Cognitive Computation (2019) 11:241–261
https://doi.org/10.1007/s12559-018-9604-7

A Framework for Privacy Quantification: Measuring the Impact of Privacy Techniques Through Mutual Information, Distance Mapping, and Machine Learning

Yoan Miche[1] · **Wei Ren**[1] · **Ian Oliver**[1] · **Silke Holtmanns**[1] · **Amaury Lendasse**[2]

Received: 1 November 2017 / Accepted: 10 October 2018 / Published online: 12 November 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

In this paper, we propose to investigate how the effects of privacy techniques can be practically assessed in the specific context of data anonymization, and present some possible tools for measuring the effects of such anonymization. We develop an approach using mutual information for measuring the information content in any dataset, including over non-Euclidean data spaces, by means of mapping non-Euclidean distances to a Euclidean space. We further evaluate the proposed approach over toy datasets composed of timestamped GPS traces, and attempt to quantify the information content loss created by three state-of-the-art anonymization approaches. The results allow for an objective quantification of the effects of the k-anonymity and differential privacy algorithms, and illustrate on the toy data used, that such privacy techniques have very non-linear effects on the information content of the data.

Keywords Distance mapping · Non-Euclidean data · Data privacy · Privacy quantification · Mutual information

Introduction

Intelligent data processing is a very specific subset of the cognitive computation field [33, 47] in that it addresses the issue of dealing with data in a manner that is beyond basic numerical processing and uses external knowledge to modify the data for a very specific goal or use. In this sense, many current information systems can greatly benefit from such approaches, both from the point of view of privacy legislation and usability of the data after it has been processed to become legally compliant. The impact of privacy legislation upon the development of information systems and associated processes has been well documented [14, 17, 21, 34, 35, 40, 43], and in order to develop said information systems, the use of anonymization techniques and technologies has very much come to the fore as another element of a system developer's tool box in much the same way as other supporting functionality such as cryptographic functions and networking technologies.

✉ Yoan Miche
yoan.miche@nokia-bell-labs.com

[1] Nokia Bell Labs, Espoo, Finland

[2] Information and Logistics Technology Department, College of Technology, University of Houston, Houston, USA

In practice, there are very few detailed instructions on what privacy techniques should be used in which context, and the application of such privacy techniques to industry datasets is very often ad hoc and likely sub-optimal. Furthermore, recent research fields making use of cognitive-based approaches for interpretation and data mining, such as sentiment analysis [2, 9, 23, 25], e.g., make use of potentially highly sensitive and user-identifying data, making the need for anonymization even higher. In current light of recent regulatory efforts regarding personal data [6], it is clear that plain data mining and data analysis on any kind of data will likely cause legal issues; we feel there is a strong need to address data compliance legal frameworks to allow for the continuation of such research work as sentiment analysis, e.g., while processing the raw data in a manner that allows for sufficient privacy.

Anonymization however is interesting in that while it is almost being mandated [6, 14], very little exists in the way of understanding, as well as quantifying, its effective use. One can see too many examples of poorly anonymized data, e.g., hashing an identification field is not anonymization per se, nor is complete reliance upon recent promising techniques such as differential privacy [11, 12, 38].

The main topic of this paper is to delve into the possibility of quantifying privacy (or lack thereof), by means of information theoretic measures, here. We basically

propose an initial framework (by no means complete nor exhaustive) with some critical properties: (i) a formalization of what it means for a dataset to be legal and/or useful, in the context of privacy; (ii) an information theoretic approach for measuring the usability of a dataset; (iii) an early analysis of the effects of a widely used privacy approach (differential privacy [12, 13]) on the usability of a specific dataset.

This paper is an extension of the previously published work by the authors in [28–30], which was laying down initial ideas about quantifying the data usability (and possibly privacy levels) after data anonymization methods have been used. This extension of the work contains numerous improvements (mathematical and computational) to the original ideas, as well as extensive experimentation to prove empirically the soundness of the proposed approach. As far as the authors know, there is no existing approach proposed to solve this problem, as mutual information (used in this paper as a proxy for data usability) is typically used over random variables in $\mathbb{R}^d$ [4, 18, 22, 36], and not over arbitrary (i.e., non-Euclidean) data. We therefore do not have grounds for comparison, but hope that this research paper will incite such future work.

It is worth noting that we do not have (yet), in this paper, a proposed method for actually "quantifying privacy." One could argue that defining such a notion would be too dataset/situation specific, and that privacy means different things in different contexts. In order to get around this problem, we propose to use existing privacy techniques, as they are, and actually focus only on the usability part of the problem: How useful is the data (for a very specific definition of useful) once it has been anonymized? This is obviously looking at only one side of the problem, but in the absence of a proper means of quantifying privacy, this is a start.

In this paper, we address a number of aspects of anonymization of datasets by examining the effect of varying amounts of differential privacy upon common data structures, specifically, locations and timestamps.

We address this in three ways:

- a framework/structure, or set of rules for anonymization;
- a notion of metricizing usability (after anonymization);
- an analysis of an anonymization function over a given dataset.

In the following, we first describe in the "A Framework for Anonymization" section, at a high level, what it actually means for a dataset to be legal and usable, and how these two criteria can and should be combined. The "Proposed Approach for Measuring Usability of a Dataset" section then provides some notational details, as well as stating the overall problems we are looking to address, in this paper, namely how to measure the information content in a dataset in an objective way, and how this can be done for (technically) any arbitrary dataset, using mutual information. The "Distances over Non-Euclidean Spaces" section first looks into why there is a problem with estimating mutual information over such arbitrary datasets, followed by a solution to this problem, in the "Mapping Solution" section, by mapping the distances to another space. The "Mutual Information Estimation" section then presents the means of estimating the mutual information, using the aforementioned mapping solution. Finally, the "Experimental Results" section presents the experimental results for this approach of quantifying the usability of a specific dataset using mutual information, and analyzes the efficacy (and shortcomings) of this approach.

A Framework for Anonymization

Anonymization can be formulated as the application of a function to a dataset which modifies its information content such that the possibility of recovering or relinking that data so that a unique individual (or a "small" group of individuals) can be re-identified, is lower (or null, ideally).

In this section, we present our framework for anonymization and measurement, in terms of a formalization and the associated notations, along with additional properties we wish to reason over.

Anonymization and Measuring Information Content

We can list the anonymization techniques as the following set of techniques $\mathcal{A}$

$$\mathcal{A} = \{(\epsilon, \delta) - \text{Differential Privacy}, \kappa - \text{anonymization}, \ldots\}. \tag{1}$$

We may then apply an instance α of the functions in $\mathcal{A}$ with suitable set parameters to relevant *structures* inside an input dataset $\mathcal{D}_i$ to produce an anonymized output dataset $\mathcal{D}_o$ as

$$\mathcal{D}_i \xrightarrow{\alpha} \mathcal{D}_o \ , \tag{2}$$

where we assume that $\mathcal{D}_i, \mathcal{D}_o \in \mathcal{D}$, with $\mathcal{D}$ the space of all the possible datasets we are interested into. We therefore do not consider here, for simplicity reasons, the case whereby the input dataset $\mathcal{D}_i$ is transformed through α into a dataset that is no longer in $\mathcal{D}$ (which would for example happen when using an anonymization function α which modifies the structure of the data).

Note that we apply a function α to a structure and not to a field or fields, though in many cases these will be the same. Avoiding using the terminology of fields means that we can avoid confusions seen in cases such as locations which are

represented as a structure of fields ((latitude, longitude), for example) where each field alone is not necessarily meaningful.

In order to state whether an anonymization function is reducing the information content, a measure m can be created—for example an information entropy metric of some form—which maps to a metric space such that the following diagram commutes

$$\begin{array}{ccc} \mathcal{D}_i & \xrightarrow{[\alpha]^+} & \mathcal{D}_o \\ \downarrow m & & \downarrow m \\ \mathcal{M}_i & \xrightarrow{>} & \mathcal{M}_o \end{array} , \tag{3}$$

where $\mathcal{M}_i = m(\mathcal{D}_i)$ and $\mathcal{M}_o = m(\mathcal{D}_o)$ are both values in the image $\mathcal{M}$ of the space $\mathcal{D}$, with $\mathcal{M} = m(\mathcal{D})$, and $[\alpha^+]$ denotes the application of multiple anonymization functions α_i. This space is not necessarily a scalar space such as $\mathbb{R}^d$, but can be any space which the measure m maps to (from $\mathcal{D}$), as long as a means of comparing values exists in this space (in this case, so that we can have a relationship $\mathcal{M}_i > \mathcal{M}_o$, e.g.).

Note that the use of an information theory (entropy in this case) based metric for m is a design choice, here. This relates to the other part of this paper, on actually creating such a metric and estimating it.

In effect, we are, in this paper, making the assumption that the information content of a dataset relates directly to its *usability*: we assume that the common information between pairs of data fields helps quantify how usable a dataset is, in terms of how well the data can be modelled.

In the above, we use $[\alpha]^+$ as a shorthand for multiple applications of anonymization functions where $\alpha_i \in \mathcal{A}$

$$\mathcal{D}_i \xrightarrow{\alpha_1} \mathcal{D}_1 \xrightarrow{\alpha_2} \mathcal{D}_2 \xrightarrow{\alpha_n} \ldots \qquad \mathcal{D}_o \ . \tag{4}$$

Of course one problem here is the selection of a suitable metric m; the property that under anonymization the information content is reduced must hold. The above diagram therefore gives us our first property of anonymization that we wish to hold

$$||\mathcal{M}_i - \mathcal{M}_o|| \leq \varepsilon_{\text{anon}}, \tag{5}$$

that is, that the information content $\mathcal{M}_o$ in dataset $\mathcal{D}_o$ obtained after the application of the anonymization functions $[\alpha]^+$ on dataset $\mathcal{D}_i$ is similar enough (within $\varepsilon_{\text{anon}}$) to the initial information content $\mathcal{M}_i$ on dataset $\mathcal{D}_i$. We denote by $||\cdot||$ a norm over the space $\mathcal{M}$, in this equation.

Our choice of information measurement is relatively free, and in this respect, we have chosen mutual information [7, 16] as the metric due to its relevance in machine learning, and our former attempts to compute it over non-Euclidean data [28–30]. Mutual information quantifies the degree of relationship between random variables, typically taken across a dataset.

Mutual information also has a good set of estimation algorithms specifically for the purposes we require [22, 36]. These are known as estimators, and while varying in quality for certain kinds of data, they provide the measurement properties we require. It will be noted that in some extreme and edge-case situations, these measurements become degenerate but this is a known behaviour and can be accounted for (e.g., the Kraskov estimators do not work well in practice with more than 3 random variables being compared [22]).

Desirable Properties

A dataset can be defined to have a number of desirable properties for use in an information system. We restrict ourselves to two such properties:

- whether the dataset is legal;
- whether the dataset is useful.

Privacy lawyers discuss datasets in terms of whether the dataset is compliant and *sufficiently* anonymized such that it can be used or released. Data scientists would prefer the dataset to be as correct and detailed as possible to increase the accuracy of processing and statistical inference (amongst others).

These two properties can be expressed as characteristic functions over a data set. For example, we can define a function χ_L which states simply whether the dataset is *sufficiently* anonymized and thus compliant (for some legal definition of sufficient and compliant of course!)

$$\begin{array}{ccc} \mathcal{L} & \longrightarrow & \mathbf{1} \\ \downarrow & & \downarrow \\ \mathcal{D} & \xrightarrow{\chi_L} & \{0, 1\} \end{array} , \tag{6}$$

where $\mathcal{L}$ represents a set of legal requirements (which would make a dataset *legal* in this context).

We can define χ_L as below where $\mathcal{D}_i$ is a dataset and ε_L is a value in the range [0, 1] for the information content of $\mathcal{D}_i$.

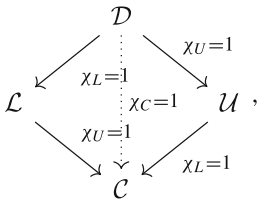
$$\chi_L(\mathcal{D}_o \in \mathcal{D}, \varepsilon_L : \mathbb{R}_{[0,1]} | \mathcal{D}_i) = \begin{cases} \mathbf{1}, & ||\mathcal{M}_i - \mathcal{M}_o|| \geq \varepsilon_L \\ \mathbf{0}, & \text{otherwise} \end{cases} , \tag{7}$$

that is, when the difference in information content $||\mathcal{M}_i - \mathcal{M}_o|| = ||m(\mathcal{D}_o) - m(\mathcal{D}_i)||$ (possibly estimated with mutual information) is above a certain given threshold ε_L, then the dataset is legal, i.e., sufficiently anonymized in that the data has been transformed "enough."

A similar function χ_U can be defined for classifying whether a dataset is usable or not. These datasets are termed $\mathcal{U}$ below.

Given the above we can define a mapping $\chi_C = \chi_L \circ \chi_U$ (the composition of function χ_L with function χ_U, i.e., a

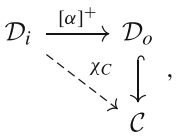
function defining datasets that are both legal and usable) which "extracts" usable *and* legal datasets $\mathcal{C}$



(8)

with an abuse in notations, in that we write in this diagram that $\chi_L(\mathcal{D}) = \mathcal{L}$ and $\chi_U(\mathcal{D}) = \mathcal{U}$, i.e., that the image of the set $\mathcal{D}$ of datasets by the "legal?" functions is the set of legal datasets $\mathcal{L}$ (and similarly for $\mathcal{U}$ and χ_U). We also assume here, for simplicity, that the composition $\chi_C = \chi_L \circ \chi_U = \chi_U \circ \chi_L$ is reversible and leads to the same set $\mathcal{C}$, which is not strictly correct potentially in practice, but avoids over-complicated notations and details in this section.

We can finally summarize this with the following diagram



(9)

which holds when a sequence of anonymization functions $[\alpha]^+$ results in an output dataset $\mathcal{D}_o$ which exists amongst the "set" of all compliant datasets $\mathcal{C}$ (assuming such a set can exist in reality).

In the following section, we put this framework to the test, and start by defining notations, before going into the proposed solution to measuring the usability of data before and after it has been anonymized, in an attempt to quantify the effects of the privacy techniques in terms of the usability of the dataset.

Proposed Approach for Measuring Usability of a Dataset

The main problem with arbitrary datasets (in the sense of the data it holds being totally arbitrary in "form"), is that it typically has to be converted to a Euclidean space [41] (read "put into $\mathbb{R}^d$") to be further used by most machine learning methods [45, 46, 48, 49], for example. Specifically, and since we want to measure the information content in a dataset, in this work, we have the problem of defining what such information content means, in a general, arbitrary dataset, and how to quantify it.

While it is often possible to convert the data into a Euclidean form (by converting the data values into integers/real numbers), this usually not only distorts the data in some sense, but also creates an underlying order, structure, and distance, between the data elements that have been converted. These structures, inherited from the Euclidean space into which the data gets converted, are potentially totally artificial to the original data, and might not be desirable.

In this context, we want, in this work, to minimize the amount of data transformations for our goal of estimating the data information content. For this, we attempt to calculate our information content metric of choice (mutual information), on arbitrary (non-Euclidean) data, in the most computationally efficient way [27], using machine learning. This poses one major problem, as the typical estimators of mutual information [22, 36], as explained in the following, rely on the data being in a Euclidean space, endowed with the Euclidean distance. Thus, in the following, we respectively propose a means to map the distances (and not the data itself) from a non-Euclidean space, to the canonical Euclidean space, and evaluate the effects of this approach on the mutual information estimation.

Figure 1 illustrates the overall proposed methodology to estimate the mutual information between two random variables Data1 and Data2, each in their own (potentially non-Euclidean) metric space. The initial step is to use the structure of the metric space to learn a probabilistic mapping of the distances between elements (in the non-Euclidean space) toward Euclidean distances in an Euclidean space. Once this model has been learned, the distances between elements in the origin non-Euclidean space can be mapped to Euclidean distances in an Euclidean space, and the mutual information can be estimated (as if it was being computed in a Euclidean space only) between the two random variables.

Notations and Definitions

As in the data privacy literature, one traditionally defines a dataset of N records by $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^T$, the matrix of N samples (records) with d attributes $\{\boldsymbol{A}^{(1)}, ..., \boldsymbol{A}^{(d)}\}$. A record $\boldsymbol{x}_l$ is now defined as $\boldsymbol{x}_l = [a_l^{(1)}, a_l^{(2)}, ..., a_l^{(d)}]$, $a_l^{(j)} \in \mathbb{X}^{(j)}$, where $\mathbb{X}^{(j)}$ is the set of all the possible values for a certain attribute $\boldsymbol{A}^{(j)}$. Hence, we can see the vector $[a_1^{(j)}, a_2^{(j)}, ..., a_N^{(j)}]^T \in \mathbb{X}^{(j)}$ as a discrete random variable for a certain attribute $\boldsymbol{A}^{(j)}$ over all the N samples.

Let us consider a metric space $\mathcal{X}^{(j)} = (\mathbb{X}^{(j)}, d^{(j)})$ using the set $\mathbb{X}^{(j)}$ explained above, endowed with the distance function $d^{(j)} : \mathbb{X}^{(j)} \times \mathbb{X}^{(j)} \longrightarrow \mathbb{R}_+$. Generally, $\mathcal{X}^{(j)}$ need not be a Euclidean space.

However, most data mining and machine learning methods rely on the Euclidean distance and its properties, i.e., that of a Minkowski-2 norm, for example, the fact that the distribution of pairwise distances between points drawn in the $\mathbb{R}^d$ unit cube is a Gaussian with predictable mean and variance [15]. One problem with this approach is that it forces the data provider to process the original raw data (before it can be considered legal and useful and therefore,

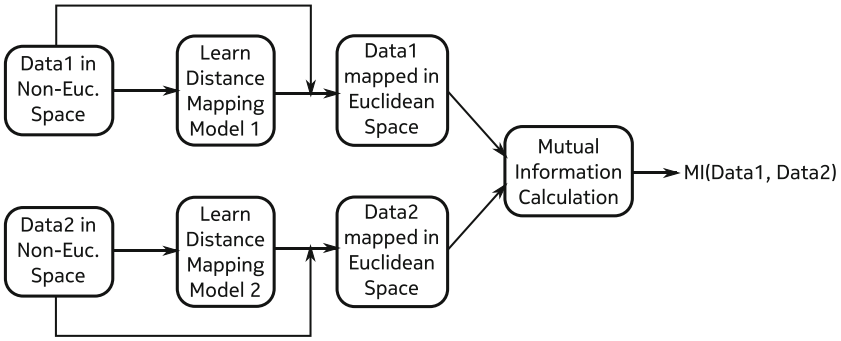


Fig. 1 High-level illustration of the proposed methodology to evaluate mutual information over non-Euclidean data: The input data (Data1 and Data2) is first "analyzed" to learn a mapping between the metric space it lies into, and the canonical Euclidean space $(\mathbb{R}^d, d_{\text{Euclidean}})$. The mapping function is then used to map the pairwise distances between elements in Data1 (resp. Data2), which are needed to evaluate the mutual information between the two datasets

released), in such a way that it is in $\mathbb{R}^d$, while it might not be natural to do so. Thus, we want to verify that in such a case, the distances in a non-Euclidean metric space behave close enough to the Euclidean distances over a Euclidean space (so as to be able to use machine learning/data mining approaches, eventually).

Distances over Non-Euclidean Spaces

Now, we consider two metric spaces $\mathcal{X}^{(i)} = (\mathbb{X}^{(i)}, d^{(i)})$ and $\mathcal{X}^{(j)} = (\mathbb{X}^{(j)}, d^{(j)})$. Let us assume $\mathcal{X}^{(i)}$ to be a canonical Euclidean space with the distance function $d^{(i)}$ the Euclidean norm and $\mathbb{X}^{(i)} = \mathbb{R}^d$, while $\mathcal{X}^{(j)}$ is a non-Euclidean space endowed with a non-Euclidean distance function $d^{(j)}$.

Assume $\boldsymbol{x}^{(j)}$ and $\boldsymbol{y}^{(j)}$ are two discrete independent and identically distributed (iid) random variables for a certain attribute over $\mathbb{X}^{(j)}$. The distances within the metric space $\mathcal{X}^{(j)}$ can then be constructed as another random variable $\boldsymbol{z}^{(j)}$

$$\boldsymbol{z}^{(j)} = d^{(j)}(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j)}), \tag{10}$$

where the values of $\boldsymbol{z}^{(j)}$ are over $\mathbb{R}_+$, and are thus the distances between the respective values in $\boldsymbol{x}^{(j)}$ and $\boldsymbol{y}^{(j)}$.

We denote by $f_{\boldsymbol{z}^{(j)}}(t)$ the probability density function (PDF) of $\boldsymbol{z}^{(j)}$, which describes the pairwise distance distribution over the non-Euclidean metric space $\mathcal{X}^{(j)}$. In the same way, we define $f_{\boldsymbol{z}^{(i)}}(t)$ to be the distribution of pairwise distances over the Euclidean metric space $\mathcal{X}^{(i)}$.

We assume that there exists a mapping G between the distribution of the non-Euclidean distances $f_{\boldsymbol{z}^{(j)}}(t)$ and a new distribution $f^{map}_{\boldsymbol{z}^{(j)}}(t) = G\left(f_{\boldsymbol{z}^{(j)}}(t)\right)$ in such a way that $f^{map}_{\boldsymbol{z}^{(j)}}(t)$ can be as close as possible to the Euclidean distance distribution $f_{\boldsymbol{z}^{(i)}}(t)$

$$||f^{map}_{\boldsymbol{z}^{(j)}} - f_{\boldsymbol{z}^{(i)}}|| \leq \varepsilon, \tag{11}$$

with $\varepsilon > 0$ and the norm $||\cdot||$ is for example an L^2 norm between functions.

As we are using a large number of realizations of the random variables to estimate the distribution $f_{\boldsymbol{z}^{(j)}}(t)$, Eq. 11 is based on the assumption that we can "afford" to draw sufficiently large enough number of realizations of the variables to possibly estimate $f^{map}_{\boldsymbol{z}^{(j)}}$ to be close enough to $f_{\boldsymbol{z}^{(i)}}$.

Distance-mapping algorithms are a useful tool in data applications, for example, data clustering and visualizations. Another good use case for distance-mapping, is mutual information [7, 22, 36], which is used to quantitatively measure the mutual dependence between two (or more) sets of random variables in information theory. Mutual information is most often estimated by constructing the k-nearest neighbors [22, 36] graphs of the underlying data, which thus rely on the Euclidean distances, in a Euclidean space. Hence, there is a strong need to map the distances over a potentially non-Euclidean space. We present the mapping approach in the following "Mapping Solution" section, by solving an integral equation so as to obtain equal probability masses between the pairwise distances probability functions in both spaces (Euclidean and non-Euclidean).

Mapping Solution

We propose to use machine learning (more specifically, universal function approximators [8, 19], and in this case, an extreme learning machines [10, 20, 39])-based method to map the distribution $f_{\boldsymbol{z}^{(j)}}$ of the non-Euclidean distance to the distribution $f_{\boldsymbol{z}^{(i)}}$ of the Euclidean distance, with the fact that most Machine learning techniques are able to fit a continuous input to another different continuous output (at least universal function approximators).

The specific method used in this mapping solution is named OP-ELM [31, 32] (for optimally pruned extreme

learning machine) as it has the advantage of being rather parameter free, since the least useful neurons get automatically pruned out of the model to make certain the model is performing as best as possible.

We then want to make it so that given a certain distance $z = d^{(j)}(x, y)$ obtained over $\mathcal{X}^{(j)}$, we calculate α such that

$$\int_0^z f_{z^{(j)}}(t)dt = \int_0^\alpha f_{z^{(i)}}(t)dt. \tag{12}$$

We want to obtain the α values so that the probability masses of the distances in the non-Euclidean metric space $\mathcal{X}^{(j)}$ and the Euclidean metric space $\mathcal{X}^{(i)}$ are the same. The α value is thus the mapped distance over $\mathcal{X}^{(i)}$. To obtain α, firstly we need to calculate the integral in the left part of Eq. 12 with the given z; secondly, we need to calculate the integral in the right part of Eq. 12 as a function of α; the α value can then be solved with Eq. 12.

The following "Algorithm Used for Distance Mapping" section describes in practice the algorithm used to achieve the proposed distance mapping.

Algorithm Used for Distance Mapping

To calculate the integral in the left part of Eq. 12, we first choose to construct the cumulative distribution function (CDF) $F_{z^{(j)}}$ using OP-ELM for functional estimates. We propose to use the data directly obtained from the cumulative histogram of the random variable $\boldsymbol{z}^{(j)}$ as input and output for the OP-ELM method, i.e., the bin values of the cumulative histogram as inputs, and the associated counts for the output values. The OP-ELM then learns directly the shape of the CDF by effectively interpolating between the bins.

The algorithm for distance mapping is explained as follows:

A.1 Draw as many samples as possible from $\boldsymbol{x}^{(j)}$ and $\boldsymbol{y}^{(j)}$ (random variables over $\mathbb{X}^{(j)}$);
A.2 Compute the random variable $\boldsymbol{z}^{(j)} = d^{(j)}(\boldsymbol{x}^{(j)}, \boldsymbol{y}^{(j)})$;
A.3 Compute the cumulative histogram of $\boldsymbol{z}^{(j)}$;
A.4 Use OP-ELM to learn this cumulative histogram: this creates an un-normalized version (with a max value differing from 1, potentially) of the cumulative distribution function $F_{z^{(j)}}$, with $dF_{z^{(j)}}/dt = f_{z^{(j)}}(t)$;
A.5 Compute the integral of $F_{z^{(j)}}(t)$ over its domain to obtain the normalizing constant $\boldsymbol{C}^{(j)}$;
A.6 Normalize the estimated CDF from **A.4** with the constant $\boldsymbol{C}^{(j)}$;
A.7 This yields a functional representation $G^{(j)}(t)$ of $F_{z^{(j)}}(t)$ as an OP-ELM that behaves as an estimate of the CDF of $\boldsymbol{z}^{(j)}$;
A.8 We can finally evaluate $G^{(j)}(t)$ in z (which was the given distance value) to obtain a value we denote β:

$$\beta = G^{(j)}(z) \approx \int_0^z f_{z^{(j)}}(t)dt, \tag{13}$$

and this is also done numerically;
A.9 We assume the cumulative distribution function (CDF) of the Euclidean distances $\alpha = d^{(i)}(x, y)$ to be $F_{z^{(i)}}(\alpha)$. Solving Eq. 12 now becomes:

$$\beta = G^{(j)}(z) \approx F_{z^{(i)}}(\alpha), \tag{14}$$

$$\alpha = F^{-1}_{z^{(i)}}(\beta), \tag{15}$$

where $F^{-1}_{z^{(i)}}(\beta)$ is the inverse of the CDF in the mapped Euclidean space $\mathcal{X}^{(i)}$. The distances are then mapped from z to α.

Note that this algorithm is independent on the nature of $\mathcal{X}^{(i)}$: at this point, $\mathcal{X}^{(i)}$ can be any metric space.

There are several reasons for using an extreme learning machine-based model for the estimation of the CDF, in this algorithm:

- We are not aware of any means of estimating "directly" the CDF as a continuous (interpolating) function. The existing solutions typically involve the empirical distribution function, therefore relying on linear step functions for the interpolation between the bins of the cumulative histogram.
- While any other universal function approximator can perform just as well this interpolation function (between the cumulative histogram and the continuous CDF estimate), ELMs have the advantage of being extremely fast to compute, with an excellent speed/performance ratio [31].
- We wanted to avoid having to calculate the PDF of the random variable and to further integrate this PDF to obtain the CDF, as this would imply using one extra numerical integration step, which could be prone to numerical instabilities. Therefore, using a kernel density estimation-based method to estimate the PDF to further devise the CDF was to be avoided if possible.
- In some cases, estimating the PDF using any method (whether it be kernel density estimation or another), is a challenging task when the PDF has an unexpected shape (for example, in the beta distribution case, or even the uniform case): devising the proper kernel for the density estimation is a possibly difficult and manual task, and we require that this whole algorithm be as automated and parameter free as possible.

For all these reasons, we have chosen to use a simple, fast, and sufficiently efficient technique to estimate the CDF directly, based on extreme learning machines. The details of the number of neurons and the overall structure of the model can be found in the "Experimental Results" section.

In the following, we look at the two possibilities of mapping a non-Euclidean space to a Euclidean space, or to another, non-Euclidean space (for completeness sake).

So, we are presented with two possibilities:

- $\mathcal{X}^{(i)} = (\mathbb{X}^{(i)}, d^{(i)})$ is the canonical Euclidean space, i.e., $\mathbb{X}^{(i)} = \mathbb{R}$ and $d^{(i)}$ is the Euclidean distance over $\mathbb{R}$, as described so far;
- $\mathcal{X}^{(i)}$ is not canonical Euclidean space, with the set of all values $\mathbb{X}^{(i)}$ does not have to be $\mathbb{R}$, while $d^{(i)}$ is also not necessarily the Euclidean distance.

First Case In the case of $\mathcal{X}^{(i)}$ being the canonical Euclidean space, we can find analytical expressions for $f_{z^{(i)}}$ in Eq. 12, by making assumptions on how the variables $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ are distributed. If such assumptions are not acceptable for some reason, it is always possible to revert to the estimation approach mentioned above, or possibly solve analytically as below for other well-known distributions.

(a) *If $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ are normally distributed*

We assume that $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ follow a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and variance σ^2. $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ are iid.

It is then clear that $z^{(i)} = d(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) = ||\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(i)}||_2$ is distributed as a folded normal distribution of mean 0 and variance $2\sigma^2$: $z^{(i)} \sim \mathcal{N}_f(0, 2\sigma^2)$. The probability density function of $z^{(i)}$ can then be described as

$$f_{z^{(i)}}(t) = \frac{1}{\sigma\sqrt{\pi}} e^{-t^2/(4\sigma^2)}, \quad \text{for } t \geq 0. \tag{16}$$

Its CDF $F_{z^{(i)}}(\alpha)$ follows as

$$F_{z^{(i)}}(\alpha) = \int_0^{\alpha} f_{z^{(i)}}(t)dt = \text{erf}\left(\frac{\alpha}{2\sigma}\right), \tag{17}$$

where erf$(\cdot)$ is the error function [1].

Finally, as we have calculated β in Eq. 14, we can solve easily

$$\alpha = 2\sigma \text{erf}^{-1}(\beta). \tag{18}$$

(b) *If $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ are uniformly distributed*

If we assume that $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$ follow a uniform distribution $\mathcal{U}(a, b)$, with $(a \leq b)$, and are iid. The probability distribution $f_{z^{(i)}}(t)$ of the distances $z^{(i)}$ is then obtained as

$$f_{z^{(i)}}(t) = \begin{cases} \frac{2}{b-a}\left(1 - \frac{t}{b-a}\right) & \text{if } 0 \leq t \leq b-a \\ 0 & \text{elsewhere} \end{cases}, \tag{19}$$

which means that

$$F_{z^{(i)}}(\alpha) = \int_0^{\alpha} f_{z^{(i)}}(t)dt = -\frac{1}{(b-a)^2}\alpha^2 + \frac{2}{b-a}\alpha. \tag{20}$$

Solving as before, we finally have that

$$\alpha = (b-a)(1 - \sqrt{1-\beta}), \tag{21}$$

given the fact that the other solution is not acceptable for our case (negative result for a distance value).

(c) *If the distances $||\boldsymbol{x}^{(i)}$-$\boldsymbol{y}^{(i)}||$ are Rayleigh distributed*

A Rayleigh distribution often arises when the metric space is analyzed by the magnitude of the orthogonal two-dimensional vector components. If we assume the distances $z^{(i)} = ||\boldsymbol{x}^{(i)} - \boldsymbol{y}^{(i)}||_2$ follow a Rayleigh distribution with the scale parameter σ, then

$$f_{z^{(i)}}(t) = \frac{t}{\sigma^2} e^{-t^2/(2\sigma^2)}, \quad \text{for } t \geq 0. \tag{22}$$

The cumulative density then becomes

$$F_{z^{(i)}}(\alpha) = \int_0^{\alpha} f_{z^{(i)}}(t)dt = 1 - e^{-\alpha^2/(2\sigma^2)}. \tag{23}$$

The mapped distances α can then be solved by

$$\alpha = \sigma\sqrt{-2\log(1-\beta)}. \tag{24}$$

(d) *Other distributions*

We have discussed above the most common distributions of the distances $z^{(i)}$ in the canonical Euclidean space $\mathcal{X}^{(i)}$. Certainly, the PDF of the distances $z^{(i)}$ can exist in other less common forms in certain specific circumstances. However, for the datasets used in practice with this work, which consist of timestamped GPS coordinates in the form of latitudes and longitudes, the discussed typical distributions are sufficient enough to illustrate the mapped distributions in the canonical Euclidean space. We discuss this in the "Experimental Results" section.

Second Case In the second case, where $\mathcal{X}^{(i)}$ is not the canonical Euclidean space, we basically have to perform the same estimate of $f_{z^{(i)}}$ and its integral from 0 up to α, as we did for $f_{z^{(j)}}$. The result is another function $G^{(i)}(\alpha)$:

$$G^{(i)}(\alpha) = \int_0^{\alpha} g^{(i)}(t)dt \approx \int_0^{\alpha} f_{z^{(i)}}(t)dt. \tag{25}$$

We then have to solve numerically

$$G^{(i)}(\alpha) = \beta \tag{26}$$

for α.

Implementation Details and Technical Improvements

When implementing the initial mapping solution, the algorithm spent most of the CPU time on calculating the integral of $f_{z^{(j)}}(t)$ over the distances $z^{(j)}$ numerically as in Eq. 13. This consumes lots of computational time, because the number of pairwise distances $z^{(j)}$ is $N(N-1)/2$, which can obviously grow to a very large value when the data size N increases. Thus, we avoided the integration calculations by using OP-ELM to learn directly the CDF $F_{z^{(j)}}$, instead

of learning the PDF $f_{z^{(j)}}$, as described in **A.4**. This yields a functional representation of $F_{z^{(j)}}(t)$ (with the normalization constant directly from $F_{z^{(j)}}$). β can then be obtained straight from $F_{z^{(j)}}(z)$.

The second most CPU-consuming step in this algorithm is to find the most suitable distribution for $F_{z^{(i)}}(t)$ in the Euclidean space $\mathcal{X}^{(i)}$. To choose whether the non-Euclidean distribution should best be mapped to a normal, uniform, Rayleigh, or other distribution, we fit the $F_{z^{(j)}}(t)$ to well - defined canonical Euclidean distance distributions and find the optimal parameters for the best suitable distribution with the least errors.

Again, if we use the pairwise distances $z^{(j)}$ in $F_{z^{(j)}}$ directly, the fitting computation is very heavy as we are trying to fit the data with $N(N-1)/2$ points. To make it easier, we use the functional representation of $F_{z^{(j)}}(t)$ with the user-defined distances in the pre-defined domain, with the purpose only to find the best distribution and its parameters (the functional presentation of $F_{z^{(i)}}(t)$). Then, the mapped distance α can be obtained from Eq. 15 with the calculated β and the inverse functional representation of $F_{z^{(i)}}(t)$.

In all instances of the algorithm where learning an OP-ELM model was required, we have used 100 neurons with a hyperbolic tangent as the activation function, in addition to having "linear neurons" (the details of how this is achieved can be found in the original article [31]). This is in line with the overall idea of having as little parameters for this methodology as possible, as the OP-ELM basically prunes the neurons that are not strictly needed, in an effort to avoid overfitting the data.

In the following "Distance Mapping Test" section, we present results over the data used for this work, timestamped GPS traces (latitude and longitude).

Distance Mapping Test

We have tested the proposed mapping algorithm by mapping the pairwise distances in the dataset of GPS coordinates in the form of latitudes and longitudes, which is illustrated in Fig. 5 as the route for id_4. Assume we have a dataset $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^T$ to depict the trajectory of one specific person, where the attributes of each record $\boldsymbol{x}_i$ explain the locations by latitude and longitudes coordinate at the corresponding time $\boldsymbol{t}_i$.

Note that the metric space of the GPS coordinates $\mathcal{X}^{(gps)} = (\mathbb{X}^{(gps)}, d^{(gps)})$ is a non-Euclidean space, because the distance $d^{(gps)}$ of two GPS coordinates

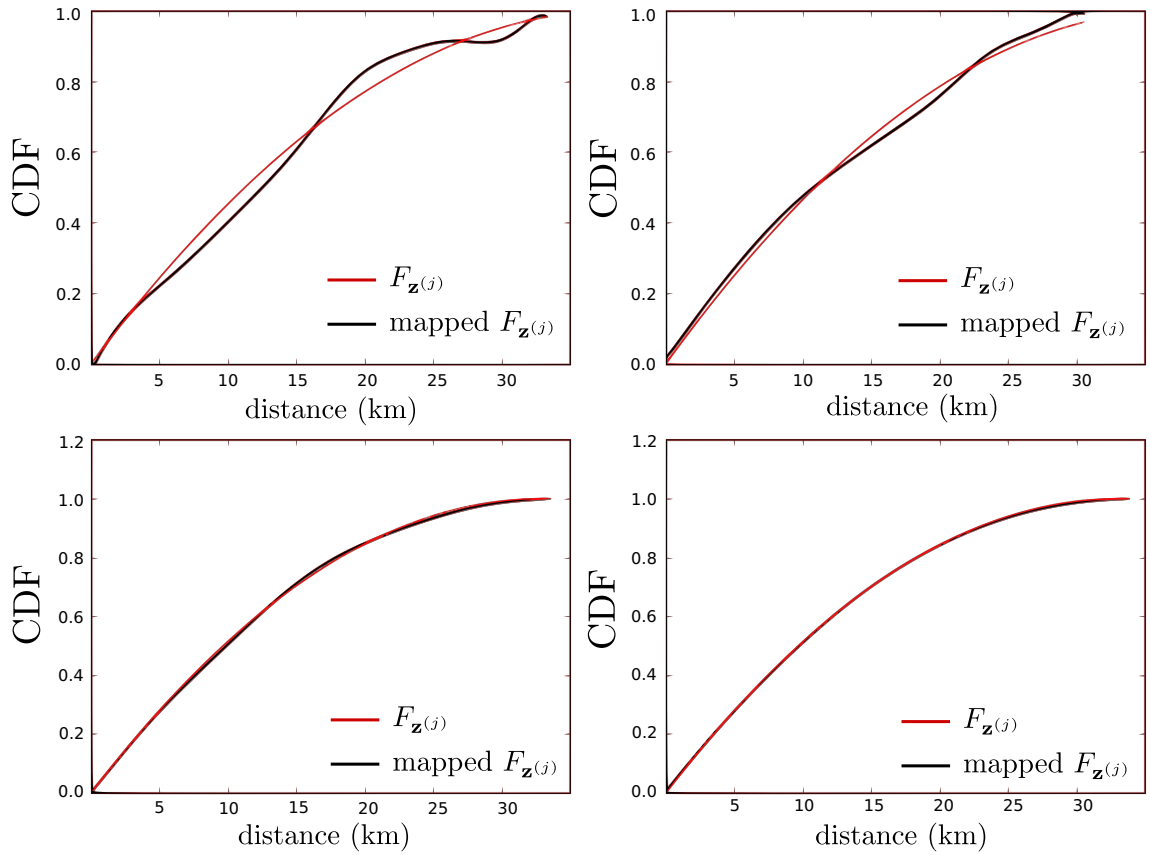


Fig. 2 CDF $F_{z^{(j)}}$ of the pairwise distances in the experimental dataset, and its corresponding mapped CDF $F_{z^{(i)}}$, with $N = 10, 30, 100, 1000$

($\boldsymbol{lat}, \boldsymbol{lon}$) is the shortest route between the two points on the Earth's surface, namely, a segment of a great circle.

We first explore the limit condition on the number of records N in Eq. 11, in that N needs to be sufficiently large to estimate $F_{z^{(j)}}$ to be close enough to $F_{z^{(i)}}$. We test on experimental toy datasets with various $N = 10, 30, 100, 1000$, within which each location record is randomly chosen along the introduced trajectory in Fig. 5 for the route of id_4.

Figure 2 depicts the comparison of the CDF $F_{z^{(j)}}(t)$ of the pairwise distances obtained from $\mathcal{X}^{(\boldsymbol{gps})} = (\mathbb{X}^{(\boldsymbol{gps})}, d^{(\boldsymbol{gps})})$, and the CDF $F_{z^{(i)}}(t)$ of the mapped distances in the Euclidean space, with $N = 10, 30, 100, 1000$ for the four subplots respectively.

One can see that in this specific simple case, with small N values of 10 and 30, there exist discrepancies between the CDF in $\mathcal{X}^{(\boldsymbol{gps})}$ and in the mapped Euclidean space. Meanwhile, with larger N values of 100 and 1000, the convergence on N is better satisfied.

Thus, we can see that a number of records $N = 100$ is sufficient to closely estimate the distribution $F_{z^{(j)}}(t)$ to $F_{z^{(i)}}(t)$ in this very simple case. The standard errors (SEs) of the mapped distribution $F_{z^{(i)}}$ are calculated, meanwhile selecting N values from 5 to 5000, along the specific route. Figure 3 shows the SEs of the mapped CDF with the dependence on N. The SE of the mapped $F_{z^{(i)}}(t)$ reaches values close enough to 0 already at $N \simeq 50$.

Using ELM to Learn the Functional Distribution

We propose to use an extreme learning machine (ELM) [19, 20]-based method as the mapping tool between distance functions. The reason for choosing this specific machine learning technique is its excellent performance/computational time ratio. The model is simple and involves a minimal amount of computations. Since we are dealing with the limit problem of the number of records N to estimate the distribution $F_{z^{(j)}}(t)$ (in Eq. 11), the ELM -based model is applicable in that it can learn the mapping in reasonable time for large amounts of data, if such a need arises. ELM is a universal function approximator, which can fit any continuous function.

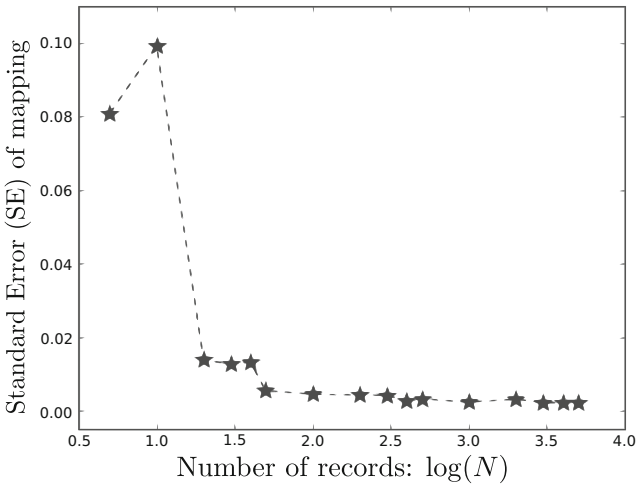


Fig. 3 Standard errors of the mapped distributions $F_{z^{(i)}}$ against the number of records N

The ELM algorithm was originally proposed by Guang-Bin Huang et al. in [20], and further developed, e.g., in [31, 32, 44], and analyzed in [5]. It uses the structure of a single-layer feedforward neural network (SLFN) [3]. The main concept behind the ELM approach is its random initialization, instead of a computationally costly procedure of training the hidden layer. The output weight matrix is then to be found between the hidden representation of the inputs and the outputs.

It works as following: Consider a set of N distinct observations $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, with $\boldsymbol{x}_i \in \mathbb{R}^d$, $\boldsymbol{y}_i \in \mathbb{R}^c$, and $i = 1, ..., N$. In the case the SLFN would perfectly approximate the data, with the errors between the estimated outputs $\hat{\boldsymbol{y}}_i$ and the actual outputs $\boldsymbol{y}_i$ being zeros, i.e., $\hat{\boldsymbol{y}}_i = \boldsymbol{y}_i$, the relation between inputs, weights, and outputs is then

$$\sum_{j=1}^{n} \beta_j \phi(\boldsymbol{w}_j \boldsymbol{x}_i + b_j) = \boldsymbol{y}_i, \quad i \in [1, N], \tag{27}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^c$ is the activation function of the hidden neurons; $\boldsymbol{w}_j$ are the input weights; b_j are the biases; and β_j are the output weights. Equation 27 can also be written compactly as

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{Y}, \tag{28}$$

with $\boldsymbol{\beta} = (\beta_1^T, ..., \beta_n^T)^T$, and $\boldsymbol{Y} = (\boldsymbol{y}_1^T, ..., \boldsymbol{y}_n^T)^T$.

The output weights $\boldsymbol{\beta}$ can be solved from the hidden layer representation of inputs $\boldsymbol{H}$ and the actual outputs $\boldsymbol{Y}$

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{Y}, \tag{29}$$

where $\boldsymbol{H}^{\dagger}$ is the Moore-Penrose generalized inverse [37] of the matrix $\boldsymbol{H}$.

The ELM training does not require iterations, so the most computationally costly part is the calculation of a pseudo-inverse of the matrix $\boldsymbol{H}$. This makes ELM an extremely fast machine learning method. Thus, we propose to use an ELM-based approach to learn the cumulative distribution of the pairwise distances over non-Euclidean spaces $F_{z^{(j)}}(t)$.

In the following section, we look at the mutual information estimation approach (and how the estimators actually work) and investigate the effects of the proposed distance mapping algorithm on the estimation, as well as the convergence rates of the estimators with respect to the size of the dataset used.

Mutual Information Estimation

In information theory, mutual information (MI) [7, 16] between two (or more) random variables is a measure of the mutual dependence between the two variables. It quantifies the amount of information that the two random variables share. Intuitively, MI measures how much knowing one random variable reduces the uncertainty about the other one. MI can be interpreted to be intricately linked to the concept of entropy of a random variable, which defines the amount of information held in the random variable.

Assume we have two bi-variate sets of random variables $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$, with the measurements to be $\{x_k^{(i)}\}$ and $\{x_k^{(j)}\}$, respectively, where $k = 1, ..., N$. Formally, the MI between $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ is then defined as:

$$I(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \int_{\boldsymbol{x}^{(j)}} \int_{\boldsymbol{x}^{(i)}} dx^{(j)} dx^{(i)} \; p(x^{(i)}, x^{(j)}) \log\left(\frac{p(x^{(i)}, x^{(j)})}{p(x^{(i)}) p(x^{(j)})}\right), \tag{30}$$

where $p(x^{(i)})$ and $p(x^{(j)})$ are the marginal probability distributions of $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$, respectively; $p(x^{(i)}, x^{(j)})$ is the joint probability distribution function of $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$.

The unit of information measured by MI is determined by the base of the logarithm in Eq. 30, for example, base two leads the units to be bits. If $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ are strictly independent, $I(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$ is zero: to clarify, $\boldsymbol{x}^{(i)}$ does not give any information about $\boldsymbol{x}^{(j)}$, and vice versa. In another extreme case, if $\boldsymbol{x}^{(i)}$ is a deterministic function of $\boldsymbol{x}^{(j)}$, then all information conveyed by $\boldsymbol{x}^{(j)}$ is shared with $\boldsymbol{x}^{(i)}$. The MI in this case is the same as the entropy of $\boldsymbol{x}^{(i)}$ and as the entropy of $\boldsymbol{x}^{(j)}$.

When compared with the covariance of $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$, MI contains information about both linear and nonlinear dependence of $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ by measuring the weighted sum of the joint probabilities, while the covariance contains the linear dependence only by creating the weighted sum of the product of $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$.

Since the marginal densities $p(x^{(i)})$ and $p(x^{(j)})$ as well as the joint $p(x^{(i)}, x^{(j)})$ density are usually unknown, estimating MI is not always easy. The most straightforward approach is by partitioning the marginals $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ into bins of finite size, and then calculate the approximation of MI by the finite sum of the bins. However, this approach fails when the densities are not properly distributed and the obtained MI approximation shows a high-resolution dependence on the finite bin size.

In this paper, we resort to MI estimators using k-nearest neighbor (knn) statistics, as proposed in Kraskov's original work [22]. Two estimators are proposed in the original paper, which are shortly explained as follows.

Kraskov's First Estimator

With the previous notations, the first estimator $I^{(1)}$ for MI is defined as

$$I^{(1)}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \psi(k) - \langle \psi(\boldsymbol{n}_{\boldsymbol{x}^{(i)}}+1) + \psi(\boldsymbol{n}_{\boldsymbol{x}^{(j)}}+1) \rangle + \psi(N). \tag{31}$$

Here, $\psi(x)$ is the digamma function; k is the number of neighbors in the knn algorithm (to be decided by the user); $\boldsymbol{n}_{\boldsymbol{x}^{(i)}}$ and $\boldsymbol{n}_{\boldsymbol{x}^{(j)}}$ are the vectors with each element holding the count of neighbors in a certain ball for each measurement, in the subspaces $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$, respectively. The elements of $\boldsymbol{n}_{\boldsymbol{x}^{(i)}}(m)$ and $\boldsymbol{n}_{\boldsymbol{x}^{(j)}}(m)$, with $m = 1, ..., N$, are defined as

$$\boldsymbol{n}_{\boldsymbol{x}^{(i)}}(m) = \text{Card}\left(\left\{x_l^{(i)} \in \boldsymbol{x}^{(i)} : d^{(i)}(x_l^{(i)} - x_m^{(i)}) \leq \varepsilon_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}(m)/2\right\}\right), \tag{32}$$

and

$$\boldsymbol{n}_{\boldsymbol{x}^{(j)}}(m) = \text{Card}\left(\left\{x_l^{(j)} \in \boldsymbol{x}^{(j)} : d^{(j)}(x_l^{(j)} - x_m^{(j)}) \leq \varepsilon_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}(m)/2\right\}\right). \tag{33}$$

Here, $\varepsilon_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}(m)/2$ is the distance between the sample m and its k-th nearest neighbor $knn(m)$ in the joint space $z = (\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$:

$$\varepsilon_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}(m)/2 = \|z_m - z_{knn(m)}\| = \max\left\{\|x_m^{(i)} - x_{knn(m)}^{(i)}\|, \|x_m^{(j)} - x_{knn(m)}^{(j)}\|\right\}. \tag{34}$$

Note here that it is therefore paramount to a proper estimation to have "similar" ranges and behaviors for the norms $||x_m^{(i)} - x_{knn(m)}^{(i)}||$ and $||x_m^{(j)} - x_{knn(m)}^{(j)}||$. If there are strong discrepancies, the ε value will be biased constantly.

Kraskov's Second Estimator

In the second algorithm, instead of using the joint-space distance $\varepsilon_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}(m)/2$, we denote by $\varepsilon_{(\boldsymbol{x}^{(i)})}(m)/2$ and $\varepsilon_{(\boldsymbol{x}^{(j)})}(m)/2$ the distances between the same points projected into the $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ subspaces. To clarify

$$\varepsilon_{(\boldsymbol{x}^{(i)})}(m)/2 = \|x_m^{(i)} - x_{knn(m)}^{(i)}\|, \quad \varepsilon_{(\boldsymbol{x}^{(j)})}(m)/2 = \|x_m^{(j)} - x_{knn(m)}^{(j)}\|. \tag{35}$$

The counting vectors of $\boldsymbol{n}_{\boldsymbol{x}^{(i)}}$ and $\boldsymbol{n}_{\boldsymbol{x}^{(j)}}$ are redefined by changing $\varepsilon_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}(m)/2$ to $\varepsilon_{(\boldsymbol{x}^{(i)})}(m)/2$ in Eq. 32, and changing $\varepsilon_{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})}(m)/2$ to $\varepsilon_{(\boldsymbol{x}^{(j)})}(m)/2$ in Eq. 33.

The second estimator $I^{(2)}$ for MI is then:

$$I^{(2)}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \psi(k) - 1/k - \langle \psi(\boldsymbol{n}_{\boldsymbol{x}^{(i)}}) + \psi(\boldsymbol{n}_{\boldsymbol{x}^{(j)}}) \rangle + \psi(N). \tag{36}$$

The same note can be made here, about the fact that distances have to be "comparable," between the spaces, to have a proper estimation on these quantities.

It is worth mentioning that the distances used in the Kraskov estimators are typically defined and computable for a Euclidean space, with the traditional Euclidean distance function. However, in the case of non-Euclidean subspaces with arbitrary distances, this problem can be solved according to the distance mapping mechanism as discussed in the "Mapping Solution" section.

Implementation of Kraskov's Estimators

Here, we show the result of mutual information calculation for $I^{(1)}$ and $I^{(2)}$, for Gaussian distributions. Assuming $\boldsymbol{X}$ and $\boldsymbol{Y}$ are Gaussian distributed random variables with 0 mean and 1 variance, and with covariance $r = 0.9$, the theoretical mutual information $I(\boldsymbol{X}, \boldsymbol{Y})$ is known exactly as

$$I_{\text{Gaussian}}(\boldsymbol{X}, \boldsymbol{Y}) = -\frac{1}{2}\log(1 - r^2). \tag{37}$$

In this case, for $r = 0.9$, $I(\boldsymbol{X}, \boldsymbol{Y}) = 0.830366$.

Figure 4 shows the results of $I^{(1)}(\boldsymbol{X}, \boldsymbol{Y})$ and $I^{(2)}(\boldsymbol{X}, \boldsymbol{Y})$ calculated from Eqs. 31 and 36, respectively. $I^{(1)}$ and $I^{(2)}$ values are compared with various N values, and using $k = 1$ for the k-th nearest neighbor. We can see that the $I^{(1)}$ and $I^{(2)}$ estimators converge to the theoretical value 0.830366 when N increases to large enough values: $N \geq 100$. The error bars become smaller when N is larger, due to the diminishing effects of statistical errors with increasing N in the estimators.

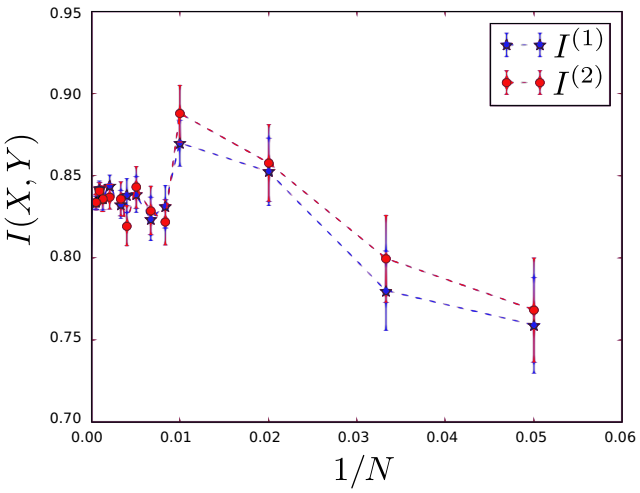


Fig. 4 The results of $I^{(1)}(\boldsymbol{X}, \boldsymbol{Y})$ and $I^{(2)}(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are Gaussian distributed random variables with 0 mean and 1 variance, and with covariance $r = 0.9$. $k = 1$ is used in both estimators

Experimental Results

In this section, we test the previously proposed approach on a practical and real use case. We here used the mutual information estimation approach presented in the previous section to estimate the shared information content between two non-Euclidean random variables, and look at the effects of the state of the art privacy techniques on the data through this proposed metric.

We first present the dataset used in the following experiments, and then look at the effects of (i) hashing data fields; (ii) κ-anonymity; and (iii) two flavors of differential privacy. The effects of these privacy techniques are investigated, as described throughout this paper, in the light of the mutual information computed over the dataset.

Original Routes

Assume we have a dataset $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N]^T$ to depict the trajectory of one specific person, where the attributes of each record $\boldsymbol{x}_i$ explain the location $\boldsymbol{gps}_i$ at the corresponding time $\boldsymbol{t}_i$ for this specific person. The locations are represented in GPS coordinates ($\boldsymbol{gps}$) with latitudes ($\boldsymbol{lat}$) and longitudes ($\boldsymbol{lon}$). Each record $\boldsymbol{x}_i$ can then be explained by: $\boldsymbol{x}_i = (\boldsymbol{gps}_i, \boldsymbol{t}_i) = ((\boldsymbol{lat}_i, \boldsymbol{lon}_i), \boldsymbol{t}_i)$. Hence, what we define here as the mutual information $\boldsymbol{I(X)}$ of the dataset $\boldsymbol{X}$, is a $d \times d$ matrix (in this case $d = 2$: the number of attributes) with the elements holding the mutual information values of the pairwise attributes, illustrated by

$$\begin{aligned}\boldsymbol{I(X)} &= \{I(\boldsymbol{x}^i, \boldsymbol{x}^j)\}_{1 \leq i,j \leq d} \\ &= \begin{bmatrix} I(\boldsymbol{gps}, \boldsymbol{gps}) & I(\boldsymbol{gps}, \boldsymbol{t}) \\ I(\boldsymbol{t}, \boldsymbol{gps}) & I(\boldsymbol{t}, \boldsymbol{t}) \end{bmatrix}\end{aligned} \tag{38}$$

Note that the metric space of the GPS coordinates $\mathcal{X}^{(\boldsymbol{gps})} = (\mathbb{X}^{(\boldsymbol{gps})}, d^{(\boldsymbol{gps})})$ is a non-Euclidean space, because the distance of two GPS coordinates $(\boldsymbol{lat}, \boldsymbol{lon})$ is the shortest route between the two points on the Earth's surface, namely, a segment of a great circle, which is not a Euclidean distance. Meanwhile, the metric space of time $\mathcal{X}^{(\boldsymbol{t})} = (\mathbb{X}^{(\boldsymbol{t})}, d^{(\boldsymbol{t})})$ is here a Euclidean space with a typical Euclidean distance function.

We illustrate the mutual information matrices $\boldsymbol{I(X)}$ by introducing five experimental datasets, with each dataset recording the trajectory for one person. For each person, 100 timestamps and the corresponding $\boldsymbol{gps}$ locations are recorded, where the locations are measured at uniform timestamps. The trajectories in the datasets are shown in Fig. 5.

Table 1 shows the mutual information (MI) matrices of the five experimental ids, respectively. Here, we use $I^{(1)}$ and $I^{(2)}$ to represent the values of MI calculated from the first and second Kraskov estimators, respectively.

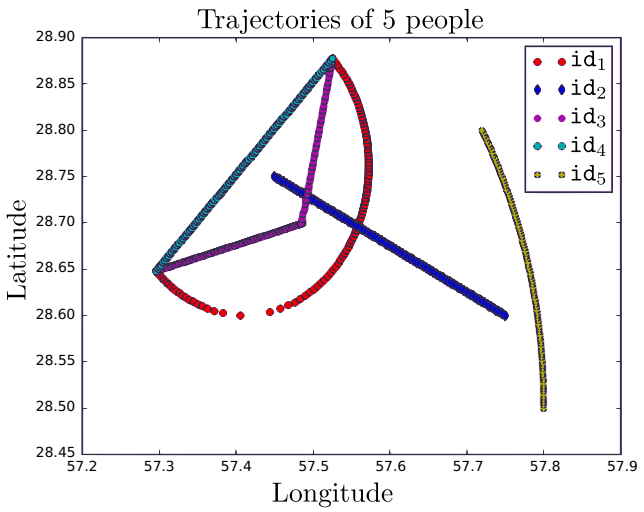


Fig. 5 Five trajectories from the five experimental datasets, respectively

We can see that the MI element value of two identical attributes stays constant, regardless of the variables of the attribute itself: $I^{(1)}(\boldsymbol{gps}, \boldsymbol{gps}) = I^{(1)}(\boldsymbol{t}, \boldsymbol{t}) = 5.18$, and $I^{(2)}(\boldsymbol{gps}, \boldsymbol{gps}) = I^{(2)}(\boldsymbol{t}, \boldsymbol{t}) = 4.18$, illustrating empirically that the distance mapping approach used in this paper is experimentally sound.

Meanwhile, the MI matrices are symmetric with $I(\boldsymbol{gps}, \boldsymbol{t}) = I(\boldsymbol{t}, \boldsymbol{gps})$ for both estimators. The MI values of non-identical pairwise attributes (e.g., $I(\boldsymbol{gps}, \boldsymbol{t})$) are found to be relatively smaller than those values of two identical attributes (e.g., $I(\boldsymbol{gps}, \boldsymbol{gps})$), with the obvious reason that the two identical sets of variables are more mutually dependent than two different variables sets.

The values of $I(\boldsymbol{gps}, \boldsymbol{t})$ are calculated to be in the ranges of 3.65–3.69 and 2.60–3.20 for $I^{(1)}$ and $I^{(2)}$, respectively, compared with the $I(\boldsymbol{gps}, \boldsymbol{gps})$ values of 5.18 and 4.18 for the two estimators. We can see that $I^{(2)}$ is more sensitive than $I^{(1)}$ for ids with different trajectories, by giving disparate $I^{(2)}(\boldsymbol{gps}, \boldsymbol{t})$ values. For example, the $I^{(2)}(\boldsymbol{gps}, \boldsymbol{t})$ of id_1 with the value of 3.20 is larger than those for id_2, id_3, and id_4, with values around 2.7. This is mainly due to the relatively more peculiar trajectory of id_1.

Degeneracy of MI

It is clear from Table 1 that all MI values calculated from $I^{(1)}$ are relatively larger than those from $I^{(2)}$. In principle, both estimators should give very similar results. The difference here is because the number of records with $N = 100$ in each dataset is so small that in the estimators, the quantities $n_x(i)$ and $n_y(i)$ tend to be also very small with considerably large relative fluctuations. This will cause large statistical errors. We discuss here about the MI degeneracy (convergence) of both estimators with increasing numbers of records N.

We take the trajectory of id_4 for example to explain the MI degeneracy. In the original dataset, there are 100 uniform timestamps and the corresponding 100 uniform locations. We increase the number of records N to 200, 300, 400, ..., 2000, by interpolating uniformly denser timestamps and locations into the trajectory. $I^{(1)}(\boldsymbol{gps}, \boldsymbol{t})$ and $I^{(2)}(\boldsymbol{gps}, \boldsymbol{t})$ are then calculated with the ratio of k/N kept to be 0.01 in the estimators.

The dependence of $I(\boldsymbol{gps}, \boldsymbol{t})$ values over number of record N is illustrated in Fig. 6. It can be seen that the discrepancy of $I^{(1)}$ and $I^{(2)}$ values is getting smaller with increasing N. When N is larger than 800, $I^{(1)}$ and $I^{(2)}$ converge to the values around 4.6.

Anonymization and Effects on MI

In this section, we specifically focus on the effects of three common privacy approaches, and their effects on the information content in a dataset, as measured by the mutual information approach proposed.

Table 1 Mutual information (MI) matrices of the five experimental datasets. $I^{(1)}$ and $I^{(2)}$ represent the MI calculated by the first and second *Kraskov* estimators

	$\begin{bmatrix} I^{(1)}(\boldsymbol{gps}, \boldsymbol{gps}) & I^{(1)}(\boldsymbol{gps}, \boldsymbol{t}) \\ I^{(1)}(\boldsymbol{t}, \boldsymbol{gps}) & I^{(1)}(\boldsymbol{t}, \boldsymbol{t}) \end{bmatrix}$	$\begin{bmatrix} I^{(2)}(\boldsymbol{gps}, \boldsymbol{gps}) & I^{(2)}(\boldsymbol{gps}, \boldsymbol{t}) \\ I^{(2)}(\boldsymbol{t}, \boldsymbol{gps}) & I^{(2)}(\boldsymbol{t}, \boldsymbol{t}) \end{bmatrix}$
id_1	$\begin{bmatrix} 5.18 & 3.65 \\ 3.65 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 4.18 & 3.20 \\ 3.20 & 4.18 \end{bmatrix}$
id_2	$\begin{bmatrix} 5.18 & 3.69 \\ 3.69 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 4.18 & 2.65 \\ 2.65 & 4.18 \end{bmatrix}$
id_3	$\begin{bmatrix} 5.18 & 3.67 \\ 3.67 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 4.18 & 2.74 \\ 2.74 & 4.18 \end{bmatrix}$
id_4	$\begin{bmatrix} 5.18 & 3.69 \\ 3.69 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 4.18 & 2.60 \\ 2.60 & 4.18 \end{bmatrix}$
id_5	$\begin{bmatrix} 5.18 & 3.69 \\ 3.69 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 4.18 & 2.93 \\ 2.93 & 4.18 \end{bmatrix}$

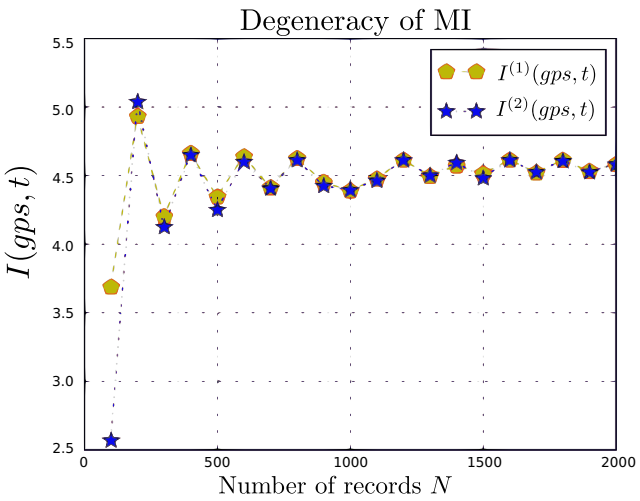


Fig. 6 Mutual information (MI) dependence on the number of samples in the dataset

Hashing the Identification Field Hashing is a very common technique for hiding data, with cryptographic hashes being the primary mechanism. One of the problems with hashing is that it is often mistaken as a mechanism for reducing information content over the dataset, whereas the actual function is to hide the contents of the data and change it to another form.

We have tried hashing the identification field in the datasets, and re-calculated the $I^{(1)}(\boldsymbol{gps}, \boldsymbol{t})$ and $I^{(2)}(\boldsymbol{gps}, \boldsymbol{t})$ values. The $I(\boldsymbol{gps}, \boldsymbol{t})$ values of the original data and the hashed data for each individual are compared in Fig. 7.

It is not surprising to see in Fig. 7 that there is not any change after hashing the identities. For $I^{(1)}$, the $I(\boldsymbol{gps}, \boldsymbol{t})$ values of the hashed ids are precisely identical to the values of the original ids, respectively for each individuals, staying in the range of 3.65–3.69, which agrees with Table 1 completely. For $I^{(2)}$, as we have discussed in Table 1 that $I^{(2)}(\boldsymbol{gps}, \boldsymbol{t})$ gives disparate values among individuals in

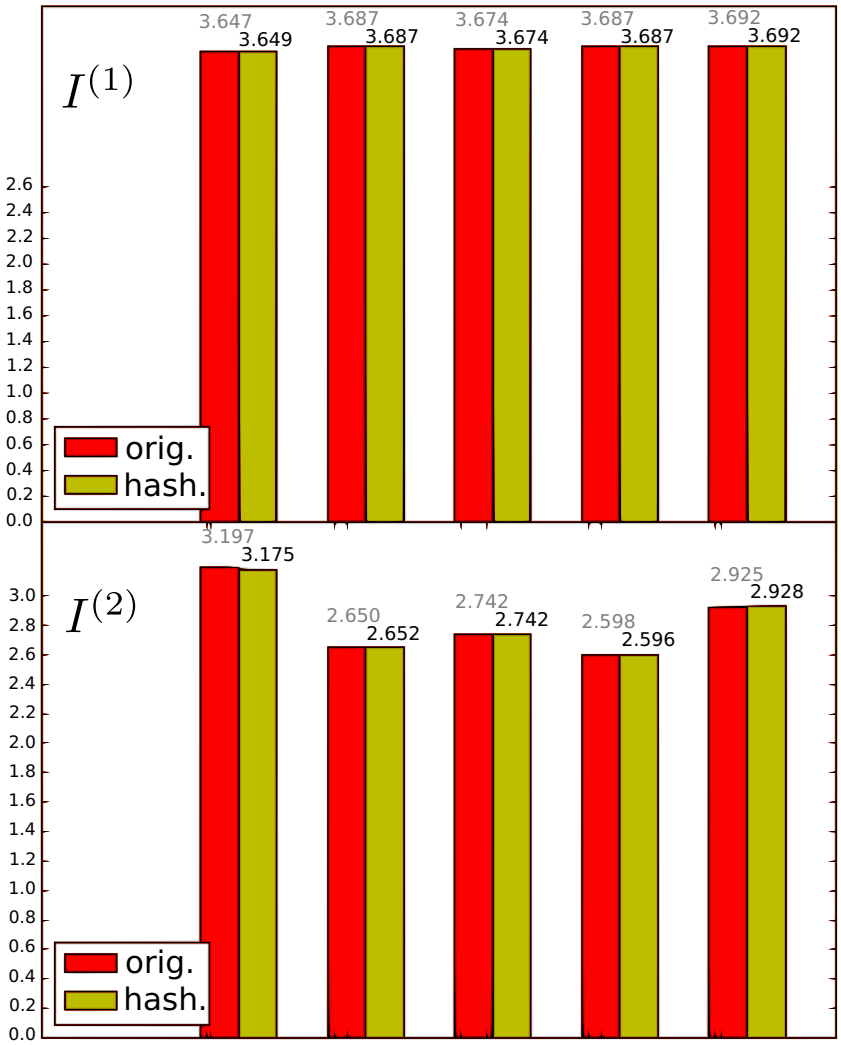


Fig. 7 Comparison of $I(\boldsymbol{gps}, \boldsymbol{t})$ values between the original data sets and the datasets with hashed identities; upper panel $I^{(1)}$, and lower panel $I^{(2)}$

the wider range of 2.60–3.20, due to the divergence of the trajectories.

This is obvious, as hashing only changes the unique individual identification into another unique identification form, nothing more. While this experiment may be totally trivial, it serves here to "prove" that hashing a user identifier does not prevent re-identification of said users: the rest of the information remains identical, and thus, the data is effectively the same. The internal coherence of the datasets is not really reduced. Thus, we confirm that hashing an identification field is a poor anonymization function, which does not reduce the information content of the datasets, as expected.

k-Anonymity Effects on the Trajectory Datasets

We have here used the generalization approach from κ-anonymity [24, 26, 42] to modify the dataset, and explore the influence of such changes on the mutual information values.

In the following Table 2, κ-anonymity applied to the GPS field means that we have in practice rounded the GPS coordinates (lat and lon) by 2 digits, compared to the original precision; when applied to the time field, we have also rounded the time to 10 minutes intervals (instead of second precision).

It should be noted that we only report the values for the first estimator, here. In practice, the changes in mutual information incurred by the chosen k-anonymity values on the GPS are relatively minimal, as can be seen in Table 2. It is interesting to note that the changes on the time cause much more distortion in the data (in terms of the mutual information), possibly because the granularity of the generalization is higher for the time, given the "rounding" chosen. The most interesting feature is that by altering both GPS and time at the same time, the mutual information is higher than when time alone is affected. We explain this by the fact that when these two fields are changed in the same fashion at the same time, the disturbance to the relationship between them is less than when only changing the time. This change to both fields "preserves" some of the relationship better, it seems.

ϵ-Differential Privacy

We have used the ϵ-differential privacy technique as the canonical example of an anonymization function to obfuscate the trajectory datasets by the Laplace mechanism [11, 13]. We define the privacy function to be a family set of $h = \{h^{(\boldsymbol{gps})}, h^{(\boldsymbol{t})}\}$, where $h^{(\boldsymbol{gps})}$, $h^{(\boldsymbol{t})}$ are the obfuscating functions to perturb the GPS field and time field, respectively.

Differential privacy was applied by adding controllable noise to the corresponding attribute in the dataset, which satisfies the Laplace distribution with mean 0 and standard deviation b

$$h^{(i)} = \texttt{diff}^{(i)}(\epsilon), \tag{39}$$

where ϵ is the differential privacy parameter. The standard deviation b of the Laplace noise can be then obtained by

$$b = \frac{\Delta f}{\epsilon}, \tag{40}$$

where Δf is the sensitivity of the attribute field.

We are then left with the question of selecting the structures within the dataset on which to apply differential privacy, to select suitable ϵ values, and to investigate the effects upon the dataset as a whole.

Table 2 Effects of k-anonymity on mutual information (MI) matrices for the five experimental datasets

$$\begin{bmatrix} I^{(1)}(\boldsymbol{gps},\boldsymbol{gps}) & I^{(1)}(\boldsymbol{gps},\boldsymbol{t}) \\ I^{(1)}(\boldsymbol{t},\boldsymbol{gps}) & I^{(1)}(\boldsymbol{t},\boldsymbol{t}) \end{bmatrix}$$

k-Anon	None (original)	GPS only	Time only	GPS and time
id1	$\begin{bmatrix} 5.18 & 3.65 \\ 3.65 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.70 \\ 3.70 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 2.25 \\ 2.25 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.35 \\ 3.35 & 5.18 \end{bmatrix}$
id2	$\begin{bmatrix} 5.18 & 3.69 \\ 3.69 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.73 \\ 3.73 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 2.25 \\ 2.25 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.58 \\ 3.58 & 5.18 \end{bmatrix}$
id3	$\begin{bmatrix} 5.18 & 3.67 \\ 3.67 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.79 \\ 3.79 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 2.30 \\ 2.30 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.58 \\ 3.58 & 5.18 \end{bmatrix}$
id4	$\begin{bmatrix} 5.18 & 3.69 \\ 3.69 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.42 \\ 3.42 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 2.23 \\ 2.23 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.07 \\ 3.07 & 5.18 \end{bmatrix}$
id5	$\begin{bmatrix} 5.18 & 3.69 \\ 3.69 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.64 \\ 3.64 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 2.24 \\ 2.24 & 5.18 \end{bmatrix}$	$\begin{bmatrix} 5.18 & 3.48 \\ 3.48 & 5.18 \end{bmatrix}$

In the following discussion, we used three family sets of privacy functions, which are:

$$\begin{aligned} h_1 &= \{\texttt{diff}^{(gps)}(\epsilon), \quad \emptyset^{(t)} \quad\}; \\ h_2 &= \{\emptyset^{(gps)}, \quad\quad \texttt{diff}^{(t)}(\epsilon)\}; \\ h_3 &= \{\texttt{diff}^{(gps)}(\epsilon), \quad \texttt{diff}^{(t)}(\epsilon)\}, \end{aligned} \tag{41}$$

where $\emptyset^{(i)}$ stands for taking no anonymization action to the attribute i. For example, h_1 means adding Laplace noise only to the GPS attribute, while the timestamps stay the same; h_2 means adding Laplace noise only to the timestamps attribute; h_3 means adding Laplace noises to both GPS and timestamp attributes.

Figure 8 shows the obtained pairwise MI values of $\boldsymbol{I}(h^{(gps)}, h^{(t)})$, where the privacy function sets are applied to the GPS field and time fields with various privacy parameters ϵ from 0 to 20. We can see that $\boldsymbol{I}(h^{(gps)}, h^{(t)})$

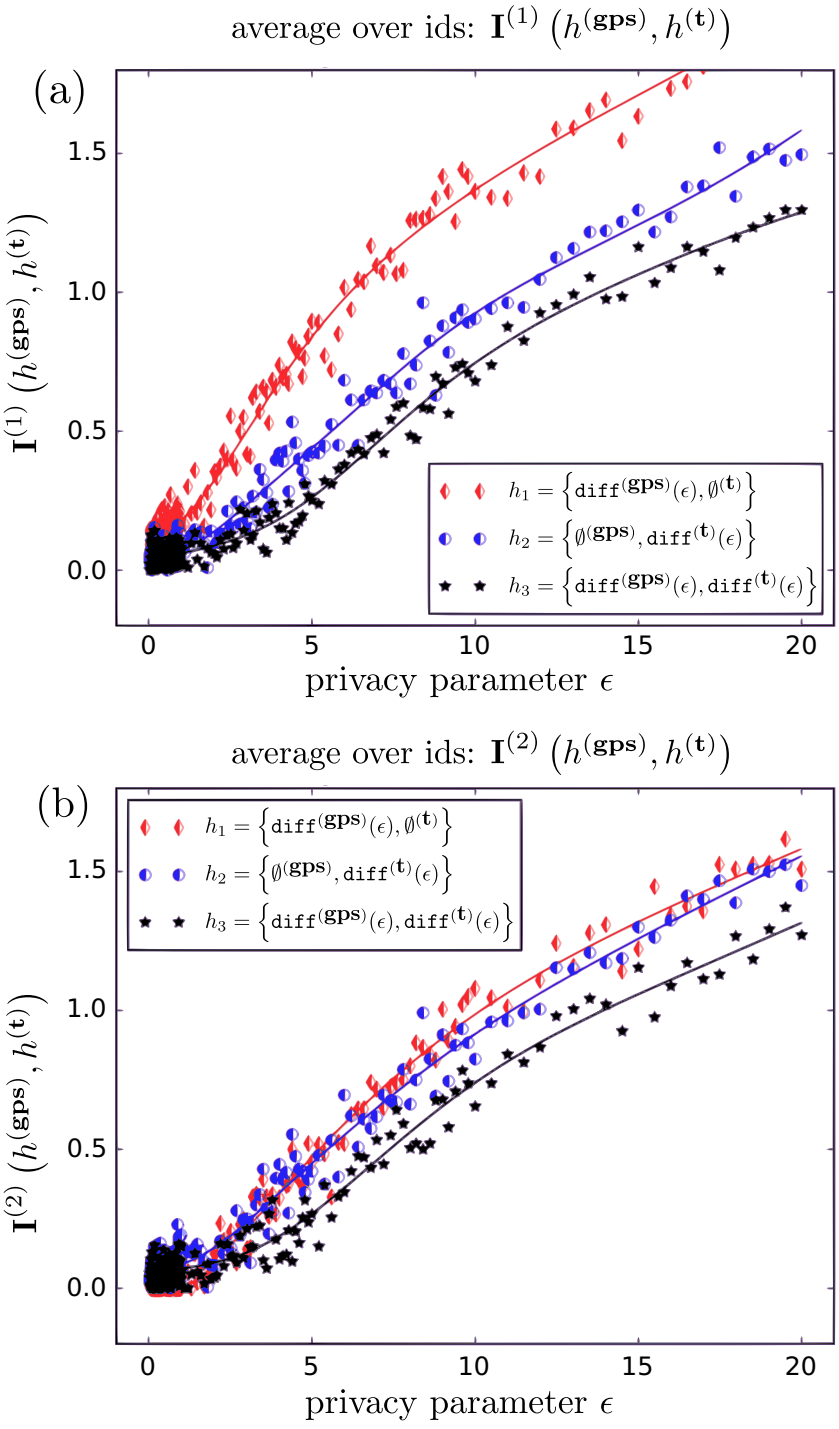


Fig. 8 Mutual information (MI) values of $\boldsymbol{I}(h^{(gps)}, h^{(t)})$, with the GPS field and time field obfuscated by differential privacy technique with the privacy functions of $h^{(gps)}$ and $h^{(t)}$, respectively. The MI values are obtained by taking the average of $\boldsymbol{I}(h^{(gps)}, h^{(t)})$ values from the 5 trajectories. The discrete markers are the obtained averaged MI values, while the corresponding solid lines are the fitted functions with ELM. **a** and **b** denote the MI values calculated by the first and second Kraskov estimators, respectively

is monotonically decreasing when the privacy parameter ϵ decreases (meaning we "increase" privacy).

When ϵ turns to close enough, but not equal, to 0, the MI values collapse at 0, where the fluctuations are the statistic errors caused by the small number of samples N in the datasets. It can be well explained by the fact that with smaller values of ϵ, the amplitudes of the Laplace noise (calculated by Eq. 40) become larger, which distort the metric space or topology of the original datasets more extensively to higher levels with increasing privacy. In other words, we can say that a small ϵ in differential privacy technique creates greatly anonymized datasets, and effectively alters the metric space with large distortion, while the information contents extracted from the anonymized datasets is reduced as a trade-off of increasing privacy. The linkability between the attributes is then weakened to prevent re-identification of the individuals. Hence the pairwise MI values are decreased.

The efficiency of altering the MI values by the privacy functions h_1, h_2, and h_3 is compared in Fig. 8. Both estimators indicate that when applying the differential privacy technique on GPS field (h_1) and time field (h_2) separately at the same privacy parameters ϵ, the time field is more sensitive to reduce the MI values, compared to the GPS field. Moreover, differential privacy applied on both GPS and time (h_3) fields at the same time is the most efficient among the anonymization functions h_1, h_2, and h_3.

As we have discussed before, small MI values stand for high distortions of the data, at possibly the cost of unusable data, while large MI values imply small alteration of the dataset topology, with a high re-identifiability risk. Therefore, we want to find an acceptable range of MI values, where the dataset is sufficiently anonymized to make sure there is a low risk of re-identification, while the amount of information in the distorted data is still sufficiently usable for future data analysis. Our goal is to control and quantify the distortion, by restricting the privacy parameters in the anonymization functions, or by conveying restrictions over the obfuscation functions in a controllable manner.

Derivatives of MI over Privacy Parameter ϵ

To measure the efficiency of the privacy parameter ϵ on mutual information, in the standard ϵ-differential privacy technique, we have calculated the derivatives of the MI values (from Fig. 8) over ϵ. The derivatives are shown in Fig. 9. Both estimators show that the derivatives for h_1 and h_2 reach the maximum when $\epsilon \sim 3.5$, implying the mutual information decreases the most rapidly with ϵ around this value. Meanwhile, the maximum of the derivatives for h_3 locate at $\epsilon \sim 7.5$, relatively higher than those for h_1 and h_2. The derivatives of MI over ϵ provide the information of the MI decreasing rates, which benefits finding efficiently the optimized parameters of the anonymization functions to fulfill the privacy restrictions.

It is worth noting that the behaviors of the MI and their derivatives (w.r.t. ϵ) in Figs. 8 and 9 are consistent with each other, i.e., they would lead to similar (if not identical) choices for the ϵ parameter, despite the differences in convergence speed noted before, as well as in the range of MI values for each. This is reassuring.

$\{\epsilon, \delta\}$-Differential Privacy

The ϵ-differential privacy mechanism discussed in the "ϵ-Differential Privacy" section is the absolute case, that when an individual is added or removed in the dataset, a given differential privacy algorithm will behave the same on the query outputs from the dataset [11, 13]. This gives a strong guarantee that presence or absence of an individual will not affect the final output of the algorithm.

In contrast with the standard ϵ-differential privacy, $\{\epsilon, \delta\}$-differential privacy allows the process to fail with $\delta > 0$ probability. In another word, for a database $\mathcal{D}'$, some outputs of $\mathcal{D}'$ have a small chance δ of happening, while for its adjacent database $\mathcal{D}$, they will absolutely not happen. So, with bad luck, an attacker can know that the individual in question is in the set, with certainty. Thus, each user in the dataset has a δ probability at risk in this case. $\{\epsilon, \delta\}$-differential privacy is a more general notion of privacy. ϵ-differential privacy can also be referred as $\{\epsilon, 0\}$-differential privacy with $\delta = 0$.

In this section, we have applied $\{\epsilon, \delta\}$-differential privacy on the GPS field and time field, simultaneously, for the experimental trajectory of id_4. The dependence of $\boldsymbol{I}^{(1)}(\boldsymbol{gps}, \boldsymbol{t})$ values on the differential privacy parameters of ϵ and δ is illustrated in Fig. 10a. We can see that regardless of the roughness of the $\boldsymbol{I}^{(1)}$ manifold due to the statistical errors of the estimators and the fluctuations of the noise generator, $\boldsymbol{I}^{(1)}$ is monotonically decreasing when ϵ and δ decrease (meaning an increase in privacy).

As we have explained before, a small ϵ presents large noise amplitudes, which imply sufficient anonymization and great privacy. Meanwhile, δ represents the probability of individuals at risk, and probability of the process to fail. A small δ implies a small possibility of the individuals failing to be anonymized by the Laplace mechanism. Thus, small parameter values for ϵ and δ signify "abundant" anonymization with big distortion of the metric space, and thus reduces the linkability between GPS field and time field. Thus, $\boldsymbol{I}^{(1)}$ behaves as a monotonically decreasing function when ϵ and δ decrease, as would be expected.

We can see in Fig. 10a that, when δ is fixed at 100% (which means that there is 100% probability that differential privacy fails to apply to the individuals), all the individuals in the dataset are at risk with fully intact queried information

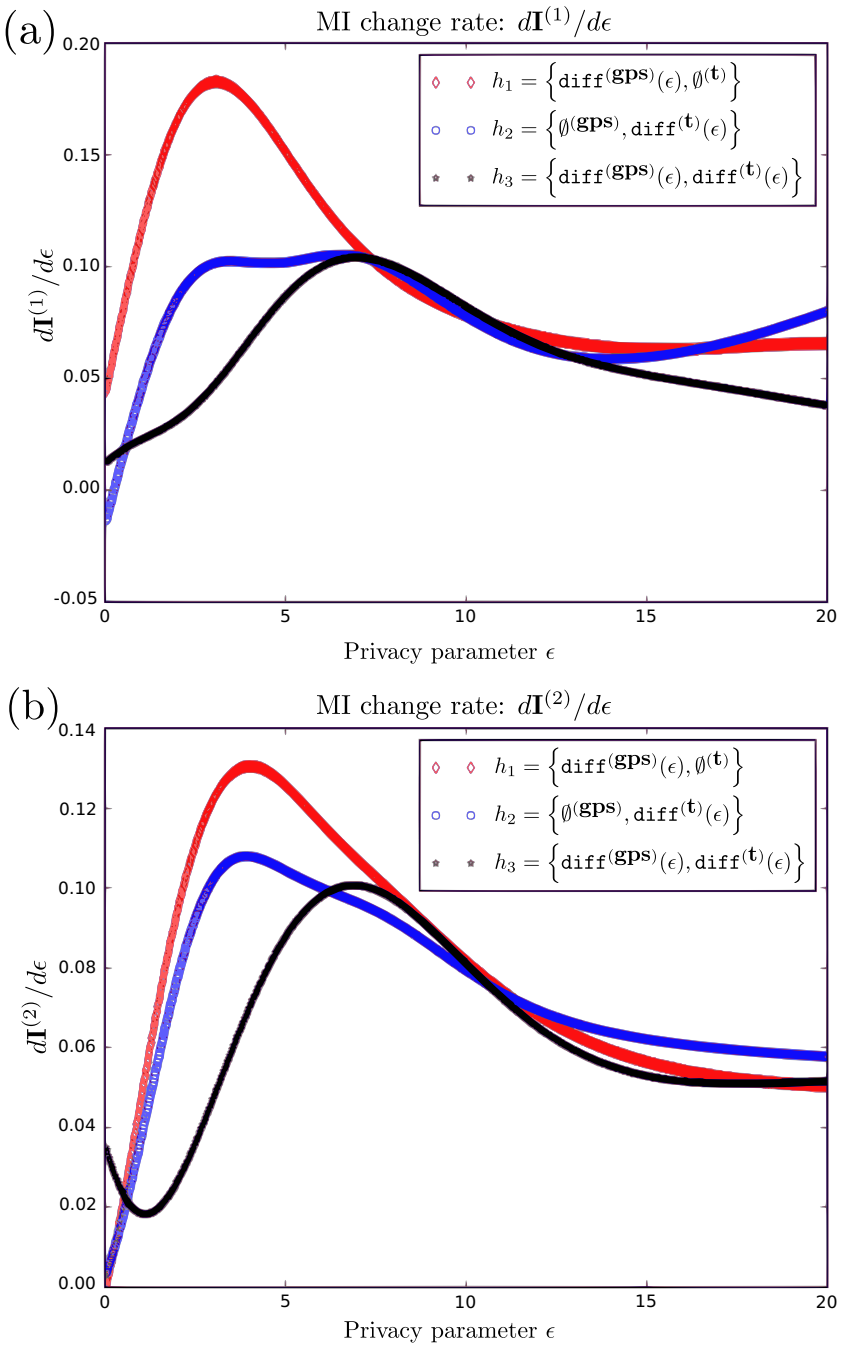


Fig. 9 Derivatives of $I^{(1)}$ and $I^{(2)}$ (from Fig. 8) over privacy parameter ϵ

available. In this case, the anonymization function does not affect the information at all, leaving the $\boldsymbol{I}^{(1)}$ value constant at 3.69 (agrees with Table 1), regardless of the changes of ϵ. Meanwhile, when δ is fixed at 0%, $\boldsymbol{I}^{(1)}$ decreases gradually from ∼1.3 to 0 with ϵ varying from 20 to 0. The case now becomes the standard ϵ-differential privacy mechanism with absolutely no risk of failure, precisely the same as the h_3 line (the black star and line) in Fig. 8a.

If we consider the parameter ϵ as a fixed constant, the varying of δ from 100 to 0% indicates the process of differential privacy going through complete failure to absolute success, with the probabilities of non-anonymized individuals from 100 to 0%. When comparing the tendencies of $\boldsymbol{I}^{(1)}$ on δ, at $\epsilon = 20$ and at $\epsilon \approx 0$, we can see that $\boldsymbol{I}^{(1)}$ is more sensitive at the large scale of δ in the case of $\epsilon \approx 0$, where the parameter δ plays an essential role on effectively diminishing $\boldsymbol{I}^{(1)}$ to 0. When ϵ is fixed at 20 (with small Laplace noise magnitudes and low privacy), $\boldsymbol{I}^{(1)}$ reveals a gradual decrease from 3.69 to 1.4 with δ varying from 100 to 0%. Meanwhile, when ϵ is fixed around 0

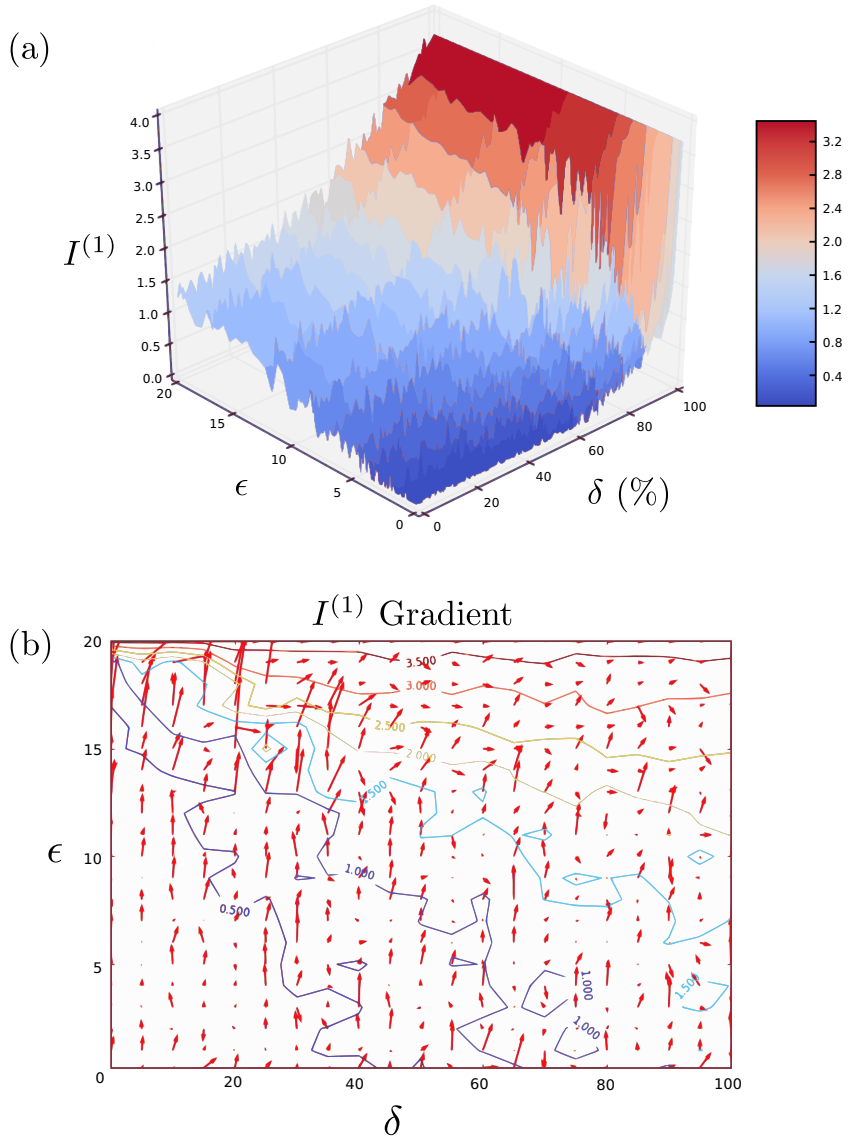


Fig. 10 **a** MI values of $\boldsymbol{I}^{(1)}\left(\texttt{diff}^{(gps)}(\epsilon,\delta),\texttt{diff}^{(t)}(\epsilon,\delta)\right)$, with the GPS field and time field simultaneously obfuscated by the $\{\epsilon,\delta\}$-differential privacy. Due to the heavy computations, $\boldsymbol{I}^{(1)}$ values are calculated only for one experimental trajectory in Fig. 5. **b** The gradient of $\boldsymbol{I}^{(1)}$, $\nabla\boldsymbol{I}^{(1)}(\epsilon,\delta)$, is depicted as a projected vector field, and its contour lines

(with considerable noise magnitudes and immense privacy), $\boldsymbol{I}^{(1)}$ shows a dramatic drop from 3.69 to ~ 0 with δ varying from 100 to 60, while $\boldsymbol{I}^{(1)}$ stays constant around 0 with δ further decreasing from 60 to 0. This implies that when the magnitudes of Laplace noises are massively large, even the anonymization applying on a small proportional amount of individuals (or a large possibility of differential privacy failing on individuals) can induce a large decrease of $\boldsymbol{I}^{(1)}$. Such a result has implications here that much of the data is actually correct despite noise being added.

Figure 10b depicts the gradient of $\boldsymbol{I}^{(1)}$, denoted as $\nabla\boldsymbol{I}^{(1)}(\epsilon,\delta)$, as well as the contour lines of $\boldsymbol{I}^{(1)}$. The $\nabla\boldsymbol{I}^{(1)}(\epsilon,\delta)$ is represented as a projected vector field by red arrows, which point in the direction of the greatest rate of $\boldsymbol{I}^{(1)}$ increase, and the arrow magnitudes are the slopes of the $\boldsymbol{I}^{(1)}$ graph in that direction. We can see that at smaller range of δ, nearly all the $\nabla\boldsymbol{I}^{(1)}$ field arrows point vertically up, with increasing magnitudes alongside the rising ϵ. This implies that when the proportion of non-anonymized individuals is small, ϵ plays a more important role on affecting $\boldsymbol{I}^{(1)}$, than ϵ. And with smaller magnitudes of Laplace noise, the performance of changing $\boldsymbol{I}^{(1)}$ caused by ϵ is more effective. On the other hand, when δ becomes larger, with larger probabilities of individuals at risk, the direction of the greatest rate of increase of $\boldsymbol{I}^{(1)}$ is not that uniform or straightforward: at higher ϵ, it goes practically to the δ rising orientation, while at lower ϵ, it goes roughly to the ϵ rising orientation.

The contour lines with the corresponding $\boldsymbol{I}^{(1)}$ values in the gradient vector field make it very clear to quantitatively observe the changes in mutual information, by adjusting the privacy parameters ϵ and δ. Assume that we have predefined the characteristic function $\chi_C = \chi_L \circ \chi_U$ (discussed in the Desirable Properties section), which restricts the mutual

information content of the dataset to the legal and useful domains. The parameters in the anonymization functions α (in this case, ϵ and δ in differential privacy) can then be narrowed down from the contour curves to the suitable bounds. Hence, we can achieve our goal that we can use mutual information to measure the distortion caused by the anonymization, and that we can quantitatively control the distortion by selectively choosing the parameters in the anonymization functions.

Discussion and Conclusion

In this paper, we have focused on establishing a practical and usable (although early and incomplete) framework, with the associated techniques, to estimate the effects of privacy techniques on a dataset, in terms of its usability (information content based).

We have developed and implemented a distance-mapping algorithm, which maps probabilistically the pairwise distances in a non-Euclidean space to a canonical Euclidean space, in a way that the pairwise distance distributions are approximately preserved (in terms of their probability mass). This approach allows for the computation of Mutual Information (the proposed information content metric, in this work) over non-Euclidean data.

We have demonstrated a mechanism for measuring the mutual information content of a given dataset and a structure for deciding whether that dataset is legally compliant and/or useful. In terms of privacy requirements, the possible future impact of automation and of the mathematical formalization of a legal test, such as the GDPR [6], is profound.

We have applied differential privacy to anonymize the dataset such that the personal data in the information considered, is reduced to prevent re-identification of individuals. The distortion caused by the anonymization functions is then quantified by means of mutual information. The linkability between attributes in the data is measured by the pairwise mutual information values. We found that having a predefined classification function $\chi_C = \chi_L \circ \chi_U$ (discussed in the Desirable Properties section) which restricts the linkability of the information after applying differential privacy functions, helps the anonymization process in such a way that the parameters in the anonymization functions can be narrowed down to practical ranges.

The examples and structures shown in this paper demonstrate that a metric for privacy, a structure for anonymization, and classification functions generated for particular privacy aspects can be constructed. However, deciding whether a dataset is useful (i.e., usable for a specific data-based process) is relatively "trivial" in comparison to deciding whether it is legal. This remains exceptionally difficult, in that we still do not have a good notion of this context in general for a dataset. For example, the situations in which the data is to be used; how re-identification/deanonymization might take place; and how one might even construct a classifying function such as χ_L from a text such as the GDPR.

The proposed methodology is, as far as the authors are aware, unique in that it allows for the computation of mutual information between two random variables that can have values in any (non-Euclidean) metric space. A direct advantage of this approach is to be able to quantify objectively the usability of the data by measuring how much of the initial mutual information (before the anonymization process takes place) is lost by the anonymization process. If a measure of "legality" of the dataset, and/or "privacy" of the data set, is available, the problem of how much to anonymize a dataset (and the determination of the optimal parameters to do so) becomes a multi-criteria optimization problem, and is quantifiable. This methodology has very clear limitations though, due to both the assumed underlying structure we take for granted in this work (a continuous metric space), and the use of mutual information as an information content metric:

- Many datasets have non-continuous data, and the jump from non-continuous to continuous for the proposed method is simply not possible as is. This limits severely the use of this method in numerous cases;
- The assumption that there exists a distance function in the space of the data is also a very limiting factor, as such a distance would not make sense for many data formats, or at least be far from obvious to devise;
- Mutual information is a very computationally expensive metric to calculate and is also relatively unstable, in some cases [22, 36]. In this sense, it is not ideal, but we are not aware (yet) of any better means of quantifying properly the information content shared by two random variables.

Future work extending this research will focus specifically on improving the computational time required, as well as accommodating the non-continuous data case. As such, the current implementation of this methodology requires the training of a machine learning model on large amounts of data, as the interpolation capabilities of the model currently do not work well with small amounts of data during the training. In addition, there is a current need, in this methodology, to be able to draw the data for the learning, from the metric space that supports the random variable, with the appropriate distribution. This assumes that one has access to such knowledge, which is not always the case, in practice. In our experiments, we know the details of how GPS coordinates can be drawn from the space of all GPS coordinates, and the structure of that space (boundaries, continuity,...). Ideally, one would like to get away from

needing this information, and be able to use solely the data that is provided, without requiring any assumption about the underlying structure nor distribution of the data. Extended experimentation on varying data types will provide further verification of the applicability (and limitations) of the methodology. We have so far validated our methodology using company internal data, as well as the timestamped GPS data presented in this paper, but need to have experiments that reach across varying data types and structures to illustrate how the approach works in very different cases.

Funding Information This work was supported by the research from SCOTT project. SCOTT (www.scott-project.eu) has received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement no. 737422. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Spain, Finland, Ireland, Sweden, Germany, Poland, Portugal, Netherlands, Belgium, and Norway.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflicts of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Abramowitz M. Handbook of mathematical functions, with formulas, graphs, and mathematical tables. New York: Dover Publications; 1974.
2. Asgarian E, Kahani M, Sharifi S. The impact of sentiment features on the sentiment polarity classification in Persian reviews. Cogn Comput. 2018;10(1):117–35. 00001.
3. Auer P, Burgsteiner H, Maass W. A learning rule for very simple universal approximators consisting of a single layer of perceptrons. Neural Netw. 2008;21(5):786–95.
4. Belghazi MI, Baratin A, Rajeswar S, Ozair S, Bengio Y, Courville A, Hjelm RD. MINE: mutual information neural estimation. arXiv:1801.04062 [cs, stat]. 2018. 00003.
5. Cambria E, Huang G-B, Kasun LLC, Zhou H, Vong CM, Lin J, Yin J, Cai Z, Liu Q, Li K, et al. Extreme learning machines [trends & controversies]. IEEE Intell Syst. 2013;28(6):30–59.
6. European Commission. European Commission's press release announcing the proposed comprehensive reform of data protection rules, 25 January. 2012.
7. Cover TM, Thomas JA. Elements of information theory (Wiley series in telecommunications and signal processing). New York: Wiley-Interscience; 2006.
8. Cybenko G. Approximation by superpositions of a sigmoidal function. Math Control Signals Syst (MCSS). 1989;2(4):303–14.
9. Dashtipour K, Poria S, Hussain A, Cambria E, Hawalah AYA, Gelbukh A, Zhou Q. Multilingual sentiment analysis: state of the art and independent comparison of techniques. Cogn Comput. 2016;8(4):757–71. 00025.
10. Ding S, Zhao H, Zhang Y, Xu X, Nie R. Extreme learning machine: algorithm, theory and applications. Artif Intell Rev. 2015;44(1):103–15.
11. Dwork C. Differential privacy. Berlin: Springer; 2006, pp. 1–12.
12. Dwork C. Differential privacy: a survey of results. In: Theory and applications of models of computation, volume 4978 of Lecture Notes in Computer Science. Berlin: Springer; 2008. p. 1–19.
13. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. Berlin: Springer; 2006, pp. 265–84.
14. EU. 2000/520/EC: Commission Decision of 26 July 2000 pursuant to Directive 95/46/EC of the European Parliament and of the Council on the adequacy of the protection provided by the safe harbour privacy principles and related frequently asked questions issued by the US Department of Commerce (notified under document number C(2000) 2441) (Text with EEA relevance.) 2000.
15. François D. High-dimensional data analysis: optimal metrics and feature selection. VDM Verlag, 01. 2008.
16. Goss RN. Information theory with applications (silviu guiaşu). SIAM Rev. 1979;21(4):579–80.
17. Hafiz M. A collection of privacy design patterns. In: Proceedings of the 2006 conference on pattern languages of programs, PLoP '06. New York: ACM; 2006. p. 7:1–13.
18. Holmes C, Nemenman I. Progress in estimation of mutual information for real-valued data. Bulletin of the American Physical Society; 2018.
19. Huang G-B, Chen L, Siew CK, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw. 2006;17(4):879–92.
20. Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: theory and applications. Neurocomputing. 2006;70(1):489–501.
21. The Information Commissioner's Office (UK). Direct marketing: data protection act privacy and electronic communications regulations, 24 November 2013. Version 1.1.
22. Kraskov A, Stögbauer H, Grassberger P. Estimating mutual information. Phys Rev E. 2004;69(6):066138.
23. Lauren P, Qu G, Yang J, Watta P, Huang G-B, Lendasse A. Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks. Cognit Comput. 1–14. 2018. 00000.
24. Li N, Li T. t-closeness: privacy beyond κ-anonymity and ℓ-diversity. In: Proceedings of IEEE 23rd international conference on data engineering (ICDE'07); 2007.
25. Ma Y, Peng H, Khan T, Cambria E, Hussain A. Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis. Cognit Comput. 2018;10(4):639–650. https://doi.org/10.1007/s12559-018-9549-x.
26. Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M. ℓ-diversity: privacy beyond κ-anonymity. In: 2013 IEEE 29th international conference on data engineering (ICDE); 2006. p. 24.
27. Mahmud M, Kaiser MS, Hussain A, Vassanelli S. Applications of deep learning and reinforcement learning to biological data. IEEE Trans Neural Netw Learn Syst. PP. 2017. 00004.
28. Miche Y, Oliver I, Holtmanns S, Akusok A, Lendasse A, Björk K-M. On mutual information over non-Euclidean Spaces, data mining and data privacy levels. Cham: Springer International Publishing; 2016, pp. 371–83.
29. Miche Y, Oliver I, Holtmanns S, Kalliola A, Akusok A, Lendasse A, Björk K-M. Data anonymization as a vector quantization problem: control over privacy for health data. In: Availability, reliability, and security in information systems, Lecture Notes in Computer Science. Cham: Springer; 2016. p. 193–203.
30. Miche Y, Oliver I, Ren W, Holtmanns S, Akusok A, Lendasse A. Practical estimation of mutual information on non-Euclidean spaces. In: Machine learning and knowledge extraction. Cham: Springer; 2017. p. 123–36.

31. Miche Y, Sorjamaa A, Bas P, Simula O, Jutten C, Lendasse A. Op-elm: optimally pruned extreme learning machine. IEEE Trans Neural Netw. 2010;21(1):158–62.
32. Miche Y, Van Heeswijk M, Bas P, Simula O, Lendasse A. Trop-elm: a double-regularized elm using lars and tikhonov regularization. Neurocomputing. 2011;74(16):2413–21.
33. Molina D, LaTorre A, Herrera F. An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions. Cognit Comput. 2018;10(4):517–544. https://doi.org/10.1007/s12559-018-9554-0.
34. Nissenbaum H. A contextual approach to privacy online. Daedalus. 2011;140(4):32–48.
35. Oliver I. Privacy engineering: a data flow and ontological approach. CreateSpace Independent Publishing, July 2014. 978-1497569713.
36. Pál D, Póczos B, Szepesvári C. Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graph. In: Advances in neural information processing systems; 2010. p. 1849–57.
37. Rao CR, Mitra SK. Generalized inverse of matrices and its applications. 1971.
38. Reed J, Pierce BC. Distance makes the types grow stronger: a calculus for differential privacy. In: ACM SIGPLAN international conference on functional programming (ICFP), Baltimore; 2010.
39. Savitha R, Suresh S, Kim HJ. A meta-cognitive learning algorithm for an extreme learning machine classifier. Cogn Comput. 2014;6(2):253–63. 00048.
40. Schneier B. Architecture of privacy. IEEE Secur Priv. 2009; 7(1):88.
41. Singh PK. Similar vague concepts selection using their euclidean distance at different granulation. Cogn Comput. 2018;10(2):228–41. 00001.
42. Sweeney L. κ-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl-Based Syst. 2002;10(5):557–70.
43. Ustaran E, editor. European Privacy: Law and Practice for Data Protection Professionals. An IAPP Publication, 2012. 978-0-9795901-5-3.
44. Van Heeswijk M, Miche Y, Oja E, Lendasse A. Gpu-accelerated and parallelized elm ensembles for large-scale regression. Neurocomputing. 2011;74(16):2430–7.
45. Wang H, Zhang Y, Waytowich NR, Krusienski DJ, Zhou G, Jin J, Wang X, Cichocki A. Discriminative feature extraction via multivariate linear regression for SSVEP-based BCI. IEEE Trans Neural Syst Rehabil Eng. 2016;24(5):532–41.
46. Wang R, Zhang Y, Zhang L. An adaptive neural network approach for operator functional state prediction using psychophysiological data. Integrated Computer Aided Eng. 2015;23:81–97. 00006.
47. Zeng D, Zhao F, Shen W, Ge S. Compressing and accelerating neural network for facial point localization. Cognit Comput. 2018;10(2):359–67. 00001.
48. Zhang Y, Wang Y, Jin J, Wang X. Sparse Bayesian learning for obtaining sparsity of EEG frequency bands based feature vectors in motor imagery classification. Int J Neural Syst. 2017;27(02):1650032. https://doi.org/10.1142/S0129065716500325.
49. Zhang Y, Zhou G, Jin J, Zhao Q, Wang X, Cichocki A. Sparse Bayesian classification of EEG for brain-computer interface. IEEE Trans Neural Netw Learn Syst. 2015;27:1–1. 00058.