HAL open science

Data Anonymization as a Vector Quantization Problem: Control Over Privacy for Health Data

Yoan Miche, Ian Oliver, Silke Holtmanns, Aapo Kalliola, Anton Akusok, Amaury Lendasse, Kaj-Mikael Björk

▶ **To cite this version:**

Yoan Miche, Ian Oliver, Silke Holtmanns, Aapo Kalliola, Anton Akusok, et al.. Data Anonymization as a Vector Quantization Problem: Control Over Privacy for Health Data. International Conference on Availability, Reliability, and Security (CD-ARES), Aug 2016, Salzburg, Austria. pp.193-203, 10.1007/978-3-319-45507-5_13 . hal-01635008

HAL Id: hal-01635008

https://inria.hal.science/hal-01635008v1

Submitted on 14 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Data Anonymization as a Vector Quantization Problem: Control over Privacy for Health Data

Yoan Miche[1], Ian Oliver[1], Silke Holtmanns[1], Aapo Kalliola[1,4], Anton Akusok[3], Amaury Lendasse[2], and Kaj-Mikael Björk[3]

[1] Bell Labs, Nokia, Finland
[2] Department of Mechanical and Industrial Engineering
and the Iowa Informatics Initiative, The University of Iowa, Iowa City, USA
[3] Arcada University of Applied Sciences, Helsinki, Finland
[4] Aalto University, Finland

Abstract. This paper tackles the topic of data anonymization from a vector quantization point of view. The admitted goal in this work is to provide means of performing data anonymization to avoid single individual or group re-identification from a data set, while maintaining as much as possible (and in a very specific sense) data integrity and structure. The structure of the data is first captured by clustering (with a vector quantization approach), and we propose to use the properties of this vector quantization to anonymize the data. Under some assumptions over possible computations to be performed on the data, we give a framework for identifying and "pushing back outliers in the crowd", in this clustering sense, as well as anonymizing cluster members while preserving cluster-level statistics and structure as defined by the assumptions (density, pairwise distances, cluster shape and members...).

1 Introduction

In this paper, we limit ourselves to the problem of user re-identification from a dataset. We decide to focus on two very specific questions: given a set of records with no obvious information that would allow for easily identifying a single person from the dataset, (i) can we make sure that no one is easily identifiable from the data (and identify it), and (ii) if some individuals are easy to identify, can we modify the data so as to "blend them in" while retaining the key characteristics of the data statistics? The approach we take is to consider the data fields (over a set of records) as separate entities and try to build clusters of records based on metric proximity: if the records have similar values accross several vector elements, they are likely to be grouped together. We assume then that if such a group is large enough and that the records inside that group have been "stirred" enough, identification of a single individual becomes impossible. One of the main assumption in this paper is that whatever further processing is to be performed on the anonymized data, is relying on these "group statistics/properties" to be as intact as possible. The proposed "stirring" of the data implies that global data statistics and structures will be preserved, but local ones are disturbed.

We basically want to preserve the underlying manifold structure (in terms of the cluster of data that it is composed of) as much as possible, while locally shuffling the data around. In order to clarify some of the notions presented, we introduce here an example data set of health-related information in Table 1. We consider for the purpose of this example that this represents the full health records from a certain medical institution.

	Non-Sensitive			Sensitive
ID	Zip Code	Age	Nationality	Condition
01	13053	28	Russian	Heart Disease
02	13068	29	American	Heart Disease
03	13068	21	Japanese	Viral Infection
04	13053	23	American	Viral Infection
05	14853	50	Indian	Cancer

Table 1. Example of Health Data records from a medical institution.

The records from Table 1 show no obvious easily identifiable information when considering single fields. Nevertheless, relationships between the non-sensitive fields in this data can probably make it relatively easy to identify some individuals: within a zip code, the nationality and the age allow someone to restrict the set of possible individuals dramatically. The last individual in the table is even more striking as her age, nationality and zip code surely make her stand out of the rest.In such a situation, the proposed approaches in this paper seek to "blend in" this last individual from Table 1 with the rest of the records, as well as making sure that all the records get "shuffled" (regarding each of the fields) so as to make the data anonymized. In effect, we want here to actively modify the data values, not by changing their nature (as would hashing the values do, e.g.) or by omitting them, but really by modifying the values to realistic ones (belonging to the same category/set) in a way that preserves some of the information.

2 High-Level Motivation for Data Privacy

In this section, we propose a high level description of the problem tackled in this paper. The next sections then describe the proposed means of doing so. In an ideal situation, data mining and classification or partition of data, in particular for health and medical data [6], can be made in an unambiguous manner; meaning that, for example, a classification of the data can be made and the number of border cases is minimal. Application of algorithms that increase the privacy (or the entropy) of a system distort this in some known manner, in terms of the direct effects on the data fields. For example κ-anonymity [2] and ℓ-diversity [8] reduce the distribution and amounts of unique values in the discrete valued cases; differential privacy [4] adds noise in the continuous valued

cases, for example, speeds, distances etc. A privacy function distorts a system such that classification and/or mining either cannot be made or becomes difficult to make in a reasonable manner [5].

In this work, we consider the case of such privacy functions that modify the data in such a way as to avoid changing the "format" of the data, and thus the underlying space in which the data lies. Indeed, another way to consider this is that the privacy functions usually alter the underlying space or topology of the space rather than moving the elements themselves. This altering of the topology in the best case involves continuous (in the sense of metric preserving or homotopy preserving) stretching and shrinking, but may also include non-continuous tearing and creasing of the space such that the resolution of the original metric function is no longer possible. The challenge here is then to avoid this deformation of the underlying space, by attempting to shuffle and move the data around in the best manner (regarding increasing the privacy and minimising the distortions on the data). The work in this paper is aimed at this problem: proposing several practical solutions to the identification problem from Table 1. We first define in the next section, some notations and assumptions on the structure of the data, and first look at the problem of moving a single sample (or a group of them) back into bigger clusters. We then tackle the problem of increasing the privacy for the samples within a cluster so as to make sure that re-identification, even within a cluster, is more difficult.

3 Methodology for Data Anonymization in a Data Clustering Context

3.1 Some Notational Details

Traditionally in the data privacy literature, one defines a table $\mathbf{T}$ of N records as $\mathbf{T} = \{\mathbf{t}_1, \ldots \mathbf{t}_N\}$, with attributes $\{A_1, \ldots, A_d\}$. We have that $\mathbf{T} \in \Omega$, the set of all possible records (samples), and $\mathcal{A} = \{A_1, \ldots, A_d\}$ is the set of all possible d attributes (in this case, all possible attributes are used in table $\mathbf{T}$). Typically, one denotes the value of a certain attribute A_j for sample $\mathbf{t}_i$ as $\mathbf{t}_i[A_j]$. In this paper, and for the developments below, we take the liberty to note $\mathbb{X}^{(j)}$ the set of all possible values for a certain attribute A_j. Referring to Table 1 for our example case, if A_j is the attribute for the Zip Code of the patients, this means that $\mathbb{X}^{(j)}$ represents the set of all possible Zip Codes (possibly limited to the existing ones that make sense within the context of this table, e.g. limited to a country).

We then assume that it is possible to define a distance function $d^{(j)} : \mathbb{X}^{(j)} \times \mathbb{X}^{(j)} \longrightarrow \mathbb{R}_+$ over this set $\mathbb{X}^{(j)}$. Note that the metric space $\mathcal{X}^{(j)} = (\mathbb{X}^{(j)}, d^{(j)})$, defined by these two entities need not be Euclidean. Some considerations on such distance functions over non-Euclidean spaces are detailed in the following section 3.2. Departing slightly from the data privacy notations and denoting by $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^T$, the matrix of N samples holding the health records. A record $\mathbf{t}_i$ is now defined as $\mathbf{t}_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,d}]$, $a_{i,j} \in \mathbb{X}^{(j)}$, with $\mathbb{X}^{(j)}$ the set considered as part of the metric space $\mathcal{X}^{(j)} = (\mathbb{X}^{(j)}, d^{(j)})$.

With these extended notations, we can see the column $[a_{1,j}, \ldots, a_{N,j}]^T \in \mathbb{X}^{(j)\,N\times 1}$ as a discrete random variable (or a set of realizations of the underlying random variable, more precisely) over $\mathcal{X}^{(j)}$. The following section discusses the previous assumption of being able to define a distance function over a potentially non-Euclidean space.

3.2 Distance Functions over non-Euclidean spaces

The argument for considering the use of distances over non-Euclidean spaces in this work, is that it is possible to tweak and modify such non-Euclidean distances so that their distribution and properties will be "close enough" to that of the original Euclidean distance. Most of the developments in this paper rely on having "meaningful and consistent" distance functions across all the dimensions, so that they can be at least compared, even if this means re-mapping the distribution of its values.

More formally, let us assume that we have two metric spaces $\mathcal{X}^{(i)} = (\mathbb{X}^{(i)}, d^{(i)})$ and $\mathcal{X}^{(j)} = (\mathbb{X}^{(j)}, d^{(j)})$, with $\mathcal{X}^{(i)}$ the canonical Euclidean space (i.e. $\mathbb{X}^{(i)} = \mathbb{R}^d$ and $d^{(i)}$ the Euclidean norm) and $\mathcal{X}^{(j)}$ a non-Euclidean metric space endowed with a non-Euclidean metric. Drawing uniformly samples from the set $\mathbb{X}^{(j)}$, we form $\mathbf{x}^{(j)} = \left[x_1^{(j)}, \ldots, x_n^{(j)}\right]$, a set of values (realizations of the underlying random variable), with $x_l^{(j)} \in \mathbb{X}^{(j)}$. Denoting then by $f_{d^{(j)}}$ the distribution of pairwise distances over all the samples in $\mathbf{x}^{(j)}$, we assume that it is possible to modify the distribution of the values of the non-Euclidean metric $d^{(j)}$ (into a distribution $f_{d^{(j)}}^{\text{map}}$) such that

$$\lim_{n\to\infty} f_{d^{(j)}}^{\text{map}} = f_{d^{(i)}}, \tag{1}$$

where $f_{d^{(i)}}$ is the distribution of the Euclidean distances $d^{(i)}$ over the Euclidean space $\mathcal{X}^{(i)}$ and $f_{d^{(j)}}^{\text{map}}$ is a non-linear transformation of the original distribution $f_{d^{(j)}}$ by a certain function.

The limit here is over n as the distribution $f_{d^{(j)}}$ is considered to be estimated using a limited number n of realizations of the random variables, and we are interested in the limit case where we can "afford" to draw as many samples as possible to be as close to the Euclidean metric as possible. That is, that we can make sure that the non-Euclidean metric behaves over its non-Euclidean space, as would a Euclidean metric over a Euclidean space. This assumption is "theoretically reasonable", as it comes down to being able to transform a distribution $f_{d^{(j)}}$ into another $f_{d^{(j)}}^{\text{map}}$, given both. And while this may not be simple nor possible using linear transformation tools, most Machine Learning techniques are able to fit a continuous input to another different continuous output (this is basically the well-known Universal Function Approximator property [3]). It can be noted that using such tools, the mapping will not be perfect (as we will work with discrete versions of the distributions) and will not result in the equality case from Eq. 1. Nevertheless, we assume in this paper that this is sufficient for our needs. With this assumption in mind, we come to the problem of addressing Differential Privacy approaches as a Vector Quantization matter.

4 Considering Differential Privacy as a Vector Quantization problem

Using the previous notations introduced, Differential Privacy aims at finding sets or clusters (groups) C_l of samples

$$C_l = \{\mathbf{t}_i, \mathbf{t}_j \in \mathbf{T} | \forall i, j \in [\![1, N]\!], i \neq j, \forall k \in [\![1, d]\!], \\ d^{(k)}(a_{i,k}, a_{j,k}) \leq \varepsilon_k \Big\}, \tag{2}$$

with ε_k the maximum radii of the cluster C_l (each dimension k can have a separate radius, thus). The total number of clusters C is here determined by the choices made for the maximum radii of them, i.e. the ε_k. Intuitively, these C_l are clusters of samples that are "not too distant from each other". If all the metric spaces (across all dimensions) were Euclidean, Eq. 2 would simply define the sets of samples that have pairwise Euclidean distance smaller than a certain epsilon. In this respect, we are considering similarity between groups of sample as a defined by cluster density across all dimensions. In our case, we generalize this definition by potentially having a different distance function for each dimension, thus bounded by different ε_k. Note that samples that are "alone" in their cluster basically represent outliers in terms of the data they hold (and thus, individuals): they might be very easy to identify/recognize out of the rest of the records, as they do not "fit with others". This observation can be generalized to clusters that have "few samples" in the ball they define. "Few" has to be defined, in this case. This is directly related to the κ in κ-anonymity.

Denoting by m, the previous "few", if $|C_{l1}| \leq m$, there are not enough records in the cluster: They might be easy to identify, or represent too obvious a group. The goal is then to modify as few dimensions as possible (so as to minimize distortion of the data) to bring these records in the nearest cluster C_{l2} which respects $|C_{l2}| > m$ or so that $|C_{l1}| + |C_{l2}| > m$. To be able to find which nearby cluster is the most fitting for such a lonesome sample, we decide to rely on centroids. We thus need to calculate a centroid (or representative) of the clusters such that $|C_l| > m$. Note that as the sets $\mathbb{X}^{(j)}$ across which the data is defined do not necessarily have any implied order, we have to use solely the distances between samples to calculate the most fitting centroid.

This comes to determining the centroid c_l of cluster C_l with only inter-records distances (pairwise distances for all samples within one cluster):

$$c_l^{(j)} = \arg \min_{a_{k,j} \in \mathbb{X}^{(j)}} \left[\sum_{a_{i,j} \in C_l} d^{(j)}(a_{i,j}, a_{k,j}) \right], \tag{3}$$

where $c_l^{(j)}$ denotes the j-th coordinate of the centroid c_l of cluster C_l, and Eq. 3 has an abuse of notations in the summation index to avoid too heavy notations: the summation is made over the j-th coordinate $a_{i,j}$ of all the samples $\mathbf{t}_i$ in cluster C_l. This is to avoid defining the set of samples in the cluster formally.

From Eq. 3, it can be seen that the centroid coordinates are picked from the sets $\mathbb{X}^{(j)}$, and not calculated as some mean value over the samples present in the cluster. This would not have any sense in the case of discrete $\mathbb{X}^{(j)}$, so this definition is more practical for the general purpose.

We do not discuss in this paper the algorithmic means of finding such centroids based on this definition from Eq. 3. With the centroids of each cluster estimated, we can then decide how to move samples that are lonesome and too easy to identify.

4.1 Moving samples to nearby clusters

The task of moving a sample (or a small enough group of them) into a near cluster first requires the determination of the most suitable cluster for each of these samples.

Identifying the most suitable cluster Intuitively, and in order to preserve data as much as possible, the most suitable cluster C_l for this application is such that the total distortion, approximated in this case by how much the outlier $\mathbf{t}_o$ is moved across all dimensions, is minimal. Thus, denoting by $\mathbf{t}_o = [a_{o,1}, \ldots, a_{o,d}]$ an outlier, $\mathcal{C} = \{C_k\}_{1 \leq k \leq C}$ the set of all the clusters (which have a sufficient amount of samples in them), and by $d^{(j)}_{\text{map}}$ the mapped version of the distance function $d^{(j)}$ (so that the distribution of its values matches that of an Euclidean metric, see section 3.2), we get

$$C_l = \arg\min_{C_k \in \mathcal{C}} \left[\sum_{j=1}^{d} d^{(j)}_{\text{map}} \left(a_{o,j}, c^{(j)}_k \right) \right]. \tag{4}$$

One argument for using the mapped distances $d^{(j)}_{\text{map}}$ in this determination of the suitable cluster, is that we have to make a decision over all the dimensions at once, regarding the distortion generated by moving the outlier into a cluster. Therefore, in order to quantify this distortion across all dimensions at once, it is important that the distances are all within similar ranges and following similar distributions (otherwise, some dimensions will be "favoured" by the sum, possibly unjustly). Actual weighting of the distances in order to artificially favour some dimensions is the subject of further work.

Moving the sample to the decided cluster Once the most suitable cluster C_l for outlier $\mathbf{t}_o$ has been determined (note that there might not be a unique solution to this cluster determination), the problem is to move the outlier within that cluster so as to modify the actual values of the outlier as little as possible. We identify three ways to do this in practice, out of which the first is probably the best in terms of low distortion, but also the most difficult — and thus probably not achievable in real cases. In all the following three cases, the following steps are applied:

$$\forall k \in [\![1, d]\!], \begin{cases} a_{o,k} = a_{o,k}^{\text{new}} \text{ if } d^{(k)}(a_{o,k}, c_l^{(k)}) > \\ \qquad \max\limits_{a_{i,k} \in C_l} \left[d^{(k)}(a_{i,k}, c_l^{(k)}) \right] \\ a_{o,k} \qquad \text{unchanged otherwise} \end{cases}, \tag{5}$$

where $a_{o,k}^{\text{new}}$ is the new value to be given to the k-th coordinate of the outlier $\mathbf{t}_o$, and $\max_{a_{i,k} \in C_l} \left[d^{(k)}(a_{i,k}, c_l^{(k)}) \right]$ is in fact the maximum intra-cluster distance between cluster elements and the centroid c_l of the cluster C_l. Thus, we ensure that the modification of this specific dimension does not modify too much the intra cluster distances. We then propose three approaches to determine the new $a_{o,k}^{\text{new}}$ value: (a) Setting it to the centroid value, and adding some noise; (b) Setting it to the centroid value only; (c) Setting it to an existing cluster element value.

(a) Centroid and Noise In this case, we set the new value of the outlier coordinate $a_{o,k}^{\text{new}}$ as

$$a_{o,k}^{\text{new}} = c_l^{(k)} + r, \tag{6}$$

where r is randomly drawn from a certain distribution such that the distribution of the distances from the cluster samples to the cluster centroid is not modified too much. More precisely, with $f_{d_{C_l}}$ the distribution of the distances between the samples in cluster C_l and its centroid, and $f_{d_{C_l}}^{\text{new}}$ the same distribution after modifying the outlier coordinate $a_{o,k}$, we want to make sure that $\text{KL}(f_{d_{C_l}}, f_{d_{C_l}}^{\text{new}}) \leq \varepsilon$, where $\text{KL}(\cdot, \cdot)$ stands for the Kullback-Leibler divergence between the two distributions [7]. In practice, other metrics could be used in this place, such as the Earth-Mover Distance [1, 9], e.g. This approach, as said before, although probably very desirable, is rather difficult to achieve practically, as drawing the noise value r in such a way as described above is difficult.

(b) Flattening to Centroid This case is a direct simplified version of the previous one. Here, we set the new value of $a_{o,k}^{\text{new}}$ as

$$a_{o,k}^{\text{new}} = c_l^{(k)}. \tag{7}$$

While this approach has a very clear advantage of being simple, it might lead to moving the outlier "too close" to the centroid. Remember here that the centroid is likely not an actual sample from the data, and we are thus inserting a sample with unseen before coordinates, in this cluster.

(c) Flattening to a Cluster Element Value Finally, this third approach is probably a good compromise of simplicity of execution and low distortion of the data. Here, we set the new value of $a_{o,k}^{\text{new}}$ as

$$a_{o,k}^{\text{new}} = a_{i,k} \text{ with } a_{i,k} \text{ drawn at random from } C_l. \tag{8}$$

In this case, we thus draw from the existing sample values from this cluster for this specific dimension. This ensures that we avoid disturbing too much the existing samples within the cluster, while moving effectively the outlier within the cluster. Once we have the outlier(s) moved back into the most appropriate cluster, we can assume that the isolated individuals these samples were representing are no longer as easy to re-identify as before. We can move on to the second part of this work: anonymizing the data within a cluster.

4.2 Anonymizing the data within a cluster

The idea behing this approach is to provide some methods to anonymize the data (by modifying its inherent values) while retaining, as in the previous sections, the structure of it — in the same data clustering sense as in the rest of the paper. For this, we propose two approaches that aim at anonymizing the data within a cluster (and not the whole cluster in itself), so that samples (individuals) within a cluster cannot be re-identified easily. The two approaches are relatively destructive on purpose, in order to provide a means of destroying "intelligently" the data for some specific data fields. The first one relies on flattening the required dimensions: if a specific data field is deemed sensitive, it can be summarized, for a single cluster, by a single value. The second approach is a lot less destructive, and tries to preserve the overall cluster statistics as much as possible, by randomizing the values within a cluster for all the samples so that the cluster remains similar.

Flattening dimensions Referring back to Table 1, it is for example likely that the data in the "Sensitive" field, namely the Condition, would need to be modified before this data is released. For such cases where destructive data alterations are desirable or even needed, it would be possible to replace the sensitive values by empty or unusable ones. This would effectively destroy some of the data statistics and structures within the cluster considered. But in the cases where one would want to preserve some of this information in order to keep some structure within the cluster, the question becomes: how do we modify this data so that it is as close to destroying it as possible, while maintaining the cluster structure/statistics? The proposed straightforward way to do this is to "flatten" the sensitive field (dimension) to the value of the centroid. The effect of collapsing a specific dimension is illustrated on Figure 1. In effect, what happens for each cluster C_l is

$$\forall k \in \mathcal{S},\ \forall \mathbf{t}_i \in C_l, a_{i,k} = c_l^{(k)}, \tag{9}$$

where $\mathcal{S}$ is the set of the considered sensitive fields to be anonymized "destructively". While this approach effectively destroys the data structure within the cluster to some extent, there is a risk that the cluster is already initially as on Figure 1; this could likely happen if the initial clustering of the data samples is efficient already in the first place. The flattening procedure proposed here would

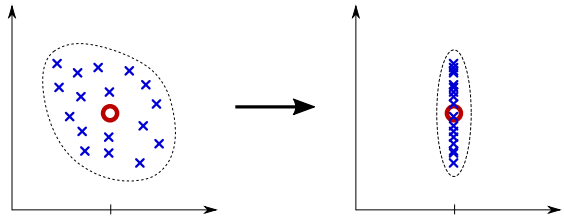

Fig. 1. Example of collapsing of one dimension to the centroid value. This obviously breaks cluster distribution and distances to the centroid.

then have no effect and one could argue that the anonymization is not carried out.

This is unlikely to happen for all clusters at the same time, although this is obviously highly data-dependent. For this reason, we propose the second method, also considered as destructive regarding the data values, but "safer" in this respect.

Shuffling data around This second method is about preserving the intra-cluster data structure as much as possible, while still modifying the sample values as much as possible. This approach is the most costly in terms of computations and general costs. In this case, we shuffle the samples (on one dimension at a time only) around the centroid. In effect, for a cluster C_l,

$$\forall k \in \mathcal{S},\ \forall \mathbf{t}_i \in C_l, a_{i,k} = a_{i,k}^{\text{new}} \text{ s.t. } \text{KL}(f_{d_{C_l}}, f_{d_{C_l}}^{\text{new}}) \leq \varepsilon, \tag{10}$$

where, as before, $f_{d_{C_l}}$ is the distribution of the distances between the samples in cluster C_l and its centroid, and $f_{d_{C_l}}^{\text{new}}$ the same distribution after modifying the coordinate $a_{i,k}$, and $a_{i,k}^{\text{new}} \in \mathbb{X}^{(k)}$ is the new value for the coordinate k of sample $\mathbf{t}_i$ in C_l. This approach is illustrated on Figure 2, where one can see that the overall effect is to "shuffle around" within the cluster, while preserving the distances between the samples in the cluster and the cluster centroid.

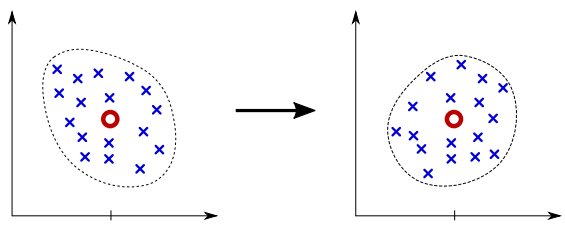

Fig. 2. Example of re-distributing the samples within a cluster (or adding noise to them in a controlled fashion): The distribution of the distances to the centroid is preserved and the overall cluster structure is preserved.

Note in this case that we do not try to preserve explicitly the pairwise distances between the samples within a cluster. Such distances will, at the whole cluster level, be preserved somewhat in any case, by preserving the distances to the centroid.

5 Conclusions and Future Work

In this paper, we propose an early version of a data anonymization framework, focusing on making individual re-identification difficult, while preserving clusters/group statistics and structure, over any type of data field (provided it can be abstracted as a metric space). We first develop the means of identifying outliers in terms of clustering the data, and propose ways to modify the data so as to "push back" this outlier with the rest of the crowd. We then propose several methods to "stir" the data within a cluster, effectively modifying the data values completely, but retaining the internal structure of the clusters. This to allow for further data processing at a somewhat global level, while ensuring the privacy of individuals. As can be noted, some of the data alterations proposed in this work are relatively computationally heavy, and currently require many iterations to converge to an acceptable solution (e.g. the case from section 4.2 where one shuffles data around within a cluster so as to minimize the distortions on the distances distributions). Current and future work will focus on developing efficient algorithms to perform the proposed anonymization tasks, and experiment the proposed framework over large data sets composed of very different data fields.

References

1. V.I. Bogachev and A.V. Kolesnikov. The monge-kantorovich problem: achievements, connections, and perspectives. *Russian Math. Surveys*, 67:785–890, 2012.
2. V. Ciriani, S. Capitani di Vimercati, S. Foresti, and P. Samarati. κ-anonymity. In *Secure Data Management in Decentralized Systems*, volume 33 of *Advances in Information Security*, pages 323–353. Springer US, 2007.
3. G. Cybenko. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.
4. C. Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, volume 4978 of *LNCS*, pages 1–19. Springer, April 2008.
5. P. Kieseberg, H. Hobel, S. Schrittwieser, E. Weippl, and A. Holzinger. *Protecting Anonymity in Data-Driven Biomedical Science*, pages 301–316. 2014.
6. P. Kieseberg, B. Malle, P. Frühwirt, E. Weippl, and A. Holzinger. A tamper-proof audit and control system for the doctor in the loop. *Brain Informatics*, pages 1–11, 2016.
7. S. Kullback and R.A Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.
8. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond κ-anonymity. *International Conference on Data Engineering (ICDE)*, 0:24, 2006.
9. C. L. Mallows. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, 43(2):508–515, April 1972.