied in other different communities. In this section, we present a comparison of our approach with related problems from several areas, and highlight our contributions relative to existing work.

A. Entity resolution

A matching problem studied in the database community is that of identifying different data records that refer to the same real-world object [5]. Similarly, in natural-language processing, the problem of linking different mentions of the same underlying entity in text [7] is analogous to the objective in the user-matching problem. Another example from the information-retrieval literature is the problem of classifying documents by their

authors, given documents from different authors with the same name [6]. User matching has also been studied in the social-networks community in which the objective is to identify different profiles that belong to the same underlying user [10]. Such problems fall under the umbrella-term entity resolution (ER) [11]. In these problems, the available information about the users is often not in the form of histograms, and the solutions proposed are often based on heuristics and practical convenience; whereas the solution we propose in this paper is specific to the setting in which the only information available about the users is in the form of histograms, and in this setting, the solution is optimal for minimizing the probability of misclassification.

B. De-anonymization attacks

Our work is also closely related to the literature on de-anonymization methods [4], [12] studied in the literature on privacy. A number of works on de-anonymization focus on demonstrating that even when users' data are anonymized, the data belonging to each user is often unique. In such examples, an adversary who has access to auxiliary information about the users can de-anonymize the anonymized datasets by exploiting the uniqueness of the data belonging to each user. For example, in [13] the authors perform a study on the top k locations most frequently visited by users in a nationwide call-data record (CDR) dataset. They consider various levels of spatial granularity (such as sector, cell, zip code, city, state, and country) and temporal granularity (such as day and month), and they show that the most frequently visited locations can act as quasi-identifiers to re-identify anonymous users. Thus an adversary can de-anonymize such a dataset by obtaining access to auxiliary information about the users' zip codes and times of activity. The adversary's goal is essentially a matching task, i.e., the adversary seeks to match the auxiliary information about the users with the unique aspects of the users' data. Some other works such as [13]–[19] study the uniqueness of mobility data traces. There is a line of work on studying the uniqueness of web browsing history patterns of users [20],[21]. In [20] the authors consider a dataset where every record is the set of visited websites by a user during some period of time. The authors investigate how

unique is a single user's record compared to other users' records in the dataset.

Although our work is related to de-anonymization, it differs in several aspects. First, we assume that the only information about the users in the two datasets are time-averaged statistics of the users' data. In most works on user matching and de-anonymization [4], [22], [23], the vulnerability to privacy breaches often arises due to the sparsity of the temporal evolution of the users' data. For instance, the fact that a user watched and rated a movie during a particular time-period or was at a specific location during a particular time can be used to easily identify the user's data from the anonymized dataset. Other de-anonymization works focus on identifying the temporal patterns of the data collected from the users. For example, in [17], [18], a Markov model is constructed based on the temporal evolution of the mobility patterns of the users, and then similarity measures are used for de-anonymization. Such temporal information in the users' data, however, is removed when only statistics in the form of histograms from each user is collected or released. Often this results in a much lower uniqueness in the information available about the users; hence matching users' statistics is, in general, much more difficult than matching users' datasets.

Second, we assume that the two sets of the statistical information are mutually statistically independent. For example, in the case of mobility data, this could be because the two datasets were obtained over different time periods. We seek to perform the matching by only exploiting the fact that users' habits remain stationary and ergodic across the two datasets. This is in contrast to the approach of works such as [15], [16], [24] that perform de-anonymization by using auxiliary information collected over the same period of time as the anonymized dataset. In such cases, the auxiliary information is not independent of the anonymized user data. In [20], the authors investigate the stability of the set of visited websites by a user across time. In particular, they record the set of visited websites by a user during one day. They use the Jaccard index to measure the similarity between the sets collected for one user over different days. They show that the set of visited websites by a user is stable during a four-week period. A special case of our work is when the labeled and unlabeled

histograms are obtained from the same source in different time periods. The accuracy of our matching algorithm in such cases is dependent on how much the statistical characteristics of the data is preserved over time.

Third, we perform simultaneous matching of the information available about all users and not one user at a time. Simultaneous matching takes into account all the information available about the users at the same time, and hence out-performs matching users one at a time. Simultaneously taking into account all the information for various attacks has already been employed in different domains [9], [25]–[27] and in this paper we employ it in the domain of histogram matching. There is also a related line of work on graph de-anonymization, also known as graph alignment [22], [23], [28]. It is the problem of matching the nodes across two

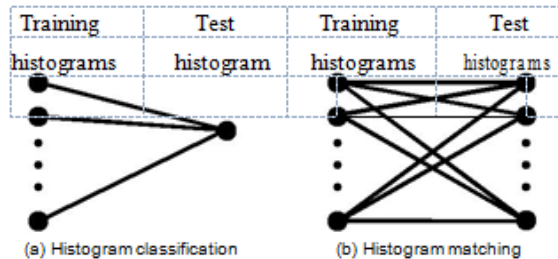


Fig. 1. (a) The problem of histogram classification, which is to classify the test histogram to the correct class based on the training histograms.

(b) The problem of histogram matching studied in this paper, which is to simultaneously classify the test histograms to the training histograms subject to the constraint that each test histogram belongs to a distinct class.

similar graphs, where the only available information is the two graphs. For example, given the graph of connections between users on two different social networks (e.g., Facebook and LinkedIn), it might be possible to match users across the two social networks by exploiting the fact that the structure of the underlying graphs are expected to be similar. This problem is different from that studied in the present paper because, in our setting, the graph-based connections among the users are not available.

C. Supervised learning

The matching task studied in this paper is closely related to the classification task studied in supervised learning [29], where the objective is to classify test data to the correct class based on labeled training data observed under each of the classes. Nevertheless, a key aspect of our approach that differs from supervised learning is that we seek to simultaneously

classify test data that belong to a group of users subject to the constraint that each user belongs to a distinct class (refer to Figure 1). Thus our solution, originally introduced in [1], [2], can be interpreted as a solution to a constrained classification problem. Our solution is tailored to the setting in which the available information is in the form of histograms. It could be possible to extend this solution to more general kinds of data by combining the matching algorithm presented in this work with feature extraction techniques in machine learning [29].

D. Contributions

Compared to existing works on the user-matching problem, our work is unique in several respects. Our main contributions can be summarized as follows:

- We demonstrate that statistics about users' behaviors contain a significant amount of information that can be used as fingerprints to uniquely identify users, by an adversary who has access to auxiliary information about the users. Moreover, we show that identification by using only data statistics can sometimes result in accuracy higher than existing methods based on more complicated data models (e.g., Markov Chains).
- We evaluate a provably optimal algorithm for matching users' statistics on three datasets of diverse nature and demonstrate that it outperforms heuristics-based methods. We address the practical setting of performing the matching across distinct sets of users.
- We compare the performance of our algorithm with different parameters and under different settings, such as user configuration and data resolution. We verify that, in particular, matching users simultaneously leads to a matching accuracy significantly higher than matching one user at a time.
- We analyze the performance of the matching algorithm under different privacy-preserving mechanisms such as data obfuscation and k-anonymization.

III. PROBLEM STATEMENT AND PROPOSED SOLUTION

We assume that the data belonging to each user in our system follows some fixed underlying probability law that is unknown a priori. The probability law associated with each user is unique and captures the habits of the user. For example, in the case of web-browsing histories, the probability law captures the user's relative preferences for various websites. Similarly, in the case of shopping data, the probability law could represent shopping preferences and, in the case of mobility data, the law could represent the preferences for visiting various locations. In the basic version of the user-matching problem, we are given two datasets corresponding to the same set of users, and the task is to match users across the two datasets by exploiting the fact that the underlying probability law of each user is unique. We will later generalize this to the setting in which the two datasets belong to different sets of users. Throughout this paper, we focus on the specific setting in which each dataset reveals only the histograms of each user's data and not the data itself. We use the term adversary to denote the entity that performs the user-matching task. We use feminine pronouns for referring to the users and masculine pronoun for referring to the adversary. In the following, we state the problem mathematically.

A. Problem statement

Consider a discrete alphabet set $S = \{S_1, S_2, \dots, S_K\}$ of size $|S| = K$ and a set of N users labeled $1, 2, \dots, N$. The set S represents the set of all possible values that can be taken by each instance of the data belonging to each user. For example, in the case of web-browsing data, S is the set of all websites that a user could visit, and in the case of mobility data, S is the set of all possible locations (e.g., regions of a city) that a user could visit. For the purpose of illustration, in the rest of this section, we will focus on the example of mobility data.

For a data string $s = [s(1), s(2), \dots, s(T)] \in \mathcal{S}^T$ of length T , we use Γ_s to denote the histogram (i.e., empirical distribution) of the string defined as

$$\Gamma_s(l) = \frac{1}{T} \sum_{t=1}^T \mathbb{I}\{s(t) = S_l\}, \quad l = 1, 2, \dots, K. \quad (1)$$

Generally during decision making process taking opinions from people is a common criterion. In the simplest version of the user-matching problem studied in this paper, we are given two sets of

histograms of the data generated by each of the users. Let set $\psi_1 = \{\Gamma_{x1}, \Gamma_{x2}, \dots, \Gamma_{xN}\}$ represent a set of unlabeled histograms each generated by a distinct user, and let $\psi_2 = \{\Gamma_{y1}, \Gamma_{y2}, \dots, \Gamma_{yN}\}$ represent a set of labeled histograms each generated by a distinct user. Here ψ_1 and ψ_2 represent the histograms contained in two datasets. In the case of mobility data, ψ_1 is a set of anonymized histograms of users' mobility traces that are released, and ψ_2 represents the auxiliary histograms of the users' mobility traces, which is obtained by an adversary by tracking the users over a time period. In other applications, the auxiliary histograms can be obtained by the adversary by using publicly available information. In both cases, the adversary is aware of the users' identities in the second dataset, and seeks to decode the identities of the users in the first anonymized set of histograms. The histograms of each user are assumed to be statistically independent of those of others. Furthermore, for each user, the histogram generated by the user in the first dataset is assumed to be independent of the histogram in the second dataset. In the mobility example, independence is a reasonable assumption provided that there is no overlap between the time-periods over which the histograms in ψ_1 and ψ_2 are computed. For example, ψ_1 contains histograms collected over a week and ψ_2 contains histograms collected over the following week.

In the matching problem, the objective of the adversary is to determine the true matching between the histograms of ψ_1 and ψ_2 . We represent the ground truth via an unknown permutation function,

$$\sigma: \{1, 2, \dots, N\} \rightarrow \{1, 2, \dots, N\} \quad (2)$$

such that, in reality, for each $i \in \{1, 2, \dots, N\}$, the histograms $\Gamma_{x\sigma(i)}$ and Γ_{yi} are generated by the same user i . The objective in the matching problem is, equivalently, to estimate σ . In Section III-D, we discuss the practical setting where the histograms in the sets ψ_1 and ψ_2 are generated by different sets of users.

B. Potential approach: Weighted bipartite matching

The problem of matching histograms across two sets can be best visualized as a matching problem on a bipartite graph. Let $G = (V_1, V_2, E)$ be a complete bipartite graph where each vertex in the set V_1

(respectively, set V_2) is associated with a unique histogram in the set ψ_1 (respectively, set ψ_2). There exists an edge from each element in V_1 to each element in V_2 and no edges between elements in V_1 or V_2 . Hence we have a complete bipartite graph where V_1 and V_2 form the two parts. Let node j in set V_1 and node i in set V_2 be associated with histogram Γ_{xj} in ψ_1 and Γ_{yi} in ψ_2 , respectively. The graph G is illustrated in Figure 2.

A matching in graph G is a subset of edges E of G such that no two edges in the subset share a vertex. A maximal matching is a matching such that the addition of any edge to the subset violates the matching property. Let σ_m be a permutation of $\{1, 2, \dots, N\}$, for $m = 1, 2, \dots, N!$. There are $N!$ possible maximal matchings on G corresponding to the $N!$ different permutations. The matching corresponding to permutation σ_m is the matching in which each node i from set V_2 is mapped to node $\sigma_m(i)$ in V_1 ; in other words, histogram Γ_{yi} in ψ_2 is

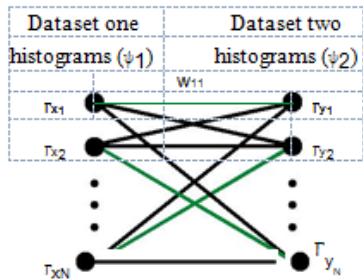


Fig. 2. The problem of matching histograms across two sets can be visualized as a matching problem on a weighted bipartite graph. Corresponding to the $N!$ different permutations, there are $N!$ possible maximal matchings on G . The green edges represent the correct matching associated with σ in (2). The solution can be obtained via a weighted bipartite matching algorithm on the graph with appropriate edge weights.

mapped to histogram $\Gamma_{x\sigma_m(i)}$ in ψ_1 . The matching associated with σ in (2) is shown by green edges in Figure 2.

An intuitive approach for estimating the correct matching between the histograms is as follows. Define a weight for every edge in G such that the weight of the edge w_{ji} from j to i is equal to some appropriately defined distance between the histograms Γ_{xj} and Γ_{yi} , i.e.,

$$w_{ji} = d(\Gamma_{xj}, \Gamma_{yi}) \quad (3)$$

for some distance measure $d(\cdot)$. Now perform a minimum-weight maximal bipartite matching on the resultant weighted bipartite graph. The minimum-weight maximal matching corresponds to a configuration where the sum of the distances between

the matched histograms is minimum, hence expected to provide a good estimate for the correct matching.

The relevant questions that arise here are: What is a good choice for the distance measure between histograms and does the choice of measure depend on the nature of the data or can there be a general-purpose measure? The literature contains various choices of prevalent distance measures that can be used in the weight function. For example, in [13] the authors use the cosine distance between the histograms of the number of calls of users at different GSM antennas as a distance measure for analyzing the call behavior of users. The cosine distance between histograms Γ_{xj} and Γ_{yi} is defined as the generalized log-likelihood. The first step is therefore to compute the generalized log-likelihood. For hypothesis H_m the generalized likelihood is obtained by maximizing the likelihood function over all possible choices of the π_i 's, and is given by

$$L_m = \sup_{\pi_i} \left(\prod_{i=1}^N \frac{\pi_i(x_{\sigma_m(i)}(t))}{\pi_i(y_i(t))} \right) \quad (8)$$

It is known that for an i.i.d.-generated string, the maximum likelihood estimator of the underlying distribution is given by the empirical distribution of the string. Hence, it is easy to see that each of the N terms in the summation (8) is maximized

Alternatively, we can use a similarity measure, such as the dot product defined in (5) as the weight function in (3). We then identify the best permutation by using a maximum weight matching on the resultant weighted bipartite graph. In the next subsection, we present a new choice of the weight function and argue that it is a judicious choice.

C. Optimal solution via hypothesis testing interpretation

The problem of finding the matching between the histograms of ψ_1 and ψ_2 can be viewed as a multi-hypothesis testing problem with $N!$ hypotheses, $\{H_1, H_2, \dots, H_{N!}\}$, where hypothesis H_m corresponds to permutation σ_m , for $m = 1, 2, \dots, N!$. In the hypothesis testing framework, we study decision rules by using probability of error under the different hypotheses as the performance metric. Typical solutions to hypothesis testing problems seek the

decision rule that leads to an optimal trade off between various error probabilities under the different hypotheses. In our prior works [1], [2], we showed that, when each user's data is generated by an i.i.d. process governed by her probability law, an optimal trade-off between the various error probabilities for the matching problem is obtained by deciding in favor of the hypothesis corresponding to the minimum-weight maximal matching on the bipartite graph G with edge weights

$$w_{ji} = D(\Gamma_{x_i} \| \frac{1}{2}(\Gamma_{x_i} + \Gamma_{y_i})) + D(\Gamma_{y_i} \| \frac{1}{2}(\Gamma_{x_i} + \Gamma_{y_i})). \quad (7)$$

In (7), $D(\bullet \| \bullet)$ is the Kullback-Leibler divergence function [30], defined as

$$D(\pi \| \mu) = \sum_{i=1}^K \pi(i) \log \left(\frac{\pi(i)}{\mu(i)} \right).$$

by setting $\pi_i = \frac{1}{2}(\Gamma_{x_i} + \Gamma_{y_i})$, for $i = 1, 2, \dots, N$. We can therefore rewrite (8) as

$$H(\pi) = -2T \sum_{i=1}^N \pi(i) \log \left(\frac{\pi(i)}{\mu(i)} \right) + D(\Gamma_{x_i} \| \frac{1}{2}(\Gamma_{x_i} + \Gamma_{y_i})) + D(\Gamma_{y_i} \| \frac{1}{2}(\Gamma_{x_i} + \Gamma_{y_i})).$$

where $H(\cdot)$ is the Shannon entropy function [30], defined as

$$H(\pi) = -\sum_{i=1}^N \pi(i) \log(\pi(i)).$$

The & $H = \arg \max_{\pi \in \Pi} H(\pi)$ where Π is the set of histograms and $\{y_i\}$ the term $H(\Gamma_{x_i} \| \frac{1}{2}(\Gamma_{x_i} + \Gamma_{y_i})) + H(\Gamma_{y_i} \| \frac{1}{2}(\Gamma_{x_i} + \Gamma_{y_i}))$ is a constant term that & Hence, by removing this constant term we can show that $H = \arg \min_{\pi \in \Pi} H(\pi)$ interpreted as the hypothesis

corresponding to the minimum-weight maximal matching on the complete bipartite graph G in Figure 2 with weights (7).

Although this optimality result was established for i.i.d. processes, we argue that the solution is a reasonable approach to use for the matching problem, provided that each user's habits follow a probability law that is stationary and ergodic. In such cases, we expect the histograms of each user in the two datasets to be similar, hence the solution for i.i.d. data is well-justified. Therefore, in this paper, we propose to use the solution given by the minimum-weight maximal matching on G with the weight metric in (7). We demonstrate, in our experiments in Section IV, that the matching accuracy obtained by using (7) is

indeed higher than those obtained by using (4), (5), and (6) under various settings.

via the maximum-weight matching on G . In this case, after negating all the edge-weight values and shifting them to make them positive, (A1) and (A2) can be used to identify the matching solution.

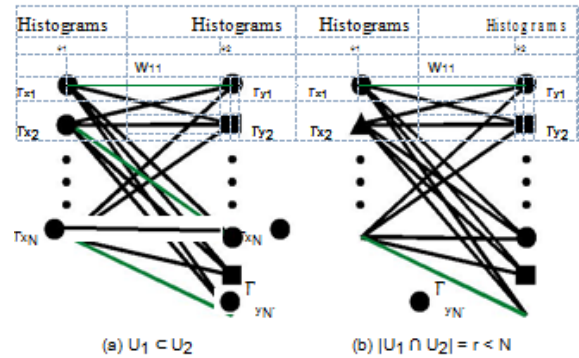


Fig. 3. Matching problem when the histograms in sets ψ_1 and ψ_2 belong to different sets of distinct users (i.e., $U_1 \neq U_2$). Histograms belonging to users in $U_1 \cap U_2$ are marked by black circles and histograms in the sets $U_1 \setminus U_2$ and $U_2 \setminus U_1$ are marked by black triangles and squares, respectively. In (a), $U_1 \subset U_2$ with $|U_1| = N$. The proposed solution is given by the minimum-weight maximal matching of the graph. In (b), $|U_1 \cap U_2| = r < N$. In this case, the proposed solution is given by the minimum-weight matching with cardinality r on the graph. The green edges represent the correct matching between the histograms in the set $U_1 \cap U_2$.

The Hungarian algorithm [32] is a popular and efficient algorithm for (A1) and can be adapted to solve (A2) as explained in [31]. In our experiments, we use the Hungarian algorithm for (A1) and a polynomial-time algorithm, based on the theory of matroids (see, e.g., [33, Ch. 8]), for (A2). The time-complexity of obtaining the matching solution on the graph G by using the Hungarian algorithm is $O(|U_1||U_2||U_1 \cap U_2|)$; i.e., it is $O(N^3)$, $O(N^2N')$, and $O(NN'r)$ for (i), (ii), and (iii), respectively. In practice, the complexity can often be reduced significantly. For instance, when histograms Γ_{x_j} and Γ_{y_i} have disjoint support, then w_{ji} in (7) takes its maximum value, which is $2 \log(2)$. Then the edge connecting the corresponding vertices in G can be removed, as it will almost certainly not be selected in the minimum-weight maximal matching. If the resulting graph has E edges, then the complexity is $O(E|U_1 \cap U_2|)$.

In a practical implementation of this de-anonymization approach, the overall complexity depends on both the complexity of computing the edge weights in graph G and of running the matching algorithm (A1) or (A2) on graph G . The former has complexity $O(NN'K)$ where K is the number of locations. In Section IV-D we present detailed time-

complexity results of our de-anonymization approach.

An alternative approach for solving (A1) and (A2) is to use an approximate minimum-weight matching algorithm on graph G instead of the Hungarian algorithm. Although finding the exact minimum-weight matching solution has the advantage of obtaining the maximum matching accuracy, it brings the inherent computational complexity of weighted bipartite matching into our solution. This could hinder the applicability of our solution to very large datasets as the number of histograms

becomes very large. An alternative approach in dealing with very large datasets is to obtain an approximate minimum-weight matching solution on graph G [34]. Although this approach reduces the matching accuracy, it makes it possible to find an approximate solution in reasonable time. For example by using the approach in [34], a $(1 - \epsilon)$ -approximate matching solution to (A1) in case (i) can be obtained with complexity $O(N^2 \epsilon^{-1} \log \epsilon^{-1})$ instead of $O(N^3)$.

IV. EXPERIMENTAL EVALUATIONS

In this section we compare the performance of the proposed matching algorithm with other methods for user identification. Although numerous identification algorithms exist in the literature, we perform comparisons primarily with identification methods that rely only on histogram information as the focus of this paper is on such methods. Nevertheless, in Section IV-C4 we compare our approach with an existing Markov-based method, for which histograms are only a subset of the information available to the method. We show that by using only histograms we can still get better de-anonymization accuracy than the Markov-based approach that exploits more information from the dataset.

We test our matching algorithms on three datasets of different nature. The first is a call-data records dataset, the second is a web browsing-history dataset, and the third is a dataset of GPS mobility traces. In our experiments, a location represents the coverage region of a GSM antenna, a website, and a region on the map in the first, second, and third dataset, respectively. We interpret the sequence of locations visited by a user as a data string. Thus a user's histogram is simply the relative fractions of visits of the user to the different locations, within the time

period considered. For each dataset, we compute the histograms of the users over two different non-overlapping time periods to obtain the sets ψ_1 and ψ_2 described in Section III-A. We then construct the complete bipartite graphs G shown in Figures 2, 3(a), and 3(b) and apply the matching algorithms proposed in Sections III and III-D on this graph with appropriately chosen edge-weights. We estimate the matching accuracy obtained with the different algorithms by calculating the percentage of common users (i.e., users in the set $U_1 \cap U_2$) whose histograms are correctly matched. We recall that we focus on the privacy from the perspective of the adversary and not of the users; hence, this particular choice for notion of accuracy is reasonable.

A. Experiments on call-data records (CDR)

1) Dataset description and preprocessing: The call-data records (CDR) dataset consists of anonymized records of phone calls between 50,000 Orange customers (i.e., users) in Ivory Coast [35], chosen randomly from millions of users. The dataset covers the two-week period from Monday 9th to Sunday 22nd of April 2012 and contains the time of every call made by every user and the identifier of the antenna to which the user was connected when making the call. Figure IV-A1 shows a map of Ivory Coast with the positions of 1237 antennas in the country indicated by black circles [35].



Fig. 4. (a) Position of Orange's GSM antennas in Ivory Coast [35]. The sub-prefectures are shown by different colors.

Dataset	Characteristics		Choice of metric in (A1)			
	N	K	proposed	l_1	cosine	dot
CDR	46986	1211	21.1%	18.6%	16.4%	13.3%
WBH	121	83219	90.0%	81.1%	72.7%	64.4%
GL	154	1024	58.4%	51.3%	52.0%	46.8%

TABLE II

MATCHING ACCURACY OBTAINED ON G IN FIGURE 2 BY USING (A1) WITH VARIOUS CHOICES FOR THE DISTANCE/SIMILARITY MEASURES BETWEEN THE HISTOGRAMS DEFINED IN (7), (6), (5), AND (4). THE PROPOSED WEIGHT FUNCTION CONSISTENTLY YIELDS THE HIGHEST ACCURACY FOR ALL THREE DATASETS.

We first split the CDR dataset into two parts, where part one corresponds to the calls made in the first one-week period from the 9th to 15th of April, and part two corresponds to the calls made in the second one-week period from the 16th to 22nd of April. We then restrict our attention only to users who are active in both weeks, i.e., the users who made at least one call in each of the two weeks. There are $N = 46986$ such users, and overall they connected to $K = 1211$ antennas. Each user, on average, made 101.2 calls and connected to 6.7 different antennas. We consider the coverage region of each antenna to be a location. We disregard the timing information of the calls and construct the histograms of the calling patterns of each user in each week. Thus, the histogram $\Gamma_{x\sigma}(i)$ (respectively, Γ_{y_i}) of user i in the first (respectively, second) week gives the relative fractions of calls made by the user in various locations in the first (respectively, second) week. The set ψ_1 (respectively, ψ_2) consists of the histograms computed over the first week (respectively, second week).

2) Matching accuracy with different metrics: After computing the histograms, we construct the complete bipartite graph G shown in Figure 2 and described in Section III-B. We choose edge weights w_{ji} given in (7) and compute, by using (A1), a minimum-weight maximal matching on G . The obtained result is shown in the first row of Table II. Of 46986 users, 9927 are correctly matched, which gives an accuracy of 21.1%. This means that, given the proportions of calls of users from different antennas during two consecutive weeks, we are able to correctly match more than one-fifth of them. We also compare the matching accuracy obtained by using the distance measure (7) with the accuracy obtained by using the distance measures given in (4) and (6), as well as the similarity measure of (5). We observe from the table that the matching

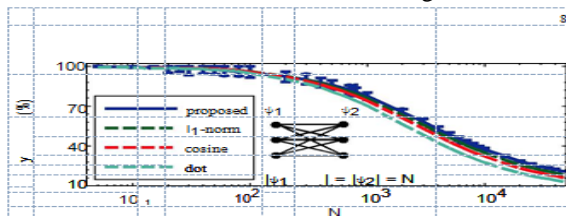


Fig. 5. The obtained average accuracy by using different edge weights as a function of the number of users N , in the setting where the histograms in sets ψ_1 and ψ_2 are generated by the same set of N users (i.e., $U_1 = U_2$). The measures are defined in (7), (8), (4), and (5). The 90% confidence interval is also shown for our proposed metric. We observe that increasing the number of users N leads to a reduction in the matching accuracy.

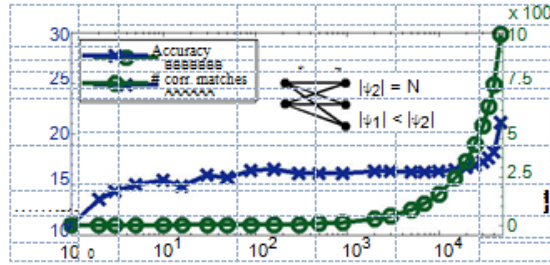
accuracy obtained by using the weight function proposed in (7) is significantly higher than that obtained by using any of the other heuristic measures. We remark that the naive approach of deciding on a purely random matching between the histograms yields, on average, one correctly matched user. The resulting accuracy (0.002%) is negligible compared to those obtained in Table II.

3) Effect of varying the number N of users: In this experiment, we keep $U_1 = U_2$ but vary $|U_1|$. We first choose uniformly at random a subset of the 46986 users considered in the previous experiment. We denote the subset size by N . We then choose sets ψ_1 and ψ_2 to be the histograms associated with the N chosen users in the first week and the second week, respectively. We then apply (A1) to the graph G of Figure 2 with different choices of edge weights. For each value of N we repeat the experiment several times, choosing the subset randomly and performing the matching. The obtained average accuracies are shown in Figure 5 as a function of N for each choice of edge weight. We observe from Figure 5 that as the value of N increases, the matching accuracy under all metrics decreases. This is expected because as N increases, the habits of the users start resembling those of others, and it becomes more difficult to distinguish the histograms of one user from those of others. Hence, the matching accuracy decreases. Furthermore, although the 21.1% accuracy obtained with the proposed metric of (7) in Table II might seem small at first, it is associated with a large value of N . If the number of users were smaller, the accuracy would be higher (e.g., 78% for 1000 users).

4) Matching different subsets of users: Following the discussion in Section III-D, here we investigate the practical scenario where the histograms in sets ψ_1 and ψ_2 belong to different sets of distinct users. In other words, in this experiment $U_1 \neq U_2$. We first consider the setting in which we are given histograms of all users on the second week but only a subset of users on the first week. That is, $U_1 \subset U_2$, as depicted in Figure 3(a).

We let ψ_2 be the collection of histograms of all the $N = 46986$ users on the second week. For ψ_1 we use the collection of histograms of a randomly chosen

subset of users on the first week. We construct G in Figure 3(a) with edge weights



Number of histograms in the first dataset ($|\psi_1|$)
Fig. 6. The average number of correct matches and the average accuracy as a function of $|\psi_1|$ in the setting where we are given histograms of all users on the second week (i.e., $|\psi_2| = N$) but only a subset of users on the first week (i.e., $|\psi_1| < N$). The leftmost point represents one-by-one user matching approach, which yields the smallest accuracy.

in (7) and run (A1). The resulting matching has a size equal to $|\psi_1|$. The number of correctly matched histograms in the set ψ_1 divided by $|\psi_1|$ defines the obtained accuracy. Figure 6 shows the average number of correct matches and the average accuracy obtained for different values of $|\psi_1|$, where the results are averaged over several repetitions of the experiment. The leftmost point represents one-by-one user matching approach, which yields the smallest accuracy. From a user's perspective, as $|\psi_1|$ increases, the adversary has more information available and thus can obtain a better matching. Hence, the obtained matching accuracy increases. This observation has important implications in the perspective of privacy of anonymized statistics. A user's privacy depends not only on how much her trajectory is revealed to the adversary, but also on how much of others' trajectories are revealed to the adversary.

In the second part of this experiment, we consider the setting where $|U_1 \cap U_2| = r < N$. This is the setting depicted in Figure 3(b). We choose uniformly at random a set of histograms from the first week and from the second week, such that $|U_1| = |U_2| = 5000$, and $|U_1 \cap U_2| = 3750$. We choose these values as an example. We then construct G in Figure 3(b) with edge weights given in (7). We first choose 3750 of the unlabeled histograms in U_1 and matched them to 3750 of the labeled histograms in U_2 , such that the summation of the distance between the matched pairs is minimized. We do this by applying (A2) with $r = 3750$ to G . Alternatively, we match all the 5000 unlabeled histograms in U_1 to the labeled histograms in U_2 by applying (A1) to G . The obtained results are shown in Table III. Although the first approach

yields a smaller number of correct matches (1340 versus 1672) compared to the second approach, it yields a larger percentage of correct matches (36% versus 33%). Therefore, it makes sense to use (A2) instead of (A1) when the adversary is interested in maximizing his percentage accuracy (i.e., number of correct matches divided by the size of the outputted matching).

5) Effect of varying the time-duration of data collection: We now investigate how the matching accuracy is affected by the time-duration over which users' statistics are computed. We consider all users who were active on each Monday of

Algorithm	matching size	# correct matches	Percentage accuracy	# incorrect matches
(A2) with $r = 3750$	3750	1340	36%	2410
(A1)	5000	1672	33%	3328

TABLE III

OBTAINED MATCHING RESULT FOR THE CASE $|U_1 \cap U_2| = r < N$ DEPICTED IN FIGURE 3(b) WITH $r = 3750$ AND $N = 5000$. COMPARED TO THE SECOND APPROACH, THE FIRST APPROACH YIELDS A SMALLER NUMBER OF CORRECT MATCHES BUT A LARGER PERCENTAGE OF CORRECT MATCHES.

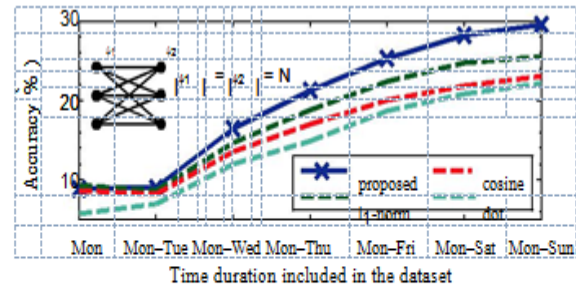


Fig. 7. The obtained matching accuracy ($N = 30937$) with different choices of edge weights as a function of time-duration over which users' statistics are computed. The measures are defined in (7), (6), (4), and (5). As long as users' habits remain stationary and ergodic, by increasing the time-duration over which statistics are computed, histograms belonging to each user become closer to each other, and thus the overall matching accuracy increases.

The two-week period, i.e., users who made at least one call on Monday 9th and on Monday 16th of April. There are $N = 30937$ such users. In the first part of this experiment, the set ψ_1 (respectively, ψ_2) corresponds to the histograms of the number of calls of this N users from the K locations (i.e., GSM antennas) during the first Monday (respectively, second Monday). We then construct graph G illustrated in Figure 2 with different choices of edge weights, and run (A1). The obtained accuracy, marked on the x-axis by "Mon", is shown in Figure 7.

In the second part of this experiment, we increase the time-duration over which we compute users'

statistics. We compute the statistics of the same N users during the Monday and Tuesday of the first week and of the second week. Thus, the set ψ_1 (respectively, ψ_2) now corresponds to the histograms of the number of calls of the N users from the K locations during the first (respectively, second) Monday and Tuesday. We then construct the graph G with different choices of edge weights and run (A1). The obtained matching accuracy, marked by “Mon-Tue”, is shown in Figure 7. Similarly, we increase the number of considered days for every user and repeat the experiment. These results are shown in the figure as well. As can be seen from Figure 7, the matching accuracy increases as we include more days in the dataset. This is because as long as users’ habits remain stationary and ergodic, by increasing the time-duration over which statistics are computed, the two histograms belonging to each user become closer to each other, and thus the overall matching accuracy increases. Furthermore, the matching accuracy obtained by using the weight function proposed in (7) is significantly higher than that obtained by using any of the other heuristic measures. A standout feature in Figure 7 is the fact that the incremental improvement in going from Mon to Mon-Tue is lower than that observed in other data points in the graph. This is probably because Monday, April 9th was White Monday, a public holiday in Ivory Coast. On the following day (i.e., Tuesday 10th of April) the users made on average only 1.9 calls compared to the average of 7.2 calls per day.

6) Effect of location aggregation: In addition to the removal of user identifiers (i.e., anonymization), an additional well-known privacy-protection mechanism that is usually applied to mobility traces is spatial-resolution reduction, which is known also as location obfuscation or location aggregation [36], [37]. Here we investigate the effect of location aggregation on the matching accuracy. The Orange call-data records dataset also includes a low-spatial resolution version [35] that contains the time of every call made by 500,000 randomly chosen users and the sub-prefectures (i.e., administrative divisions within the provinces) of the antennas to which they were connected while making the call. The sub-prefectures, shown by different colors in Figure IV-A1, in general contain multiple antennas, thus the dataset has a spatial resolution lower than the original dataset. We

consider a two-week period and randomly choose a subset of size $N = 46986$ active users out of the total 500,000 users. The set ψ_1 (respectively, ψ_2) corresponds to the histograms of the number of calls of the N users from each sub-prefecture (i.e., location) during the first week (respectively, second week). Users, in total, made calls from $K = 237$ sub-prefectures. We then construct the complete bipartite graph G illustrated in Figure 2 with edge weights given in (7), and run (A1). There are 2070 correctly matched users, which gives an accuracy of 4.40%. The obtained accuracy is much lower than the 21.1% obtained for the same number of users in the original high-resolution dataset. As antennas are aggregated into sub-prefectures, users’ histograms become less distinguishable and, as a result, the matching accuracy drops significantly.

B. Experiments on web browsing history (WBH) dataset

1) Dataset description and preprocessing: The Web History Repository [38] consists of anonymized detailed web browsing history of hundreds of users. Users can upload their anonymized usage data to the repository by using a Mozilla Firefox add-on. In order to protect the users’ privacy, all URLs and hosts are represented by a global unique identifier. The web browsing history (WBH) dataset contains the browsing history of 472 users. Users participated in the data collection for different time-periods during the course of several years. For each user, the dataset contains every visited URL (with encrypted name), the favicon identifier associated with the URL, and the time of visit to the URL. The favicon, also known as a shortcut icon, is a small icon associated with a particular website. Generally, different URLs associated with the same website (e.g., domain name) have the same favicon and hence can be mapped to a single website. For example, if a user visits the URLs “news.yahoo.com” and “mail.yahoo.com”, the URLs will appear with different encrypted names in the database; however, both URLs will have the same favicon

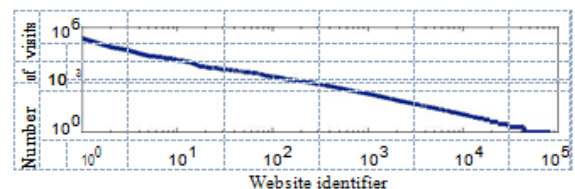


Fig. 8. The total number of visits (i.e., popularity) to the K websites by all the users in the two-week period. The figure is plotted in a log-log scale and the websites are indexed according to their popularity identifier (e.g., “1”). Thus, we can learn that the user has visited a particular website (i.e., “yahoo.com”) twice.

We remove from the dataset all URLs that do not have a favicon. We consider each website (e.g., “yahoo.com”) to be a location and treat the favicon identifier as the website identifier for each URL. We then identify the period of two consecutive weeks that has the maximum number of active users (i.e., users who visit at least one website during each of the two weeks). There are $N = 121$ active users in this two-week period. They visited $K = 83219$ different websites, 77935 of which were visited by not more than one user. Figure 8 shows a log-log plot of the total number of visits to the websites by all the users in the two-week period. The y-axis values represent the popularity of the websites.

We disregard the timing information of the visited websites and construct the histograms of the browsing patterns of each user in each week. Thus, the histogram $\Gamma\chi(i)$ (respectively, $\Gamma\gamma(i)$) of user i in the first (respectively, second) week gives the relative fractions of the visits to various websites by that user in the first (respectively, second) week. The set $\psi 1$ (respectively, $\psi 2$) consists of the histograms computed over the first week (respectively, second week).

2) Matching accuracy with different metrics: We construct the graph G shown in Figure 2 from the histograms and apply (A1) to G with different choices of edge weights. The obtained results are shown in the second row of Table II. We observe that the matching accuracy obtained by using the weight function proposed in (7) is significantly higher than that obtained by using any of the other heuristic measures. Furthermore, given the proportions of visited websites during two consecutive weeks, we are able to correctly match almost all of them.

3) Considering popular websites: One reason we obtain a high matching accuracy is that some websites are visited by only a small number of users during the two-week period, hence it is easy to match those users. We investigate this effect as follows. We

consider all users who visited at least one of the top 5 popular websites, in Figure 8. There are $N = 102$ such users. We consider a subset (of size not less than 5) of the most popular of the visited websites (refer to Figure 8). We then keep for every user i ($1 \leq i \leq 102$) the elements of $\Gamma\chi(i)$ and $\Gamma\gamma(i)$ that correspond to the considered subset of websites, and we set the remaining elements equal to zero. We then re-normalize the remaining histograms such that they sum to one. We reconstruct, by using different choices of edge weights, the bipartite graph G in Figure 2 and run

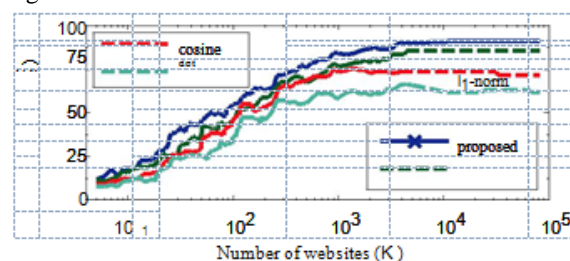


Fig. 9. Matching accuracy for the WBH dataset with $N = 102$ users by using different measures when only a subset of the popular websites are considered. The measures are defined in (7), (6), (4), and (5). The proposed weight function yields the highest percentage accuracy in the matching. The popularity of the websites is shown in Figure 8.

(A1) on the graph. We repeat the experiment by varying the size of the considered subset of popular websites. The result is shown in Figure 9. As expected, as fewer websites are considered, we have less information available for matching; hence the matching accuracy drops. However, by considering merely the top 60 most popular websites, we can still correctly match more than 50% of users. Moreover, as in Table II, the matching accuracy obtained by using the weight function in (7) is consistently higher than that obtained by using any of the other heuristic measures.

C. Experiments on GeoLife (GL) GPS dataset

1) Dataset description and preprocessing: The Geolife (GL) dataset [39] contains the GPS traces of 182 users collected over five years. The user traces in this dataset are represented by a sequence of time-stamped points, each of which contains the information of latitude and longitude. The trajectories are widely distributed over many cities in China and even some in the USA and Europe, but the majority of the data is created in the city of Beijing. In our experiments, we focus on the trajectories collected within the 5th ring road of Beijing, which is an area approximately $39 \text{ km} \times 39 \text{ km}$. We first grid this area

into $100\text{ m} \times 100\text{ m}$ squares. Each square represents a location. Figure 10(a) shows the considered area, where all the locations with a recorded GPS position are darkened. We call a particular one-week period active for a user if she has at least one recorded GPS position during the week. Figure 10(b) shows the active weeks for each user during the data collection campaign. As can be seen from Figure 10(b), the users contributed to the dataset during different periods.

We filtered out all users with number of active weeks equal to 1 and were left with $N = 154$ users. The users have on average 15.4 active weeks of data. We split each user's trajectories into two parts, where part one corresponds to the trajectories recorded in the first half of her active weeks, and part two corresponds to the trajectories recorded in the second half of her active weeks. We construct histograms of the locations visited by each user in each week. Thus, the histogram $\Gamma_{x(i)}$ (respectively, $\Gamma_{y(i)}$) of user i in the first (respectively, second) part gives the relative fractions of recorded GPS positions from

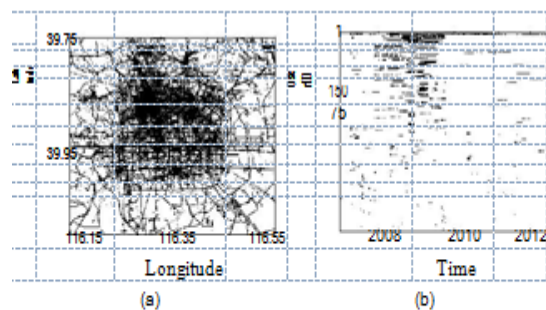


Fig. 10. (a) Gridding of the 5th ring road of Beijing into squares of $100\text{ m} \times 100\text{ m}$. In area has approximate size of $39\text{ Km} \times 39\text{ Km}$. The grids in which a GPS position is recorded for a user is darkened. (b) The active weeks for each user during the data collection campaign.

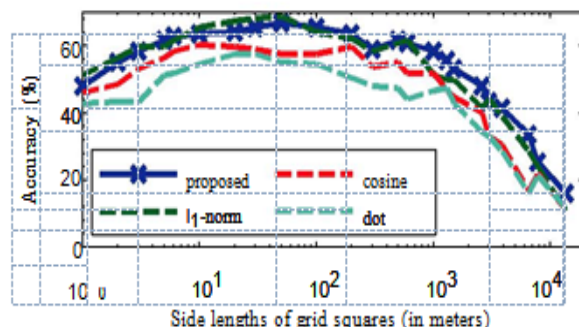


Fig. 11. The evolution of the matching accuracy for the GL dataset ($N = 154$) as a function of the grid side-length by using different metrics. The measures are defined in (7), (6), (4), and (5). The accuracy is maximum for moderate side-lengths.

various locations (i.e., grid squares on the map) in the first (respectively, second) part of her data. The set ψ_1 (respectively, ψ_2) corresponds to the histograms of the number of recorded GPS positions of the N users from the K locations in their first parts (respectively, second parts).

2) Matching accuracy with different metrics: We set the side length of grid squares equal to 1000 m , and we compute the histograms and apply (A1) to the graph G of Figure 2 with different choices of edge weights. The obtained matching accuracy, when the side length of grid squares is 1000 m , is shown in the last row of Table II. The accuracy obtained by using the weight function proposed in (7) is significantly higher than that obtained by using any of the other heuristic measures.

3) Effect of spatial resolution: We repeat the previous experiment with varying choices for the side lengths of grid squares. The resulting matching accuracies are shown in Figure 11 as a function of the side lengths. For very large side-lengths, the spatial resolution is low, hence the users' location traces are easily confused, thus leading to low matching accuracy. For very small side-lengths, there are too many locations in the sense that the inherent noise in the GPS trajectories come into effect, which leads to an over-fitting of the data, and thus the matching accuracy is again low. Therefore, the accuracy is maximum for moderate side-lengths— around 100 m in the figure.

4) Comparison with existing work: In [17] the authors propose a de-anonymization scheme based on a mobility model called the Mobility Markov Chain (MMC) and applied it to the GL dataset. In their approach, an MMC is constructed for each user from her mobility traces observed during the training phase and during the test phase. Distance metrics between MMCs are then used to link a user's trace from the test phase to the corresponding trace in the training phase. There are three main differences between their approach and ours. First, in their approach, the set of locations that a user visits is learned by applying a clustering algorithm to the user's GPS trajectories. The clustering algorithm identifies the accumulation regions of the user's trajectory that is then used to represent the set of locations that the user visits, whereas in our approach, we partition the map area

into squares that represent the set of locations. Second, they use the timing information present in the users' trajectories to learn the MMCs, whereas in our case we disregard all the timing information present in the trajectories and only consider the fraction of visits to different locations. Third, they de-anonymize the users one-by-one, whereas we simultaneously de-anonymize all the users.

In [17], the authors report a de-anonymization accuracy of up to 45% on 77 users in the setting where the regions identified from the clustering algorithm have a maximum radius equal to 500 m. In comparison, our scheme obtains a de-anonymization accuracy of up to 60% for 154 users in the setting where the side lengths of grid squares range from 300 m to 1000 m. If we do one-by-one user de-anonymization, this accuracy drops down to 50%, however it still remains higher than the 45% reported in [17]. We believe that this is because by using a complicated and dynamic model such as MMC, there is a substantial over fitting of the user data to the model. In [17], a $K \times K$ transition probability matrix is fitted to each trace, whereas in our approach a K -length probability vector is fitted. This leads to poorer performances because the model learned from the first dataset does not "generalize" well to the second dataset.

D. Running time

Here we present the timing information of performing the de-anonymization attacks that are given in Table II. We consider only the case where our proposed metric is used. The running times are given for MATLAB version 8.3.0.532 running on a Lenovo Thinkpad T410 equipped with Intel i7 processor with clock speed of 2.67 GHz, with 8 Gb of RAM, and with Microsoft Windows 7 64-bit operating system.

The running time for computing the edge weights (w_{ji} in (7)) of graph G and for running (A1) on G are 41 min and 432 min, respectively, for the CDR dataset. The respective numbers for the WBH dataset are 6 sec and 0.1 sec for computing the edge weights of G and for running (A1) on G , respectively. The respective numbers for the GL dataset are 0.9 sec and 0.2 sec for computing the edge weights of G and for running (A1) on G , respectively. Note that the reported numbers do not include the preprocessing

time, that is, the time required for computing the histograms from the raw data.

V. PRIVACY ENHANCING MECHANISMS

We demonstrated by our experiments in Section IV that applying anonymization to histograms of users' behavior is not effective in protecting the users' identities from an adversary who has access to auxiliary knowledge about the users. In this section, we discuss additional privacy-preserving mechanisms that can be applied to the histograms in order to make it difficult for the adversary to identify the users. These mechanisms essentially make the released histograms closer to each other so that there is greater scope for confusion in distinguishing them from each other, and thus the matching accuracy declines.

A. Basic data coarsening and data suppression

Two popular categories of privacy-preserving mechanisms are data obfuscation and data suppression methods [40]. An example of data coarsening is spatial resolution reduction, which can be achieved by aggregating different locations into one. We investigated the latter in our experiments in Section IV-A6 and in Figure 11. Data suppression is the process of restricting the released data associated with each user. For example, in our experiment in Figure 9 for the WBH dataset, we consider only the subset of popular websites (i.e., websites that are visited by most users) and publish the histograms values associated with this subset. Another example is time-domain restriction, which refers to the process of limiting the time-period over which the histograms are computed. We investigated this approach in our experiment in Figure 7 for the CDR dataset. Another popular privacy-preserving mechanism is k -anonymization, which we investigate in the next subsection.

B. k -Anonymization via micro-aggregation

A released dataset is said to have the k -anonymity property if the data for each user contained in the dataset is identical to the data for at least $k-1$ other users [41]. One mechanism for guaranteeing k -anonymity for a dataset is by means of micro-aggregation [42]. In micro-aggregation, users' data are partitioned into different clusters such that each cluster contains data of at least k users. The average

of the data within each cluster is computed and then used to replace the original data values of all the users within the cluster. These new data values are then released, resulting in a dataset with the k -anonymity property. In micro-aggregation, the partitioning is done by using a criterion of minimum within-cluster information loss, and it has been shown that finding the optimal partitioning is NP-hard [43]. In the following, we define micro-aggregation in mathematical terms, and describe how our matching method can be adapted to de-anonymize micro-aggregated histograms of users' data.

1) Micro-aggregation: Let $\{C_1, C_2, \dots, C_g\}$ be a partitioning of the users U_1 (i.e., the users who generate the histograms ψ_1) into g clusters. That is, $U_1 = \bigcup_{q=1}^g C_q$ and $C_q \cap C_{q'} = \emptyset$ for $q \neq q'$. We later elaborate on the criteria for choosing the set $\{C_q\}_{1 \leq q \leq g}$. Furthermore, define $k = \min_{1 \leq q \leq g} |C_q|$, and

$$\Gamma_{C_q} = \frac{1}{|C_q|} \sum_{x_i \in C_q} \Gamma_{x_i}, \quad (10)$$

for $1 \leq q \leq g$, which represent the average of histograms of all users within each cluster. In micro-aggregation, instead of releasing the set of histograms ψ_1 , the set of micro-aggregated

$$\text{histograms } \psi_1 = \{\Gamma_{x_i} \mid x_i \in U_1\} \text{ is replaced by } \{\Gamma_{C_q} \mid 1 \leq q \leq g\}.$$

when ψ_1 is released, every user in set U_1 is guaranteed k -anonymity.

Although micro-aggregation guarantees k -anonymity to the users, it distorts the released dataset. Specifically, every his-

to gram Γ_{x_i} is replaced by Γ_{C_q} . The criteria for obtaining $\{C_q\}_{1 \leq q \leq g}$ in micro-aggregation is to minimize the total distortion to the data, for a given value of k . In the literature, the l_2 -norm is often used to measure the distortion [44], however because the histograms lie on the probability simplex, we use the l_1 -norm to measure the distortion. In particular the total added distortion, which is also called information loss, is

$$L = \sum_{q=1}^g \sum_{x_i \in C_q} \|\Gamma_{x_i} - \Gamma_{C_q}\|_1.$$

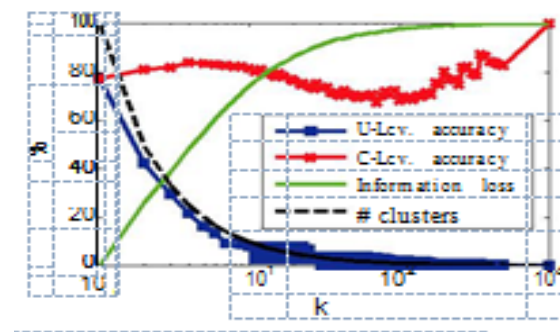
The maximum information loss occurs when all the users are partitioned into a single cluster, i.e., when $g = 1$. The information loss in this case is $L = \sum_{x_i \in U_1} \|\Gamma_{x_i} - \Gamma_{U_1}\|_1$, where $\Gamma_{U_1} = \frac{1}{N} \sum_{x_i \in U_1} \Gamma_{x_i}$. Consistent with the definition of k , we have $k = \min_{1 \leq q \leq g} |C_q|$. The information loss in this case is as follows:

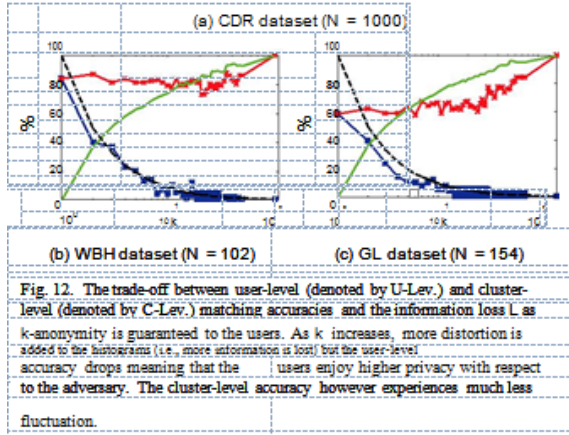
$$L = \sum_{q=1}^g \sum_{x_i \in C_q} \|\Gamma_{x_i} - \Gamma_{C_q}\|_1. \quad (11)$$

The extreme case, $L = 0$, represents the scenario where no micro-aggregation is performed (i.e., $g = N$) and where all users are guaranteed 1-anonymity. The other extreme case, $L = 1$, represents the scenario where $g = 1$ and where all users are guaranteed N -anonymity. For a given value of k , we seek the partitioning $\{C_q\}_{1 \leq q \leq g}$ whose normalized information loss L given in (11) is as small as possible. In our following experiment, we use the algorithm proposed in [44] for performing micro-aggregation, where we adapt the algorithm to measure the distortion by using l_1 -norm.

2) Experimental evaluations: Here we evaluate the effectiveness of the matching algorithm when micro-aggregation is performed on the unlabeled histograms ψ_1 . We consider an adversary who has access to the labeled histograms ψ_2 and is interested in matching these histograms to the micro-aggregated ones in ψ_1 . We consider two different notions of accuracy for the matching. Let the labeled histogram Γ_{y_i} be matched to the unlabeled micro-aggregated histogram Γ_{x_j} . According to our first notion, there is a correct match if $j = \sigma(i)$, where σ is defined in (2). According to our second notion, there is a correct match if $\Gamma_{x_j} = \Gamma_{x_{\sigma(i)}}$. The former notion of accuracy (called user-level) measures the number of correctly matched users and is the same notion that we used in our experiments in Section IV, whereas the latter notion (called cluster-level) measures the number of users whose k -anonymity class (i.e., cluster) is correctly identified.

For the CDR dataset, we consider the setting described in Section IV-A3. In particular, we randomly choose $N = 1000$ out of the 46986 users and construct the sets ψ_1 and ψ_2 . For the WBH dataset, we consider the subset of the top $K = 100$ popular websites and construct the sets of histograms ψ_1 and ψ_2 as described in Section IV-B3. For the GL dataset, we





consider the setting described in Section IV-C2 when grid side-length is set equal to 1000 m.

For each dataset, we perform micro-aggregation with different values of k on the set ψ_1 . We then perform the matching between ψ_1 and ψ_2 by using only the proposed metric of (7). The obtained accuracies are shown in Figure 12(a) and 12(b) and 12(c) for the CDR, WBH, and GL dataset, respectively. The figures also show the normalized information loss L defined in (11) and the normalized number of clusters (i.e., g/N), expressed in percentages. In the extreme case with $k = 1$, no micro-aggregation is performed; therefore, $g = N$, $L = 0$, and the user-level accuracy is equal to the cluster-level accuracy. In the other extreme case, $k = N$, and all the released unlabeled histograms are identical; therefore, the information loss is maximum ($L = 1$), and while the user-level accuracy is minimum, the cluster-level accuracy is maximum. As k increases to about 10, the user-level accuracy dramatically drops, hence the users enjoy an increased level of privacy guarantee, whereas the cluster-level accuracy remains almost the same for all values of k .

VI. CONCLUSION

We have studied the task of identifying users from the statistics of their behavioral patterns. Specifically, given an anonymized dataset in the form of histograms belonging to a set of users and another independent set of histograms generated by the same set of users, we have shown that it is possible to identify the identities of the users in the first dataset to a surprising level of accuracy by matching the statistical characteristics of the users' behaviors across the two datasets. Thus data histograms act as fingerprints for identifying users.

Our proposed solution can be implemented via a minimum-weight maximal matching algorithm on a complete weighted bipartite graph and yields higher accuracy than heuristics-based methods on three different datasets of different nature. We have studied the performance of the algorithm over a wide range of experimental conditions and demonstrated the effect of various factors, such as the number of users, the resolution of the data, the duration of the data collection, and the amount of data suppressed, on the accuracy of the matching algorithm. We have gained the insight that the simultaneous matching of the users yields higher accuracy compared to one-by-one user matching. Furthermore, we have demonstrated the power of simplicity of statistics: Identification based only on data statistics can sometimes result in higher accuracy than existing methods based on more complicated data models. We have further studied the performance of the algorithm under privacy-enhancement techniques, such as k -anonymization, and demonstrated the effect of k on the matching accuracy. Our results suggest that users can be identified, to a surprisingly high level of accuracy, even from the statistics of their behavior. Moreover, using the correct metric and optimal matching algorithm can lead to a significant improvement in matching accuracy over heuristics-based methods. Privacy enhancement via k -anonymization and data obfuscation can reduce identification accuracy, but the accuracy remains non-negligible for moderate levels of data distortion.

VII. ACKNOWLEDGMENTS

We thank Jacques Raguenez and Vincent Blondel for granting us access to the Orange CDR dataset. We thank Alessandra Sala for her feedback on the manuscript.

REFERENCES

- [1] J. Unnikrishnan and F. Movahedi Naini, "De-anonymizing private data by matching statistics," in 51th Annual Allerton Conference, 2013.
- [2] J. Unnikrishnan, "Asymptotically optimal matching of multiple sequences to source distributions and training sequences," IEEE Trans. Inf. Theory, vol. 61, 2015.

- [3] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
- [4] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of IEEE Symposium on Security and Privacy*, 2008.
- [5] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Trans. Knowl. Data Eng.*, 2007.
- [6] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray-Turan, "Web people search via connection analysis," *IEEE Trans. Knowl. Data Eng.*, 2008.
- [7] E. Bengtson and D. Roth, "Understanding the value of features for coreference resolution," in *EMNLP Conference*, 2008.
- [8] A. Stolerman, R. Overdorf, S. Afroz, and R. Greenstadt, "Breaking the closed-world assumption in stylometric authorship attribution," in *Advances in Digital Forensics X*. Springer, 2014, pp. 185–205.
- [9] S. Afroz, A. C. Islam, A. Stolerman, R. Greenstadt, and D. McCoy, "Doppelgänger finder: Taking stylometry to the underground," in *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 2014, pp. 212–226.
- [10] O. Peled, M. Fire, L. Rokach, and Y. Elovici, "Entity matching in online social networks," in *IEEE SocialCom*, 2013.
- [11] J. Liu, F. Zhang, X. Song, Y.-I. Song, C.-Y. Lin, and H.-W. Hon, "What's in a name?: an unsupervised approach to link users across communities," in *ACM WSDM*, 2013.
- [12] L. Sweeney, "Weaving technology and policy together to maintain confidentiality," *The Journal of Law, Medicine & Ethics*, 1997.
- [13] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *ACM MobiCom*, 2011.
- [14] X. Xiao, Y. Zheng, Q. Luo, and X. Xie, "Finding similar users using category-based location history," in *ACM SIGSPATIAL*, 2010.
- [15] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility traces," in *MobiCom*, 2010.
- [16] J. Freudiger, R. Shokri, and J.-P. Hubaux, "Evaluating the privacy risk of location-based services," in *Financial Cryptography and Data Security*. Springer, 2012.
- [17] S. Gams, M.-O. Killijian, and M. Núñez del Prado Cortez, "De-anonymization attack on geolocated data," *Journal of Computer and System Sciences*, 2014.
- [18] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel, "Identification via location-profiling in gsm networks," in *ACM WPES*, 2008.
- [19] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," in *IEEE ICDE*, 2008.
- [20] L. Olejnik, C. Castelluccia, and A. Janc, "On the uniqueness of web browsing history patterns," *annals of telecommunications-Annales des télécommunications*, vol. 69, no. 1-2, pp. 63–74, 2014.
- [21] T.-F. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host fingerprinting and tracking on the web: Privacy and security implications," in *NDSS*, 2012.
- [22] K. Sharad and G. Danezis, "De-anonymizing d4d datasets," in *HotPETs*, 2013.
- [23] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *ACM CCS*, 2012.
- [24] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the Crowd: The privacy bounds of human mobility," *Scientific Reports*, 2013.
- [25] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 2011, pp. 247–262.
- [26] G. Danezis and C. Troncoso, "Vida: How to use bayesian inference to de-anonymize persistent communications," in *Privacy Enhancing Technologies*. Springer, 2009, pp. 56–72.
- [27] C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede, "Perfect matching disclosure attacks," in *Privacy Enhancing Technologies*. Springer, 2008, pp. 2–23.
- [28] P. Pedarsani and M. Grossglauser, "On the privacy of anonymized networks," in *ACM KDD*, 2011.
- [29] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.

- [30] T. M. Cover and J. A. Thomas, Elements of Information Theory 2nd Edition. Wiley-Interscience, 2006.
- [31] L. Ramshaw and R. E. Tarjan, "A weight-scaling algorithm for min-cost imperfect matchings in bipartite graphs," in IEEE FOCS, 2012.
- [32] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," JACM, 1987.
- [33] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver, Combinatorial Optimization, ser. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.
- [34] R. Duan and S. Pettie, "Linear-time approximation for maximum weight matching," J. ACM, vol. 61, no. 1, pp. 1:1–1:23, Jan. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2529989>
- [35] V. D. Blondel, M. Esch, C. Chan, F. Clerot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki, "Data for development: the d4d challenge on mobile phone data," arXiv preprint arXiv:1210.0137, 2012.
- [36] K. L. Huang, S. S. Kanhere, and W. Hu, "Preserving privacy in participatory sensing systems," Computer Communications, 2010.
- [37] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," Journal of Systems and Software, 2011.
- [38] E. Herder, R. Kawase, and G. Papadakis, "Experiences in building the public web history repository," in Proc. of Datatel Workshop, 2011.
- [39] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on gps data," in ACM Ubicomp, 2008.
- [40] C. C. Aggarwal and S. Y. Philip, A general survey of privacy-preserving data mining models and algorithms. Springer, 2008.
- [41] L. Sweeney, "k-anonymity: A model for protecting privacy," INT J UNCERTAIN FUZZ, 2002.
- [42] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," IEEE Trans. Knowl. Data Eng., 2002.
- [43] J. Domingo-Ferrer, F. Seb'è, and A. Solanas, "A polynomial-time ap-proximation to optimal multivariate microaggregation," Computers & Mathematics with Applications, 2008.