Project N°: 262608

ACRONYM: Data without Boundaries

DELIVERABLE D11.3

Progress and intermediate report on new masking techniques, record linkage for output check

WORK PACKAGE 11

Improved Methodologies for Managing Risks of Access to Detailed OS Data

<table>
<thead>
<tr>
<th>PROJECT START DATE:</th>
<th>1st May 2011</th>
<th>DURATION: 48 Months</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE OF ISSUE OF DELIVERABLE:</td>
<td>February 2015</td>
<td></td>
</tr>
<tr>
<td>DOCUMENT PREPARED BY:</td>
<td>3, 11, 15, 24 and 29</td>
<td>URV, CBS, Destatis, ONS and UoMan</td>
</tr>
</tbody>
</table>

Combination of CP & CSA project funded by the European Community
Under the programme “FP7 - SP4 Capacities”
Priority 1.1.3: European Social Science Data Archives and remote access to Official Statistics
The views and conclusions expressed are those of the author(s) and do not necessarily represent those of the DwB consortium as a whole.
This document consists of several parts that together comprise to deliverable D11.3.

The first part consists of a report on new masking techniques, written by URV.

The second part consists of a report on record linkage for risk assessment, by the University of Manchester.

For the sake of scientific relevance and accuracy - and though we acknowledge that this deliverable (D11.3) should include the interim results of the work carried out by CBS, Destatis and ONS on "Guidelines for output checking" (incl. the rule of thumb and underlying concepts) - we believe it would not make sense to make those results publicly available "as is" insofar as they are - by definition - intermediate and incomplete. Therefore, we have decided to postpone their publication into a more finalized version as part of deliverable D11.8, so that they can serve as a sound basis and DwB-endorsed recommendations for work and discussion with and amongst the targeted stakeholders.
NEW MASKING TECHNIQUES
Progress and Intermediate Report on New Masking Techniques, Project “DwB-Data without Boundaries” (FP7-INFRA-2010-262608)

Josep Domingo-Ferrer
Universitat Rovira i Virgili
Department of Computer Engineering and Mathematics
UNESCO Chair in Data Privacy
Av. Països Catalans 26, E-43007 Tarragona, Catalonia
e-mail: josep.domingo@urv.cat

March 2013
## Contents

1 Overview ................................................. 4

2 Microaggregation- and Permutation-Based Anonymization of Mobility Data 6
   2.1 Introduction ............................................. 6
   2.1.1 Contribution and plan of this article .......... 7
   2.2 Related work ............................................. 9
   2.2.1 $k$-Anonymity and microaggregation ......... 9
   2.2.2 Trajectory anonymization ...................... 10
   2.2.3 Trajectory similarity measures ........ 12
   2.3 Utility and privacy requirements .............. 13
   2.3.1 Desirable utility features ................. 13
   2.3.2 Specific utility measures .................. 14
   2.3.3 Adversarial model and target privacy properties 15
   2.3.4 Discussion on privacy models .......... 16
   2.4 Distance between trajectories ............... 17
   2.4.1 Contemporary and synchronized trajectories 17
   2.4.2 Definition and computation of the distance ..... 19
   2.4.3 Intuition and rationale of the distance .... 20
   2.5 Anonymization methods ......................... 20
   2.5.1 The SwapLocations method ............... 21
   2.5.2 The ReachLocations method ............. 23
   2.5.3 Complexity of SwapLocations and ReachLocations 25
   2.6 Privacy guarantees ............................... 26
   2.6.1 Privacy guarantees of SwapLocations .... 26
   2.6.2 Privacy guarantees of ReachLocations ...... 27
   2.7 Experimental results .............................. 28
   2.7.1 Results on synthetic data ........... 28
   2.7.2 Results on real-life data ........... 37
   2.8 Conclusions and future work ............. 40

3 Anonymization methods for taxonomic microdata 43
   3.1 Introduction ............................................. 43
   3.1.1 Contribution and plan of this chapter ...... 45
   3.2 Background .............................................. 45
Chapter 1

Overview

This document contains the first part of DwB Deliverable 11.3.

We next list the masking methods presented in the following chapters of along with the publications they have yielded (all of which bear funding acknowledgment to DwB):

Chapter 2  This chapter presents masking methods to protect mobility data (trajectories). The material in this Chapter was published in


Chapter 3  This chapter presents a masking method to anonymize taxonomic data, which is based on a new concept called marginality that exploits the underlying hierarchy. The material in this Chapter was published in


Chapter 4  This chapter describes a new privacy model called probabilistic $k$-anonymity and a computational method to achieve it for microdata sets. The advantage of probabilistic $k$-anonymity is that yields the same probability $1/k$ of re-identification as $k$-anonymity but entails a lower information loss. The material in this Chapter was published in


Chapter 5  This chapter presents a new computational method to achieve $\varepsilon$-differential privacy in queries to interactive databases. The advantage of the new method
is that, unlike typical methods proposed to achieve $\varepsilon$-differential privacy (e.g. Laplace noise addition), it does not need finding the sensitivity of the query function. The contents of this chapter have been published in


**Chapter 6** This chapter presents a methodology for preventing direct and direct discrimination in data mining. The contents of this chapter have been published in:

- Sara Hajian and Josep Domingo-Ferrer, “A methodology for direct and indirect discrimination prevention in data mining”, *IEEE Transactions on Knowledge and Data Engineering*, to appear.

**Chapter 7** This chapter explores the synergies and shortcomings of combining privacy preservation and discrimination protection in a set of microdata. Its contents have been published in

Chapter 2

Microaggregation- and Permutation-Based Anonymization of Mobility Data

2.1 Introduction

Various technologies such as GPS, RFID, GSM, etc., can sense and track the whereabouts of objects (cars, parcels, people, etc.). On the other hand, the current storage capacities allow collecting such object movement data in huge spatio-temporal databases. Analyzing this kind of databases containing the trajectories of objects can lead to useful and previously unknown knowledge. Therefore, it is beneficial to share and publish such databases and let the analysts derive useful knowledge from them — knowledge that can be applied, for example, to intelligent transportation, traffic monitoring, urban and road planning, supply chain management, sightseeing improvement, etc.

However, the privacy of individuals may be affected by the publication or the outsourcing of databases of trajectories. Several kinds of privacy threats exist. Simple de-identification realized by removing identifying attributes is insufficient to protect the privacy of individuals. The biggest threat with trajectories is the “sensitive location disclosure”. In this scenario, knowing the times at which an individual visited a few locations can help an adversary to identify the individual’s trajectory in the published database, and therefore learn the individual’s other locations at other times. Privacy preservation in this context means that no sensitive location ought to be linkable to an individual.

The risk of sensitive location disclosure is also affected by how much the adversary knows. The adversary may have access to auxiliary information [58], also sometimes called side knowledge, background knowledge or external knowledge. The adversary can link such background knowledge obtained from other sources to information in the published database. Estimating the amount and extent of auxiliary information available to the adversary is a challenging task.
There are quite a few differences between spatio-temporal data and microdata, *i.e.*, records describing individuals in a standard database with no movement data. One real difference becomes apparent when considering privacy. Unfortunately, the traditional anonymization and sanitization methods for microdata [39] cannot be directly applied to spatio-temporal data without considerable expense in computation time and information loss. Hence, there is a need for specific anonymization methods to thwart privacy attacks and therefore reduce privacy risks associated with publishing trajectories.

Trajectories can be modeled and represented in many ways [37]. Without loss of generality, we consider a trajectory to be a timestamped path in a plane. By assuming movements on the surface of the Earth, the altitude of each location visited by a trajectory stays implicit; it could be explicitly restored if the need arose. More formally, let \( \text{timestamped location} \) be a triple \( (t, x, y) \) with \( t \) being a timestamp and \( (x, y) \) a location in \( \mathbb{R}^2 \). Intuitively, the timestamped location denotes that at time \( t \) an object is at location \( (x, y) \).

**Definition 1** (Trajectory). A trajectory is an ordered set of timestamped locations

\[
T = \{(t_1, x_1, y_1), \ldots, (t_n, x_n, y_n)\},
\]

where \( t_i < t_{i+1} \) for all \( 1 \leq i < n \).

**Definition 2** (Sub-trajectory). A trajectory \( S = \{(t'_1, x'_1, y'_1), \ldots, (t'_m, x'_m, y'_m)\} \) is a sub-trajectory of \( T \) in Expression (2.1), denoted \( S \preceq T \), if there exist integers \( 1 \leq i_1 < \ldots < i_m \leq n \) such that \( (t'_j, x'_j, y'_j) = (t_{i_j}, x_{i_j}, y_{i_j}) \) for all \( 1 \leq j \leq m \).

Hereinafter, we will use **triple** as a synonym for timestamped location. When there is no risk of ambiguity, we also say just “location” to denote a timestamped location.

### 2.1.1 Contribution and plan of this article

We present two heuristic methods for preserving the privacy of individuals when releasing trajectories. Both of them exactly preserve original locations in the sense that the anonymized trajectories contain no fake, perturbed or generalized trajectories. The first heuristic is based on microaggregation [23] of trajectories and permutation of locations. Microaggregation has been successfully used in microdata anonymization to achieve \( k \)-anonymity [87, 95, 24]. We use it here for trajectory \( k \)-anonymity (whereby an adversary cannot decide which of \( k \) anonymized trajectories corresponds to an original trajectory which she partly knows), first by grouping the trajectories into clusters of size at least \( k \) based on their similarity and then transforming via location permutation the trajectories inside each cluster to preserve privacy. The second heuristic aims no longer at trajectory \( k \)-anonymity, but at location \( k \)-diversity (whereby knowing a sub-trajectory \( S \) of a certain original trajectory \( T \) allows an adversary to discover a location in \( T \setminus S \) with probability no greater than \( 1/k \)); this second heuristic is based on location permutation and its strong point is that it takes reachability constraints into account: movement between locations must follow the edges of an underlying graph (e.g., urban pattern) so that not all locations are reachable from any given location. Experimental results show that achieving trajectory \( k \)-anonymity with reachability constraints
may not be possible without discarding a substantial fraction of locations, typically those which are rather isolated. This is the motivation for our second heuristic: it still considers reachability but it reduces the number of discarded locations by replacing $k$-anonymity at the trajectory level by $k$-diversity at the location level.

For clustering purposes, we propose a new distance for trajectories which naturally considers both spatial and temporal coordinates. Our distance is able to compare trajectories that are not defined over the same time span, without resorting to time generalization. Our distance function can compare trajectories that are timewise overlapping only partially or not at all. It may seem at first sight that the distance computation is exponential in terms of all considered trajectories, but we show that it is in fact computable in polynomial time.

We present empirical results for the two proposed heuristics using synthetic data and also real-life data. We theoretically and experimentally compare our first heuristic with a recent trajectory anonymization method called $(k, \delta)$-anonymity [2] also aimed at trajectory $k$-anonymity without reachability constraints. Theoretical results show that the privacy preservation of our first method is the same as that of $(k, \delta)$-anonymity but dealing with trajectories not having the same time span. For the second heuristic involving reachability constraints, no comparable counterparts seem to exist in the literature.

In summary, our contributions are:

- A distance measure for trajectories which naturally considers both spatial and temporal aspects of trajectories, is computable in polynomial time, and can cluster trajectories not defined over the same time span;

- Two methods for trajectory anonymization which yield anonymized trajectories formed by fully accurate true original locations and whose distinctive features are:
  - The first method aims at trajectory $k$-anonymity.
  - The second method takes reachability constraints into account, and it tries to reduce the fraction of discarded locations by replacing trajectory $k$-anonymity with location $k$-diversity;

- Empirical results on synthetic and real-life data portraying the performance of the two above methods. Both methods are confronted with $(k, \delta)$-anonymity [2], which has some comparable features.

The rest of this chapter is organized as follows. Section 2.2 reviews related work. Section 2.3 describes the utility features, the adversarial model being considered and the target privacy properties. Our new distance between trajectories is described in Section 2.4. Our two new anonymization methods are specified in Section 2.5. Their privacy guarantees are examined in Section 2.6. Section 2.7 reports on empirical results. Conclusions are drawn in Section 6.10.
2.2 Related work

Most trajectory anonymization methods in the literature rest on ideas inspired by microdata anonymization. We first recall two key microdata anonymization concepts: \(k\)-anonymity and microaggregation. We then review the trajectory anonymization literature. We end this section by reviewing similarity distance measures and clustering algorithms for trajectories.

2.2.1 \(k\)-Anonymity and microaggregation

A lot of work has been done in anonymizing microdata and relational/transactional databases [87, 95, 100, 63, 104, 61, 72, 29, 107]; see also the recent survey [39]. A usual goal in anonymization is to achieve \(k\)-anonymity [87, 95], which is the “safety in numbers” notion.

Anonymizing a microdata set by mere suppression of direct identifiers (e.g., names, passport numbers) is not enough to prevent privacy disclosure. Indeed, other attributes, called quasi-identifier attributes, are often available in the data set such that their combination allows re-identifying the individual to whom a record corresponds: for example, Sweeney [96] found that “87% of the US population is uniquely identified by \{date of birth, gender, 5-digit ZIP\}”. Re-identification allows linking the confidential attributes in a record (e.g., salary or health condition) with a specific individual, and this constitutes a disclosure.

An anonymized microdata set is said to satisfy \(k\)-anonymity if each combination of quasi-identifier attribute values is shared by at least \(k\) records. Therefore, this property guarantees that an adversary is unable to identify the individual to whom an anonymized record corresponds with probability higher than \(1/k\).

\(k\)-Anonymity cannot be directly achieved with spatio-temporal data, because any point or time can be regarded as a quasi-identifier attribute [2]. Direct \(k\)-anonymization would require a set of original trajectories to be transformed into a set of anonymized trajectories such that each of the latter is identical to at least \(k - 1\) other anonymized trajectories. This would obviously cause a huge information loss.

Generalization was the computational approach originally proposed to achieve \(k\)-anonymity [87, 95]. Later, Zhang et al. introduced the permutation-based approach [107], that has the advantage of not being constrained by domain generalization hierarchies. In [24] it was shown that \(k\)-anonymity could also be achieved through microaggregation of quasi-identifiers. Microaggregation [23] works in two stages:

1. **Clustering.** The original records are partitioned into clusters based on some similarity measure (some kind of distance) among the records with the restriction that each cluster must contain at least \(k\) records. Several microaggregation heuristics are available in the literature, some yielding fixed-size clusters all of size \(k\), except perhaps one (e.g. the MDAV heuristic [24]), and some yielding variable-size clusters, of sizes between \(k\) and \(2k - 1\) (e.g. \(\mu\)-Approx [27]). We will use fixed-size microaggregation.

2. **Anonymization.** Each cluster is anonymized individually. Anonymization of a cluster may be based on an aggregation operator like the average [23] or the me-
dian [24], which is used to compute the cluster centroid; each record in the cluster is then replaced by the cluster centroid. Anonymization of a cluster can also be achieved by replacing the records in the cluster with synthetic or partially synthetic data; this is called hybrid data microaggregation [19] or condensation [4].

To use microaggregation on trajectories, we need a distance measure to compute the similarity between trajectories. We deal with possible distances later in this chapter.

2.2.2 Trajectory anonymization

Just like in microdata records, suppressing direct identifiers from trajectories is not enough for privacy [57]. Consequently, several anonymity notions and methods for trajectories have been proposed [45, 41, 46, 12, 81, 11, 2, 70, 97, 66, 71, 106, 67, 3, 47, 48]. Among those works, we next review the ones that are most similar to our approach, and we highlight our comparative advantages. Other comparisons of several trajectory anonymization methods can be found in [11, 3].

Closest to our approach is the notion of \((k, \delta)\)-anonymity \([2, 3]\). In the original method –Never Walk Alone (NWA) \([2]\)–, the set of trajectories is partitioned into disjoint subsets in which trajectories begin and end at roughly the same time; then trajectories within each set are clustered using the Euclidean distance. In the follow-up method –Wait For Me (W4M) \([3]\)–, the original trajectories are clustered using the edit distance on real sequences (EDR) \([16]\). Both approaches proceed by anonymizing each cluster separately. Two trajectories \(T_1\) and \(T_2\) are said to be co-localized with respect to \(\delta\) in a certain time interval \([t_1, t_n]\) if for each triple \((t, x_1, y_1)\) in \(T_1\) and each triple \((t, x_2, y_2)\) in \(T_2\) with \(t \in [t_1, t_n]\), it holds that the spatial Euclidean distance between both triples is not greater than \(\delta\). Anonymity in this context means that each trajectory is co-localized with at least \(k - 1\) other trajectories. Anonymization is achieved by spatial translation of trajectories inside a cluster of at least \(k\) trajectories having the same time span. In the special case when \(\delta = 0\), the method produces one centroid/average trajectory that represents each and all trajectories in the cluster. Ad hoc preprocessing and outlier removal facilitate the process. Utility is evaluated in terms of trajectory distortion and impact on the results of range queries. The problem with the NWA method is that partitioning the set of all trajectories into subsets sharing the same time span may produce too many subsets with too few trajectories inside each of them; clearly, a subset with less than \(k\) trajectories cannot be \(k\)-anonymized. Also, setting a value for \(\delta\) may be awkward in many applications, e.g. trajectories recorded using RFID technology. In Section 2.7 we present an empirical comparison between this method and our two heuristics. Our heuristics avoid the above subset problem by considering all trajectories together whatever their time span; they also achieve co-localization without requiring a \(\delta\) radius. The W4M method is similar in clustering to our clustering, although it uses the EDR distance between trajectories, which has the shortcomings discussed further below.

Another \(k\)-anonymity based notion for trajectories consisting of ranges of points and ranges of times has been proposed in [70] and [71]. It uses clustering to minimize the “log cost metric”; this balances the spatial and temporal translations with user-provided weights. Minimizing the log cost therefore maximizes utility. The clusters are
anonymized by matching points of the trajectories and generalizing them into minimum bounding boxes. Unmatched points are suppressed and so are some trajectories. The anonymized data are not released; instead, synthetic “atomic” trajectories (having unit x-range, y-range and time range) are generated by sampling the bounding boxes. This approach does not release standard trajectories but only trajectories with unit ranges. In comparison, we are able to produce synthetic trajectories, with the advantage that we obtain anonymized trajectories formed by true original locations.

In [67], $k$-anonymity means that an original trajectory $T$ is generalized into a trajectory $g(T)$ (without the time information) in such a way that $g(T)$ is a sub-trajectory of the generalizations of at least $k - 1$ other original trajectories. Ignoring the time information during anonymization and complex plane tessellations used to achieve the $k$-anonymity are the main drawbacks of this method. Utility is measured by comparing clustering results. In our approach, we avoid complex tessellations and our main advantage in comparison to this anonymization scheme is that we do not ignore temporal information.

Another proposal for achieving $k$-anonymity of trajectories by means of generalization is [48]. The difference lies in the way generalization is performed: the authors propose a technique called local enlargement, guaranteeing that user locations are enlarged just enough to reach $k$-anonymity, which improves utility of the anonymized trajectories. In contrast, we preserve original locations, without generalizing them; our notion of trajectory $k$-anonymity is, however, reformulated as discussed below.

The adapted $k$-anonymity notion for trajectories in [106] is stated in terms of a bipartite attack graph relating original and anonymized trajectories such that the graph is symmetric and the degree of each vertex representing an anonymized trajectory is at least $k$. The quasi-identifiers used to define identities are the times of the positions in a trajectory, and the anonymity is achieved by generalizing points of trajectories into areas on the grid. An information loss metric defined for such areas is used to evaluate the utility of the anonymized data.

Some approaches assume that the data owner anonymizing the database knows exactly what the adversary’s knowledge is. If the adversary is assumed to know different parts of trajectories, then those are removed from the published data [97]. However, this work only considers sequential place visitation without real timestamps. If the adversary is assumed to use some prediction of continuation of a trajectory based on previous path and speed, then uncertainty-aware path cloaking [46, 47] can suppress these trajectories; this however results in high information loss.

In contrast to these methods, we perform traditional microaggregation over all original trajectories —we do not specially and separately consider trajectories having the same time span and we consider trajectories over locations, not ranges, without stripping the time information. We publish synthetic trajectories which are analogous to condensed or hybrid microdata [4, 19]. However, our synthetic trajectories are formed by locations covered by the original trajectories. This means that the location points of our anonymized trajectories remain on the underlying network map.

Additional related work about anonymization of spatio-temporal data can be found in the literature about location privacy, focused on applications such as privacy-aware location-based services (LBS) or privacy-aware monitoring of continuously moving objects. Location privacy in the LBS-setting was first proposed in [40]. See [75, 49] for
recent papers on location privacy, in which mobile objects protect the privacy of their continuous movement. Location privacy is enforced on individual sensitive locations or unlinked locations in an on-line mode; often, data are anonymized on a per-request basis and in the context of obtaining a location-based service. In this article, we focus on off-line publishing whole spatio-temporal databases rather than protecting specific individuals from LBS providers or on-line movement monitoring. In general, a solution to location privacy is not a solution for publishing anonymized trajectories, and vice versa.

2.2.3 Trajectory similarity measures

As argued in Section 2.2.1 above, using microaggregation for trajectory \( k \)-anonymization requires a distance function to measure the similarity between trajectories. Such a distance function must consider both space and time. Although most spatial distances can be extended into spatio-temporal distances by adding a time co-ordinate to spatial points, it is not obvious how to balance the weight of spatial and temporal dimensions. Furthermore, not all similarity measures for trajectories are suitable for comparing trajectories for anonymization purposes. The requirement for anonymization is not just similarity regarding shape, but also spatial and temporal closeness. Some typical distances for trajectories include the Euclidean distance, the Hausdorff distance [90], the Fréchet distance [7], the turning point distance [8], and distances based on time series [60] — e.g., dynamic time warping (DTW), short time series (STS) — and on edit distance [16] — e.g., edit distance with real penalty (ERP), longest common sub-sequence (LCSS), and the edit distance on real sequences (EDR) discussed next.

The edit distance on real sequences (EDR) [16] is the number of insert, delete, or replace operations that are needed to change one sequence into another. If \( P \) and \( Q \) are two sequences of \( m \) and \( n \) triples, respectively, where each triple \( \lambda \) has three attributes — x-position \( \lambda.x \), y-position \( \lambda.y \) and time \( \lambda.t \) — the distance \( EDR(P, Q) \) is defined as:

\[
EDR(P, Q) = \begin{cases} 
\max\{m, n\} & \text{if } m = 0 \text{ or } n = 0 \\
\min\{match(p_1, q_1) + EDR(Rest(P), Rest(Q)),
1 + EDR(Rest(P), Q), 1 + EDR(P, Rest(Q))\} & \text{otherwise}
\end{cases}
\]

where \( p_1 \) and \( q_1 \) are the first elements of a given sequence, \( Rest(\cdot) \) is a function that returns the input sequence without the first element, and where \( match(p, q) := 0 \) if \( p \) and \( q \) are “close”, that is, they satisfy either \( |p.x - q.x| \leq \epsilon \) and \( |p.y - q.y| \leq \epsilon \) for some parameter \( \epsilon \) [16] or \( |p.x - q.x| \leq \Delta.x, |p.y - q.y| \leq \Delta.y, \text{ and } |p.t - q.t| \leq \Delta.t \) for a triple of parameters \( \Delta \) [3]; otherwise, \( match(p, q) := 1 \). This definition of \( match \) means that the cost for one insert, delete, or replace operation in EDR is 1 if \( p \) and \( q \) are not “close”.

EDR has been employed for anonymization in [3]. However, the edit distance and variations thereof are not suitable to guide clustering for anonymization purposes. Indeed, Figure 2.1 shows trajectories with different degrees of “closeness” to trajectory A, but whose EDR distance from A is the same in all cases. When timestamps are considered, the situation is even worse.

In Section 2.4, we define a distance measure which is better suited for anonymization clustering: it can compare trajectories defined over different time spans and even
trajectories that are time-wise non-overlapping.

Figure 2.1: Trajectories B, C, D, E are placed at varying “closeness” from A, yet their EDR distance from A is 3 in all cases. We assume that the first point of A matches the first point of each of B, C, D, E; also, second points are assumed to match each other, and the same for third points.

2.3 Utility and privacy requirements

Every trajectory anonymization algorithm must combine utility and privacy. However, utility and privacy are two largely antagonistic concepts. What is useful in a set of trajectories is application-dependent, so for each utility feature probably a different anonymization algorithm is needed.

2.3.1 Desirable utility features

The utility features that are usually considered in trajectory anonymization are: (i) trajectory length preservation, (ii) trajectory shape preservation, (iii) trajectory time preservation, and (iv) minimization of the number of discarded locations. We include two additional utility features that are particularly meaningful in urban scenarios:

- **Location preservation.** This essentially means that no fake or inaccurate locations are used to replace original locations; otherwise put, locations in the anonymized trajectories should be locations visited by the original trajectories, without any generalization or accuracy loss. Preserving original locations helps answering several queries that may not be responded by generalization methods [67] or some microaggregation methods [2, 3]: i) what is the ranking of original (non-removed) locations, from most visited to least visited?; ii) in which original (non-removed) locations did two or more mobile objects meet?, etc. On the other hand, if trajectory anonymization rests on replacing true locations with
fake locations, an adversary can distinguish the latter from the former and discard fake locations. Hence, location preservation is desirable for both utility and privacy reasons.

- **Reachability.** In the second proposed heuristic, easy reachability between two successive locations in each anonymized trajectory is enforced. This means that the distance from the $i$-th location to the $i + 1$-th location on an anonymized location following the underlying network of streets and/or roads should be at most $R^s$, where $R^s$ is a preset parameter. Like location preservation, this is as good for utility as it is for privacy: if the adversary sees that reaching the $i + 1$-th location from the $i$-th takes a long trip across streets and roads, she will guess that the section between those two locations was not present in any original trajectory.

### 2.3.2 Specific utility measures

Basic utility measures are the number of removed trajectories and the number of removed locations, whether during pre-processing, clustering or cluster anonymization.

The distortion of the trajectory shape is another utility measure, which can be captured with the space distortion metric [2, Sec.VI.B]. This metric also allows accumulating the total space distortion of all anonymized trajectories from original ones.

**Definition 3** (Space distortion metric [2]). The space distortion of an anonymized trajectory $T^*$ with respect to its original trajectory $T$ at time $t$ when $T$ has triple $(t, x, y)$ and $T^*$ has possible triple $(t, x^*, y^*)$, is

$$\text{SD}_t(T, T^*) = \begin{cases} 
\Delta((x, y), (x^*, y^*)) & \text{if } (x^*, y^*) \text{ is defined at } t \\
\Omega & \text{otherwise}
\end{cases}$$

where $\Delta$ is a distance (e.g., Euclidean), and $\Omega$ a constant that penalizes for removed locations. The space distortion of an anonymized trajectory $T^*$ from its original $T$ is then

$$\text{SD}(T, T^*) = \sum_{t \in T} \text{SD}_t(T, T^*) ,$$

where $T$ are all the timestamps where $T$ is defined. In particular, if $T$ is discarded during anonymization, $T^*$ is empty, and so $\text{SD}(T, T^*) = n\Omega$, where $n = |T|$ is the number of locations of $T$. In this way, the space distortion of a set of trajectories $T$ from its anonymized set $T^*$ is easily defined as

$$\text{TotalSD}(T, T^*) = \sum_{T \in T} \text{SD}(T, T^*) ,$$

where $T^* \in T^*$ (which may be empty) corresponds to $T \in T$.

Another way to measure utility is by comparing the results between queries performed on both the original data set $T$ and the anonymized data set $T^*$. Intuitively, when results on both data sets are similar for a large and diverse number of queries, the anonymized data set can be regarded as preserving the utility of the original data.
set. The challenge of this utility measure is the selection of queries, which is usually application-dependent or even user-dependent, i.e., two different users are likely to perform different queries on the same trajectory data set.

In [99] six types of spatio-temporal range queries were introduced, aimed at evaluating the relative position of a moving object with respect to a region $R$ in a time interval $[t_b, t_e]$. We have used these queries in our experimental work, even though they were designed for use on uncertain trajectories (see Definition 4) rather than synthetic trajectories.

**Definition 4 (Uncertain trajectory).** Given a trajectory $T$ and an uncertainty space threshold $\sigma$, an uncertain trajectory $U(T, \sigma)$ is defined as the pair $< T, \sigma >$, where $(t, x, y) \in U(T, \sigma)$ if and only if $\exists x', y'$ such that $(t, x', y') \in T$ and the Euclidean distance between $(x, y)$ and $(x', y')$ is not greater than $\sigma$.

**Definition 5 (Possible motion curve).** A possible motion curve $PMC^T$ of an uncertain trajectory $U(T, \sigma)$ is an ordered set of timestamped locations

$$PMC^T = \{(t_1, x_1, y_1), \ldots, (t_n, x_n, y_n)\}$$

such that $(t_i, x_i, y_i) \in U(T, \sigma)$ for all $1 \leq i \leq n$.

In short, a possible motion curve defines one of the possible trajectories that an object moving along an uncertain trajectory could follow. Unlike in [99], our anonymized trajectories are not uncertain; hence, we will only use the two spatio-temporal range queries proposed in that paper that can be adapted to non-uncertain trajectories:

- **Sometime_Definitely_Inside($T, R, t_b, t_e$)** is true if and only if there exists a time $t \in [t_b, t_e]$ at which every possible motion curve $PMC^T$ of an uncertain trajectory $U(T, \sigma)$ is inside region $R$. For a non-uncertain $T$, the previous condition can be adapted as: if and only if there exists a time $t \in [t_b, t_e]$ at which $T$ is inside $R$.

- **Always_Definitely_Inside($T, R, t_b, t_e$)** is true if and only if at every time $t \in [t_b, t_e]$, every possible motion curve $PMC^T$ of an uncertain trajectory $U(T, \sigma)$ is inside region $R$. For a non-uncertain $T$, the previous condition becomes: if and only if at every time $t \in [t_b, t_e]$, trajectory $T$ is inside $R$.

### 2.3.3 Adversarial model and target privacy properties

In our adversarial model, the adversary has access to the published anonymized set of trajectories $T^*$. Furthermore, the adversary also knows that every location $\lambda \in T^*$ must be in the original set of trajectories $T$. Note that this adversary’s knowledge makes an important difference from previous adversarial models [2, 71, 67, 106], because in our model the linkage of some location with some user reveals the exact location of this user rather than a generalized or perturbed location.

Further, the method used for transforming the original set of trajectories $T$ into $T^*$ is assumed known by the adversary. However, this does not include the method parameters or the seeds for pseudo-random number generators, which are considered...
secret. Indeed, the two methods we are proposing rely on random permutations of locations and random selection of trajectories during the clustering process, and such randomness is in practice implemented using pseudo-random number generators. If an adversary knew the seeds of the generators, she could easily reconstruct the original trajectories from the anonymized trajectories.

Finally, the adversary also knows a sub-trajectory $S$ of some original target trajectory $T \in \mathcal{T}$ ($S \preceq T$) and knows that the anonymized version of $T$ is in $\mathcal{T}^*$. As in previous works, we consider that every location in $\mathcal{T}$ is sensitive, i.e. for any location, learning that a specific user visited it represents useful knowledge for the adversary.

Then, we identify two attacks:

1. Find a trajectory $T^* \in \mathcal{T}^*$ that is the anonymized version of $T$.
2. Given a location $\lambda \notin S$, determine whether $\lambda \in T$.

If the adversary succeeds in the first attack of linking a trajectory $T^*$ with the target $T$, the second is not trivial, because in general the locations in $T^*$ will not be those in $T$, but it is indeed easier. This means that both attacks are not independent. However, the second attack can trivially succeed even if the first attack does not: if all anonymized trajectories cross the same location $\lambda$ and $\lambda \notin S$, the adversary knows that $\lambda \in T$. As we show below, both attacks are related to the two well-known privacy notions of $k$-anonymity [87, 95] and $\ell$-diversity [63], respectively.

**Definition 6** (Trajectory $p$-privacy). Let $Pr_{T^*|[T|S]}$ denote the probability of the adversary’s correctly linking the anonymized trajectory $T^* \in \mathcal{T}^*$ with $T$ given the adversary’s knowledge $S \preceq T$. Then, trajectory $p$-privacy is met when $Pr_{T^*|[T|S]} \leq p$ for every trajectory $T \in \mathcal{T}$ and every subset $S \preceq T$.

**Definition 7** (Trajectory $k$-anonymity). Trajectory $k$-anonymity is achieved if and only if trajectory $\frac{1}{k}$-privacy is met.

**Definition 8** (Location $p$-privacy). Let $Pr_{\lambda|[T|S]}$ denote the probability of the adversary’s success in correctly determining a location $\lambda \in T \setminus S$, given the adversary’s knowledge $S \preceq T$. Then, location $p$-privacy is met when $Pr_{\lambda|[T|S]} \leq p$ for every triple $(T, S, \lambda)$ such that $T \in \mathcal{T}$, $S \preceq T$ and $\lambda \notin S$.

**Definition 9** (Location $k$-diversity). Location $k$-diversity is achieved if and only if location $\frac{1}{k}$-privacy is met.

### 2.3.4 Discussion on privacy models

Achieving straightforward trajectory $k$-anonymity, where each anonymized trajectory would be identical to $k - 1$ other anonymized trajectories, would in general cause a huge information loss. This is why some other trajectory $k$-anonymity definitions under different assumptions have been proposed.

The $(k, \delta)$-anonymity definition [2, 3] relies on the uncertainty inherent to trajectory data recorded by technologies like GPS. However, it may be hardly applied when accurate data sets of trajectories are needed. Furthermore, in order to achieve $(k, \delta)$-anonymity, the $k$ identical anonymized trajectories should be defined roughly in the

16
same interval of time and they must contain the same number of locations. Such constraints are indeed hard to meet.

Another trajectory $k$-anonymity definition can be found in [81]. In this work, trajectory $k$-anonymity is achieved when there are at least $k$ anonymized trajectories in $\mathcal{T}^*$ having an anonymized version of $T$ as a sub-trajectory. Although this definition ignores the time dimension, it does not require the length of the $k$ anonymized trajectories to be equal. However, suppose that the adversary has a trajectory $T$ consisting of only one location, an individual’s home; whatever the anonymization method, the anonymized version of $T$ is likely to be very similar to $T$. This means that there will be $k$ anonymized trajectories containing the single location of $T$. However, not all these anonymized trajectories start at the single location of $T$. Since an individual’s home is likely to be the first location of any individual’s original trajectory, those anonymized trajectories that do not start at the single location of $T$ (just pass through it) can be filtered out by an adversary and only the remaining trajectories are considered. The same filtering process can be performed if the adversary knows locations where the individual has never been. In this way, using side knowledge the adversary identifies less than $k$ anonymized trajectories compatible with the original trajectory $T$. Hence, this definition may not actually guarantee $k$-anonymity in the sense of Definition 7.

In conclusion, different levels of privacy can be provided according to different assumptions on the original data, the anonymized data, and the adversary’s capabilities. We defined above trajectory $p$-privacy (Definition 6) and location $p$-privacy (Definition 8) in order to capture two different privacy notions when the original locations are preserved.

### 2.4 Distance between trajectories

Clustering trajectories requires defining a similarity measure —a distance between two trajectories. Because trajectories are distributed over space and time, a distance that considers both spatial and temporal aspects of trajectories is needed. Many distance measures have been proposed in the past for both trajectories of moving objects and for time series but, as discussed in Section 2.2, most of them are ill-suited to compare trajectories for anonymization purposes. Therefore we define a new distance which can compare trajectories that are only partially or not at all timewise overlapping. We believe this is necessary to cluster trajectories for anonymization. We need some preliminary notions.

#### 2.4.1 Contemporary and synchronized trajectories

**Definition 10 ($p\%$-contemporary trajectories).** Two trajectories

$$T_i = \{(t_1^i, x_1^i, y_1^i), \ldots, (t_n^i, x_n^i, y_n^i)\}$$

and

$$T_j = \{(t_1^j, x_1^j, y_1^j), \ldots, (t_m^j, x_m^j, y_m^j)\}$$
are said to be \( p \%-\text{contemporary} \) if

\[
p = 100 \cdot \min\left(\frac{I}{t_m - t_1}, \frac{I}{t_j - t_1}\right)
\]

with \( I = \max(\min(t_m, t_j) - \max(t_1, t_1), 0) \).

Intuitively, two trajectories are 100%-contemporary if and only if they start at the same time and end at the same time; two trajectories are 0%-contemporary if and only if they occur during non-overlapping time intervals. Denote the overlap time of two trajectories \( T_i \) and \( T_j \) as \( ot(T_i, T_j) \).

**Definition 11** (Synchronized trajectories). Given two \( p \%-\text{contemporary} \) trajectories \( T_i \) and \( T_j \) for some \( p > 0 \), both trajectories are said to be synchronized if they have the same number of locations timestamped within \( ot(T_i, T_j) \) and these correspond to the same timestamps. A set of trajectories is said to be synchronized if all pairs of \( p \%-\text{contemporary} \) trajectories in it are synchronized, where \( p > 0 \) may be different for each pair.

If we assume that between two locations of a trajectory, the object is moving along a straight line between the locations at a constant speed, then interpolating new locations is straightforward. Trajectories can be then synchronized in the sense that if one trajectory has a location at time \( t \), then other trajectories defined at that time will also have a (possibly interpolated) location at time \( t \). This transformation guarantees that the set of new locations interpolated in order to synchronize trajectories is of minimum cardinality. Algorithm 1 describes this process. The time complexity of this algorithm is \( O(|TS|^2) \) where \( |TS| \) is the number of different timestamps in the data set.

**Algorithm 1** Trajectory synchronization

**Input:** \( T = \{T_1, \ldots, T_N\} \) a set of trajectories to be synchronized, where each \( T_i \in T \) is of the form:

\[
T_i = \{(t_i^1, x_i^1, y_i^1), \ldots, (t_i^n, x_i^n, y_i^n)\};
\]

1. Let \( TS = \{t_j^1 | (t_j^1, x_j^1, y_j^1) \in T_i : T_i \in T\} \) be all timestamps from all locations of all trajectories;
2. for all \( T_i \in T \) do
3.   for all \( ts \in TS \) with \( t_1^1 < ts < t_n^i \) do
4.     if location having timestamp \( ts \) is not in \( T_i \) then
5.       insert new location in \( T_i \) having the timestamp \( ts \) and coordinates interpolated from the two timewise-neighboring locations;
6.     end if
7.   end for
8. end for
2.4.2 Definition and computation of the distance

**Definition 12** (Distance between trajectories). Consider a set of synchronized trajectories $\mathcal{T} = \{T_1, \ldots, T_N\}$ where each trajectory is written as

$$T_i = \{(t^i_1, x^i_1, y^i_1), \ldots, (t^i_n, x^i_n, y^i_n)\}.$$  

The distance between trajectories is defined as follows. If $T_i, T_j \in \mathcal{T}$ are $p\%$-contemporary with $p > 0$, then

$$d(T_i, T_j) = \frac{1}{p} \sqrt{\sum_{t \in \text{ot}(T_i, T_j)} \left(\frac{(x^i_t - x^j_t)^2 + (y^i_t - y^j_t)^2}{|\text{ot}(T_i, T_j)|^2}\right)}.$$  

If $T_i, T_j \in \mathcal{T}$ are $0\%$-contemporary but there is at least one subset of $\mathcal{T}$

$$\mathcal{T}^k(ij) = \{T^{ij}_1, T^{ij}_2, \ldots, T^{ij}_{n^{ij}_k}\} \subseteq \mathcal{T}$$

such that $T^{ij}_1 = T_i$, $T^{ij}_{n^{ij}_k} = T_j$ and $T^{ij}_\ell$ and $T^{ij}_{\ell+1}$ are $p_\ell\%$-contemporary with $p_\ell > 0$ for $\ell = 1$ to $n^{ij}_k - 1$, then

$$d(T_i, T_j) = \min_{\mathcal{T}^{k(ij)}} \left(\sum_{\ell=1}^{n^{ij}_k-1} d(T^{ij}_\ell, T^{ij}_{\ell+1})\right).$$  

Otherwise $d(T_i, T_j)$ is not defined.

The computation of the distance between every pair of trajectories is not exponential as it could seem from the definition. Polynomial-time computation of a distance graph containing the distances between all pairs of trajectories can be done as follows.

**Definition 13** (Distance graph). A distance graph is a weighted graph where

(i) Nodes represent trajectories,

(ii) two nodes $T_i$ and $T_j$ are adjacent if the corresponding trajectories are $p\%$-contemporary for some $p > 0$, and

(iii) the weight of the edge $(T_i, T_j)$ is the distance between the trajectories $T_i$ and $T_j$.

Now, given the distance graph for $\mathcal{T} = \{T_1, \ldots, T_N\}$, the distance $d(T_i, T_j)$ for two trajectories is easily computed as the minimum cost path between the nodes $T_i$ and $T_j$, if such path exists. The inability to compute the distance for all possible trajectories (the last case of Definition 12) naturally splits the distance graph into connected components. The connected component that has the majority of the trajectories must be kept, while the remaining components represent outlier trajectories that are discarded in order to preserve privacy. Finally, given the connected component of the distance graph having the majority of the trajectories of $\mathcal{T}$, the distance $d(T_i, T_j)$ for any two trajectories on this connected component is easily computed as the minimum cost path between the nodes $T_i$ and $T_j$. The minimum cost path between every pair of nodes can be computed using the Floyd-Warshall algorithm [36] with computational cost $O(N^3)$, i.e., in polynomial time.
2.4.3 Intuition and rationale of the distance

In order to deal with the time dimension, our distance measure applies a linear penalty of \( \frac{1}{p} \) to those trajectories that are \( p\%-\)contemporary. This means that, the closer in time are two trajectories, the shorter is our distance between both. It should be remarked that we choose a linear penalty because the Euclidean distance is also linear in terms of the spatial coordinates and the Euclidean distance is the spatial distance measure we consider by default in this work. Other distances and other penalties might be chosen, e.g. \( \frac{1}{p^2} \).

A problem appears when considering 0\%-contemporary trajectories. How can two non-overlapping trajectories be penalized? A well-known strategy is to give a weight to the time dimension and another weight to the spatial dimension. By doing so, the time distance and the spatial distance can be computed separately, and later be merged using their weights. However, determining proper values for these weights is a challenging task.

Anyway, the following lemma guarantees that, whenever we consider two trajectories at minimum distance for clustering, they do have some overlap.

**Lemma 1.** Any two trajectories in data set \( T \) at minimum distance are \( p\%-\)contemporary with \( p > 0 \).

**Proof:** Consider a trajectory \( T_i \in T \) and another trajectory \( T_j \in T \) at minimum distance from \( T_i \). Assume that \( T_i \) and \( T_j \) are not \( p\%-\)contemporary with \( p > 0 \). Then, since the distance between \( T_i \) and \( T_j \) is defined, according to Definition 12 a subset of distinct trajectories \( T(ij) = \{ T_{i}^{1,j}, T_{2}^{1,j}, \ldots, T_{n^{ij}}^{1,j} \} \subseteq T \) must exist such that \( T_{1}^{1,j} = T_i \), \( T_{n^{ij}}^{1,j} = T_j \), and \( T_{\ell}^{1,j} \) and \( T_{\ell+1}^{1,j} \) are \( \ell\%-\)contemporary with \( \ell > 0 \) for \( \ell = 1 \) to \( n^{ij} - 1 \), and

\[
d(T_i, T_j) = \sum_{\ell=1}^{n^{ij}-1} d(T_{\ell}^{ij}, T_{\ell+1}^{ij})
\]

Then \( d(T_i, T_j) > d(T_{\ell}^{ij}, T_{\ell+1}^{ij}) \) for all \( \ell \) from 1 to \( n^{ij} - 1 \) (strict inequality holds because all trajectories in \( T(ij) \) are distinct). Thus, we reach the contradiction that \( d(T_i, T_j) \) is not minimum. Hence, the lemma must hold. \( \square \)

2.5 Anonymization methods

We present two anonymization methods, called SwapLocations and ReachLocations, respectively, both of which yield anonymized trajectories formed by original locations. The first of them is partially based on microaggregation [23] of trajectories and partially based on permutation of locations. The second method is based on permutation of locations. The main difference between the SwapTriples method [21] and the two new methods we propose here is that the latter effectively guarantee trajectory \( k\)-anonymity (SwapLocations) or location \( k\)-diversity (ReachLocations). To that end, an original triple is discarded if it cannot be swapped randomly with another triple drawn from a set of \( k - 1 \) other original triples.
Our two methods differ from each other in several aspects. The first method assumes an unconstrained environment, while the second one considers an environment with mobility constraints, like an underlying street or road network. SwapLocations effectively achieves trajectory $k$-anonymity. ReachLocations provides higher utility by design, but regarding privacy, it offers location $k$-diversity instead of trajectory $k$-anonymity. A common feature of both methods is that locations in the resulting anonymized trajectories are true, fully accurate original locations, i.e. no fake, generalized or perturbed locations are given in the anonymized data set of trajectories.

2.5.1 The SwapLocations method

Algorithm 2 describes the process followed by the SwapLocations method in order to anonymize a set of trajectories. First, the set of trajectories is partitioned into several clusters. Then, each cluster is anonymized using the SwapLocations function in Algorithm 3. We should remark here that we only consider trajectories for which the distance to other trajectories can be computed using the distance in Definition 12. Otherwise said, given the distance graph $G$ (Definition 13), our distance measure can only be used within one of the connected components of $G$; obviously, we take the trajectories in the largest connected component of $G$. It should also be remarked that Algorithm 1 is only used for computing the distance between trajectories. Once a cluster $C$ is created, the anonymization algorithm works over the original triples of the trajectories in $C$, and not over the triples created during synchronization.

We limit ourselves to clustering algorithms which try to minimize the sum of the intra-cluster distances or approximate the minimum and such that the cardinality of each cluster is $k$, with $k$ an input parameter; if the number of trajectories is not a multiple of $k$, one or more clusters must absorb the up to $k-1$ remaining trajectories, hence those clusters will have cardinalities between $k+1$ and $2k-1$. This type of clustering is precisely the one used in microaggregation [23]. The purpose of minimizing the sum of the intra-cluster distances is to obtain clusters as homogeneous as possible, so that the subsequent independent treatment of clusters does not cause much information loss. The purpose of setting $k$ as the cluster size is to fulfill trajectory $k$-anonymity, as shown in Section 2.6.1. We employ any microaggregation heuristic for clustering purposes (see Section 2.2 and details in Section 2.5.3 below).

The SwapLocations function (Algorithm 3) begins with a random trajectory $T$ in $C$. The function attempts to cluster each unswapped triple $\lambda$ in $T$ with another $k-1$ unswapped triples belonging to different trajectories such that: i) the timestamps of these triples differ by no more than a time threshold $R_t$ from the timestamp of $\lambda$; ii) the spatial coordinates differ by no more than a space threshold $R_s$. If no $k-1$ suitable triples can be found that can be clustered with $\lambda$, then $\lambda$ is removed; otherwise, random swaps of triples are performed within the formed cluster. Randomly swapping this cluster of triples guarantees that any of these triples has the same probability of remaining in its original trajectory or becoming a new triple in any of the other $k-1$ trajectories. Note that Algorithm 3 guarantees that every triple $\lambda$ of every trajectory $T \in C$ will be swapped or removed.

The SwapLocations function specified by Algorithm 3 swaps entire triples, that is, time and space coordinates. The following example illustrates the advantages of
Algorithm 2 Cluster-based trajectory anonymization(\(\mathcal{T}, R^t, R^s, k\))

**Input:**

i) \(\mathcal{T} = \{T_1, \ldots, T_N\}\) a set of original trajectories such that \(d(T_i, T_j)\) is defined for all \(T_i, T_j \in \mathcal{T}\); ii) \(R^t\) a time threshold and \(R^s\) a space threshold;

1: Use any clustering algorithm to cluster the trajectories of \(\mathcal{T}\), while minimizing the sum of intra-cluster distances measured with the distance of Definition 12 and ensuring that minimum cluster size is \(k\);
2: Let \(C_1, C_2, \ldots, C_{n_T}\) be the resulting clusters;
3: for all clusters \(C_i\) do
4: \(C^*_i = \text{SwapLocations}(C_i, R^t, R^s)\); \hspace{1cm} // Algorithm 3
5: end for
6: Let \(\mathcal{T}^* = C^*_1 \cup \cdots \cup C^*_{n_T}\) be the set of anonymized trajectories.

Algorithm 3 SwapLocations(\(C, R^t, R^s\))

**Input:**

i) \(C\) a cluster of trajectories to be transformed, ii) \(R^t\) a time threshold and \(R^s\) a space threshold;

1: Mark all triples in trajectories in \(C\) as “unswapped”;
2: Let \(T\) be a random trajectory in \(C\);
3: for all “unswapped” triples \(\lambda = (t_\lambda, x_\lambda, y_\lambda)\) in \(T\) do
4: \(U = \{\lambda\}\); \hspace{1cm} // Initialize \(U\) with \(\{\lambda\}\)
5: for all trajectories \(T'\) in \(C\) with \(T' \neq T\) do
6: Look for an “unswapped” triple \(\lambda' = (t_{\lambda'}, x_{\lambda'}, y_{\lambda'})\) in \(T'\) minimizing the intra-cluster distance in \(U \cup \{\lambda'\}\) and such that:
   \[
   |t_{\lambda'} - t_\lambda| \leq R^t
   \]
   \[0 \leq \sqrt{(x_{\lambda'} - x_\lambda)^2 + (y_{\lambda'} - y_\lambda)^2} \leq R^s;
   \]
7: if \(\lambda'\) exists then
8: \(U \leftarrow U \cup \{\lambda'\}\);
9: else
10: Remove \(\lambda\) from \(T\);
11: Go to line 3 in order to analyze the next triple \(\lambda\);
12: end if
13: end for
14: Randomly swap all triples in \(U\);
15: Mark all triples in \(U\) as “swapped”;
16: end for
17: Remove all “unswapped” triples in \(C\);
18: return \(C\).

swapping time together with space.

**Example 1.** Imagine John attended one day the political protests in Tahrir Square, Cairo, Egypt, but he would not like his political views to become broadly known. Assume John’s trajectory is anonymized and published. Assume further that an adversary
knows the precise time John left his hotel in the morning, say 6:36 AM, e.g. because the adversary has bribed the hotel concierge into recording John’s arrival and departure times. Now:

- If SwapLocations swapped only spatial coordinates, the adversary could re-identify John’s trajectory as one starting with a triple \((6:36 \text{ AM}, x_h’, y_h’).\) Furthermore, \((x_h’, y_h’)\) must be a location within a distance \(R^s\) from the hotel coordinates \((x_h, y_h)\), although the adversary does not know the precise value of \(R^s\). The re-identified trajectory would contain all true timestamps of John’s original trajectory (because they would not have been swapped), and spatial coordinates within distance \(R^s\) of John’s really visited spatial coordinates. Hence, it would be easy to check whether John was near Tahrir Square during that day. Without swapping times, privacy protection can only be obtained by taking \(R^s\) large enough so that within distance \(R^s\) of the original locations visited by John there are several semantically different spatial coordinates. To explain what we mean by semantic difference, assume \((x, y)\) is Tahrir Square and the trajectory anonymizer guarantees that he has taken \(R^s\) large enough so that \((x, y)\) could be swapped with some spatial coordinates \((x’, y’)\) off Tahrir Square; even if \((x’, y’)\) turned out to be still within Tahrir Square, John could claim to have been off Tahrir Square; the adversary could not disprove such a claim, because in fact \((x, y)\) could be at a distance \(R^s\) from \((x’, y’)\) and hence outside the Square. However, a large \(R^s\) means a large total space distortion.

- If entire triples are swapped, as actually done by SwapLocations, the adversary can indeed locate an anonymized trajectory containing (not necessarily starting with) triple \((6:36 \text{ AM}, x_h, y_h).\) However, there is only a chance \(1/k\) that this triple was not swapped from another of the \(k – 1\) original trajectories with which John’s original trajectory was clustered. Similarly, the other triples in the anonymized trajectory containing \((6:36 \text{ AM}, x_h, y_h)\) have also most likely “landed” in that anonymized trajectory as a result of a swap with some location in some of the \(k – 1\) original trajectories clustered with John’s. Hence, John’s trajectory is cloaked with \(k – 1\) other trajectories. We will prove in Section 2.6.1 that this guarantees trajectory \(k\)-anonymity in the sense of Definition 7. In particular, the triple \((t, x, y)\) corresponding to John at Tahrir Square will appear in one of the \(k\) anonymized trajectories, unless that triple has been removed by the SwapLocations function because it was unswappable (the smaller \(R^t\) and \(R^s\), the more likely it is for the triple to be removed).

2.5.2 The ReachLocations method

The ReachLocations method, described in Algorithm 4, takes reachability constraints into account: from a given location, only those locations at a distance below a threshold following a path in an underlying graph (e.g., urban pattern or road network) are considered to be directly reachable. Enforcing such reachability constraints while requiring full trajectory \(k\)-anonymity would result in a lot of original locations being discarded. To avoid this, trajectory \(k\)-anonymity is changed by another useful privacy definition: location \(k\)-diversity.
Computationally, this means that trajectories are not microaggregated into clusters of size $k$. Instead, each location is $k$-anonymized independently using the entire set of locations of all trajectories. To do so, a cluster $C_\lambda$ of “unswapped” locations is created around a given location $\lambda$, i.e., $\lambda \in C_\lambda$. The cluster $C_\lambda$ is constrained as follows: i) it must have the lowest intra-cluster distance among those clusters of $k$ “unswapped” locations that contain the location $\lambda$; ii) it must have locations belonging to $k$ different trajectories; and iii) it must contain only locations at a path from $\lambda$ at most $R_s$ long and with timestamps differing from $t_\lambda$ at most $R_t$. Then, the spatial coordinates $(x_\lambda, y_\lambda)$ are swapped with the spatial coordinates of some random location in $C_\lambda$ and both locations are marked as “swapped”. If no cluster $C_\lambda$ can be found, the location $\lambda$ is removed from the data set and will not be considered anymore in the subsequent anonymization. This process continues until no more “unswapped” locations appear in the data set.

It should be remarked that, according to Algorithm 4, two successive locations $\lambda_i^j$ and $\lambda_{i+1}^j$ of an original trajectory $T_i$ may be cloaked with respective sets of $k-1$ locations belonging to different sets of $k-1$ original trajectories; this is why we cannot speak of trajectory $k$-anonymity, see the example below.

**Example 2.** Consider $k-1$ trajectories within city $A$, $k-1$ trajectories within city $B$ and one trajectory $T_{AB}$ crossing from $A$ to $B$. When applying ReachLocations, the initial locations of $T_{AB}$ are swapped with locations of trajectories within $A$, whereas the final locations of $T_{AB}$ are swapped with locations of trajectories within $B$. Imagine that an adversary knows a sub-trajectory $S$ of $T_{AB}$ containing one location $\lambda_A$ in $A$ and one location $\lambda_B$ in $B$. Assume $\lambda_A$ and $\lambda_B$ are not removed by ReachLocations anonymization. Now, the adversary will know that the anonymized trajectory $T_{AB}^*$ corresponding to $T_{AB}$ is the only anonymized trajectory crossing from $A$ to $B$. Thus, there is no trajectory $k$-anonymity, even if the adversary will be unable to determine the exact locations of $T_{AB} \setminus S$, because each of them has been swapped within a set of $k$ locations.

Algorithm 4 swaps only spatial coordinates instead of full triples. We show in the example below that this is enough for ReachLocations to achieve location $k$-diversity (we have shown above that it cannot achieve trajectory $k$-anonymity anyway). If swapping time coordinates is not beneficial in terms of privacy guarantees, they should not be swapped, because the fact that anonymized trajectories preserve the original sequence of timestamps of original trajectories increases their utility.

**Example 3.** Let us resume Example 1, but now assume that ReachLocations is used instead of SwapLocations to anonymize trajectories. In this case, the adversary will find an anonymized trajectory starting with $(6:36\ AM, x'_h, y'_h)$. This anonymized trajectory will contain all true timestamps of John’s original trajectory. However, the spatial coordinates appearing in any location of this re-identified trajectory are John’s original spatial coordinates with a probability at most $1/k$. We will prove in Section 2.6.2 below that this guarantees location $k$-diversity in the sense of Definition 9. If we want to prevent the adversary from making sure that John visited Tahrir Square, we should take $R_s$ large enough (the discussion in Example 1 about the protection afforded by a large $R_s$ when time is not swapped is valid here).
2.5.3 Complexity of SwapLocations and ReachLocations

We first give a complexity assessment of SwapLocations and ReachLocations assuming that the distance graph mentioned in Section 2.4.2 has been precomputed and is available. This is reasonable, because the distance graph needs to be computed only once, while the anonymization methods may need to be run several times (e.g. with different parameters). Regarding SwapLocations, we have:

- Algorithm 2 can use any fixed-size microaggregation heuristic for clustering (e.g. MDAV in [24]). Most microaggregation heuristics have quadratic complexity, that is $O(N^2)$, where $N$ is the number of trajectories.

- Algorithm 2 calls the procedure SwapLocations once for each resulting cluster, that is, $O(N/k)$ times.

- In the worst case, the complexity of procedure SwapLocations (Algorithm 3) is proportional to the number of locations of the longest trajectory in $C$, say $O(n_{max})$. For each location, a search of another location for swapping is performed among the other $k - 1$ trajectories. The number of candidates for swapping is $O((k - 1)n_{max})$. Hence, the complexity of SwapLocations is $O((k - 1)n_{max}^2)$.

- The total complexity of the method is thus

$$O(N^2) + O(N/k) \cdot O((k - 1)n_{max}^2) = O(N^2) + O(Nn_{max}^2) \quad (2.3)$$

Regarding the complexity of ReachLocations, we have

- Algorithm 4 has an external loop which is called $N$ times, where $N$ is the number of trajectories in $T$. For each trajectory, a swap is attempted for each of its unswapped locations. Hence the algorithm performs $O(Nn_{max})$ swaps, where $n_{max}$ is the number of locations in the longest trajectory.

- Each swap involves forming a cluster which $k - 1$ locations selected from $TL$, which takes time proportional to the total number of locations in $TL$, that is, $O(Nn_{max})$.

- Hence, the total complexity of the method is $O(N^2n_{max}^2)$.

By comparing the last expression and Expression (2.3), we see that both SwapLocations and ReachLocations are quadratic in $N$ and quadratic in $n_{max}$, but ReachLocations is slower. Such complexity motivates the following two comments related to scalability:

- If the number of trajectories $N$ in the original data set is very large, quadratic complexity may be very time consuming. In this case, a good strategy is to use some blocking technique to split the original data set into several subsets of trajectories, each of which should be anonymized separately.
• $n_{\text{max}}$ being large may be less problematic than $N$ being large, provided that only a small fraction of trajectories have $n_{\text{max}}$ or close to $n_{\text{max}}$ locations. If a lot of trajectories are very long, a good strategy would be to split each of these into two or more trajectories and anonymize them independently.

Finally, in case we add the time complexity of the computation of the distance graph mentioned in Section 2.4.2 (which is $O(N^3)$ using the Floyd-Warshall algorithm), the time complexities of both SwapLocations and ReachLocations become $O(N^3) + O(N n_{\text{max}}^2)$ and $O(N^3) + O(N^2 n_{\text{max}}^2)$, respectively.

2.6 Privacy guarantees

2.6.1 Privacy guarantees of SwapLocations

The main difference between the SwapTriples method in [21] and the SwapLocations method here is that, in the latter, no original location survives unswapped in an anonymized trajectory.

Proposition 1. Let $S \preceq T_S$ be the adversary’s knowledge of a target original trajectory $T_S$ and $\lambda_1, \lambda_2, \cdots, \lambda_{|S|}$ be all triples in $S$. For every trajectory $T_i$, the probability that the triple $\lambda$ in $S$ appears in the anonymized version $T_i^*$ of $T_i$ produced by SwapLocations is:

$$\Pr(\lambda \in T_i^* | \lambda \in S) = \begin{cases} \frac{1}{k} & \text{if } T_S \text{ and } T_i \text{ lie in the same cluster} \\ 0 & \text{otherwise.} \end{cases}$$

Proof: By construction of Algorithm 3, if $T_S$ and $T_i$ do not lie in the same cluster, there is no possibility of swapping triples between them. Hence, in this case, $\Pr(\lambda \in T_i^* | \lambda \in S) = 0$.

Let $T_1, T_2, \cdots, T_k \in \mathcal{T}$ be $k$ trajectories that are anonymized together in the same cluster by the SwapLocations method. Without loss of generality, let us assume that $T_S = T_1$. By construction of Algorithm 3, for every $1 \leq i \leq k$, $\Pr(\lambda \in T_i^* | \lambda \in T_1)$ is 0 if $\lambda$ was removed, $\frac{1}{k}$ otherwise. Note that a swapping option is to swap a triple with itself, that is, not to swap it. Since it does not make sense to consider removed triples in $S$, we conclude that $\Pr(\lambda_j \in T_i^* | \lambda_j \in T_1) = \frac{1}{k}$, $\forall 1 \leq j \leq |S|, 1 \leq i \leq k$ and, in consequence, $\Pr(\lambda_j \in T_i^* | \lambda_j \in S) = \frac{1}{k}$, $\forall 1 \leq j \leq |S|, 1 \leq i \leq k$. □

Theorem 1. The SwapLocations method achieves trajectory $k$-anonymity.

Proof: By Proposition 1, any sub-trajectory $S' \preceq S \preceq T_1$ has the same probability of being a sub-trajectory of $T_1^*$ than of being a sub-trajectory of any of the $k - 1$ trajectories $T_2^*, \cdots, T_k^*$. Thus, given $S$, an adversary is not able to link $T_1$ with $T_1^*$ with probability higher than $\frac{1}{k}$. Therefore, SwapLocations satisfies $\frac{1}{k}$-privacy according to Definition 6; according to Definition 7, it also satisfies trajectory $k$-anonymity. □
2.6.2 Privacy guarantees of ReachLocations

We show below that ReachLocations provides location $k$-diversity.

**Proposition 2.** Any triple $\lambda$ in an original trajectory $T$ appears in the anonymized trajectory $T^*$ corresponding to $T$ obtained with ReachLocations if and only if $\lambda$ was not removed and was swapped with itself, which happens with probability at most $1/\kappa$.

**Proof:** Let us prove the necessity implication. By construction of Algorithm 4, any triple $\lambda$ whose spatial coordinates (point) cannot be swapped within a cluster $C \cup \{\lambda\}$ containing $k$ different points belonging to $k$ different trajectories is removed and does not appear in the set of anonymized trajectories. Further, the only way for a non-removed triple $\lambda \in T$ to survive unaltered in $T^*$ is precisely that its point is swapped with itself, which happens with probability $1/\kappa$. Therefore, to survive unaltered in $T^*$, a triple in $T$ needs to avoid removal and to have its point swapped with itself, which happens with probability at most $1/\kappa$.

Now let us prove the sufficiency implication. Assume that $\lambda = (t, x, y) \in T$ appears in $T^*$ without having been swapped with itself. Then, by construction of ReachLocations, $\lambda \in T^*$ must have been formed as the result of swapping a triple $(t, x', y') \in T$ with a triple $(t', x, y)$ from another original trajectory, where $(x', y') \neq (x, y)$. But then $T$ would contain two triples with the same timestamp $t$ and different spatial locations, which is a contradiction. $\square$

**Theorem 2.** The ReachLocations method achieves location $k$-diversity.

**Proof:** Assume the adversary knows a sub-trajectory $S$ of an original trajectory $T$. The sequence of timestamps in $S$ allows the adversary to re-identify the anonymized trajectory $T^*$ corresponding to $T$ (because the timestamp sequence is preserved). By Proposition 2, any triple $\lambda \in T^* \setminus S$ belongs to $T \setminus S$ with probability at most $1/\kappa$. Now, consider a triple $\lambda = (t, x, y) \in T^* \setminus S$, where $T^*$ is an anonymized trajectory different from $T^*$. The probability that $\lambda$ came to $T^* \setminus S$ from $T \setminus S$ is the probability that $\lambda$ was swapped and swapping did not alter it. This probability is zero, because swaps preserve time coordinates but take place only between triples having different space coordinates. Hence, in terms of Definition 6, $\text{Pr}_{\lambda}[T \setminus S] \leq 1/k$ for every triple $(T, S, \lambda)$ such that $T \in T$, $S \preceq T$ and $\lambda \notin S$. $\square$

Note that the previous proof also implies that, even if a triple $\lambda = (t, x, y) \notin S$ is shared by $M > 1$ anonymized trajectories, the probability of $\lambda \in T \setminus S$ remains at most $1/k$. What can be inferred by the adversary, however, is that $M$ original trajectories (in general not the ones corresponding to the $M$ anonymized trajectories) visited spatial coordinates $(x, y)$ at possibly different times. Indeed, $(t, x, y)$ can be obtained by swapping $(t', x, y)$ and $(t', x', y')$ for any $t'$ such that $|t' - t| \leq R^t$ and for any $(x', y') \neq (x, y)$ at path distance at most $R^d$. If $M$ is the total number of anonymized trajectories, then the adversary can be sure that original trajectory $T$ visited spatial coordinates $(x, y)$ at some time $t'$ such that $|t' - t| \leq R^t$. Such inference by the adversary does not violate location $k$-diversity: violation would require guessing both the spatial and temporal coordinates of a triple in $T \setminus S$. Of course, the time threshold $R^t$ must be taken large enough so that the time coordinate $t$ is sufficiently protected.
2.7 Experimental results

We implemented SwapLocations and ReachLocations. SwapLocations performs clustering of trajectories using the partitioning step of the MDAV microaggregation heuristic [24]. We used two data sets in our experiments:

- **Synthetic data set.** We used Brinkhoff’s generator [14] to generate 1,000 synthetic trajectories which altogether visit 45,505 locations in the German city of Oldenburg. Synthetic trajectories generated with Brinkhoff’s generator have also been used in [2, 70, 71, 106]. We used this data set mainly for comparing our methods with \((k, \delta)\)-anonymity [2]. The number of trajectories being moderate, we were able to run in reasonable time the methods to be compared with a large number of different parameter choices. Another advantage is that the street graph of Oldenburg was available, which is necessary to run ReachLocations. The downside of this data set having a moderate number of trajectories is that these are rather sparse, which causes the relative distortion in the anonymized data set to be substantial, no matter the method used. Anyway, this is not a serious problem to compare methods with each other.

- **Real-life data set.** We also used a real-life data set of cab mobility traces that were collected in the city of San Francisco [82]. This data set consists of 536 files, each of them containing the GPS coordinates of a cab during a period of time. After a filtering process, we obtained 4582 trajectories and 94 locations per trajectory on average. The advantage of this data set over the synthetic one is that it contains a larger number of trajectories and that these are real ones. Then, we show through a real example how appropriate is our distance metric for trajectory clustering. Also, we present utility measures on the SwapLocations method for this real-life data set using different space thresholds. The weakness of this data set is that it cannot be used for ReachLocations, because it does not include the underlying street graph of San Francisco.

2.7.1 Results on synthetic data

For the sake of reproducibility, we indicate the parameters we used in Brinkhoff’s generator to generate our Oldenburg synthetic data set: 6 moving object classes and 3 external object classes; 10 moving objects and 1 external object generated per timestamp; 100 timestamps; speed 250; and “probability” 1,000. This resulted in 1,000 trajectories containing 45,405 locations. The maximum trajectory length was 100 points, the average length was 45.4 locations, and the median length was 44 locations.

Implementation details of our methods

We have introduced a new distance measure between trajectories used by the SwapLocations proposal during the clustering process. As mentioned in Section 2.5.1 above, our distance function can only be used within one of the connected components of the distance graph \(G\). During the construction of the distance graph for the synthetic data we found 11 connected components, 10 of them of size 1. Therefore, we removed these
10 trajectories in order to obtain a new distance graph with just one connected component. In this way, we preserved 99% percent of all trajectories before the anonymization process. The removed trajectories were in fact trajectories of length one, i.e., with just one location in each one.

The SwapLocations method has been implemented using the following simple microaggregation method for trajectories: first, create clusters of size $k$ with minimum intra-cluster distance and then disperse the up to $k - 1$ unclustered trajectories to existing clusters while minimizing the intra-cluster distance. This algorithm incurs no additional discarding of trajectories.

On the other hand, the ReachLocations method does not remove trajectories, unlike the SwapLocations method. It does, however, remove non-swappable locations, which causes the removal of any trajectory consisting of non-swappable locations only.

**Implementing $(k, \delta)$-anonymity for comparison with our method**

We compared our proposals with $(k, \delta)$-anonymity [2]. Since $(k, \delta)$-anonymity only works over trajectories having the same time span, first a pre-processing step to partition the trajectories is needed. Superimposing the begin and end times of the trajectories through reduction of the time coordinate modulo a parameter $\pi$ does not always yield at least $k$ trajectories having the same time span; it may also happen that a trajectory disappears because the new reduced end time lies before the new reduced begin time.

We have used $\pi = 3$ which kept the maximum (and so discarded the minimum) trajectories. From the 1,000 synthetic trajectories, 40 were discarded because the end time was less than the begin time and 187 were discarded because there were at most 4 trajectories having the same time span. In total, 227 (22.7%) trajectories were discarded just in the pre-processing step. The remaining 773 trajectories were in 32 sets having the same time span, each set containing a minimum of 15 trajectories and 24 on average.

We performed $(k, \delta)$-anonymization for $k = 2, 4, 6, 8, 10, 15$ and $\delta = 0, 1000, 2000, 3000, 4000$ and 5000. Because of the pre-processing step, using a higher $k$ was impossible without causing a significant number of additional trajectories to be discarded.

**Utility comparison**

The performance of our proposals strongly depends on the values of the time and space threshold parameters, denoted as $R^t$ and $R^s$, respectively. In practice, these values must be chosen to maximize utility while affording sufficient privacy protection. Too large thresholds reduce utility (large space distortion if $R^s$ is too high and large time distortion is $R^t$ is too high), but too small thresholds reduce utility because of removal of many unswappable locations. As a rule of thumb, as illustrated in Example 1, the space threshold $R^s$ must be sufficiently large so that within a radius $R^s$ of any spatial location there are sufficiently distinct locations (e.g., if $(x, y)$ lies in Tahrir Square, Cairo, there should be points outside the Square within a radius $R^s$ of $(x, y)$).
In order to compute the total space distortion, a value for $\Omega$ must be chosen and this can be a challenging task. Note that the value of $\Omega$ is application-dependent, e.g. for applications where the distortion should measure the accuracy of trajectories $\Omega$ should be zero (only non-removed triples contribute to $TotalSD$), while for applications that should avoid removing any triples, $\Omega$ should be very high. That is why we propose to compare separately the following three utility properties: i) total space distortion; ii) percentage of removed trajectories; and iii) percentage of removed locations. To do so, we set $\Omega = 0$ when computing the total space distortion. Consequently, the percentage of removed triples as well as the percentage of removed trajectories are considered separately from the total space distortion.

It should be remarked that the computation of the total space distortion of the ReachLocations method is done using the Euclidean distance between locations rather than the distance defined by the reachability constraints (distance on the underlying network). Note that reachability constraints should be considered during the anonymization process but not necessarily when computing the total space distortion.

For successive anonymizations aimed at comparing the SwapLocations and ReachLocations methods with $(k, \delta)$-anonymity, we set $R_t$ and $R_s$ in a way to obtain roughly the same total space distortion values as in $(k, \delta)$-anonymity (see Table 2.1) with $\Omega = 0$. The idea is that, after making sure the three methods achieve roughly the same total space distortion, we will be able to focus on other utility properties like the percentage of removed trajectories and the percentage of removed locations. It should be remarked that our comparison is not entirely fair for any of the three methods because all of them are aimed at achieving different privacy notions. However, we believe that our results are indicative of the weaknesses and the strengths of our proposals.

Table 2.1: Total space distortion (TotalSD) of $(k, \delta)$-anonymity for several parameter values ($e6$ stands for $\times 10^6$)

<table>
<thead>
<tr>
<th>$\delta \setminus k$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>48e6</td>
<td>93e6</td>
<td>120e6</td>
<td>143e6</td>
<td>165e6</td>
<td>199e6</td>
</tr>
<tr>
<td>1,000</td>
<td>19e6</td>
<td>60e6</td>
<td>86e6</td>
<td>109e6</td>
<td>131e6</td>
<td>165e6</td>
</tr>
<tr>
<td>2,000</td>
<td>4e6</td>
<td>32e6</td>
<td>56e6</td>
<td>78e6</td>
<td>99e6</td>
<td>133e6</td>
</tr>
<tr>
<td>3,000</td>
<td>.9e6</td>
<td>14e6</td>
<td>32e6</td>
<td>52e6</td>
<td>71e6</td>
<td>104e6</td>
</tr>
<tr>
<td>4,000</td>
<td>.2e6</td>
<td>5e6</td>
<td>16e6</td>
<td>32e6</td>
<td>48e6</td>
<td>79e6</td>
</tr>
<tr>
<td>5,000</td>
<td>.03e6</td>
<td>2e6</td>
<td>7e6</td>
<td>18e6</td>
<td>31e6</td>
<td>58e6</td>
</tr>
</tbody>
</table>

The above principle of equating the space distortions with $(k, \delta)$-anonymity yields a value for the space threshold $R_s$ in each of SwapLocations and ReachLocations; however, it does not constrain the time threshold, which we set at $R_t = 100$. Regarding $R_s$, we set it to achieve the total space distortions of $(k, \delta)$-anonymity for cluster size $k = \{2, 4, 6, 8, 10, 15\}$ and

$$\delta = \{0, 1000, 2000, 3000, 4000, 5000\}$$

(parameter values considered in Table 2.1). In order to find such space thresholds efficiently, we assume that the total space distortions of our methods define a monotonically increasing function of the space threshold, i.e. the higher the space threshold,
the higher the total space distortion. Under this assumption, we perform a logarithmic search over the set of space thresholds defined by the interval $[0, 10^6]$. The reason behind defining the maximum value for the space threshold as $10^6$ is that it is high enough to achieve low numbers of removed trajectories. Indeed, as shown in Figure 2.2, for both methods there exists a value $R^s_{\text{cutoff}} < 10^6$ such that, for every space threshold $R^s > R^s_{\text{cutoff}}$, neither the total space distortion nor the percentage of removed locations and removed trajectories significantly change. Table 2.2 and Table 2.3 show the values of space thresholds used in each configuration of $(k, \delta)$-anonymity for SwapLocations and ReachLocations, respectively.

Table 2.2: Space thresholds used in SwapLocations to match the total space distortion of each configuration of $(k, \delta)$-anonymity

<table>
<thead>
<tr>
<th>$\delta \backslash k$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>1,000</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>2,000</td>
<td>899</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>3,000</td>
<td>257</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>4,000</td>
<td>86</td>
<td>1390</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>5,000</td>
<td>19</td>
<td>681</td>
<td>2507</td>
<td>$10^6$</td>
<td>$10^6$</td>
<td>$10^6$</td>
</tr>
</tbody>
</table>

As it can be seen in Tables 2.2 and 2.3, we use the maximum value ($10^6$) of the
Table 2.3: Space thresholds used in ReachLocations to match the total space distortion of each configuration of \((k, \delta)\)-anonymity

<table>
<thead>
<tr>
<th>(\delta \setminus k)</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>499875</td>
<td>(10^6)</td>
<td>(10^6)</td>
<td>(10^6)</td>
<td>(10^6)</td>
<td>(10^6)</td>
</tr>
<tr>
<td>1,000</td>
<td>25090</td>
<td>106126</td>
<td>270157</td>
<td>(10^6)</td>
<td>(10^6)</td>
<td>(10^6)</td>
</tr>
<tr>
<td>2,000</td>
<td>4780</td>
<td>52468</td>
<td>93717</td>
<td>151915</td>
<td>249999</td>
<td>(10^6)</td>
</tr>
<tr>
<td>3,000</td>
<td>749</td>
<td>37124</td>
<td>64801</td>
<td>95585</td>
<td>132857</td>
<td>238884</td>
</tr>
<tr>
<td>4,000</td>
<td>136</td>
<td>25540</td>
<td>51089</td>
<td>73088</td>
<td>94465</td>
<td>152862</td>
</tr>
<tr>
<td>5,000</td>
<td>57</td>
<td>18059</td>
<td>39061</td>
<td>58584</td>
<td>79101</td>
<td>113280</td>
</tr>
</tbody>
</table>

space threshold for several configurations. This is because in those configurations the total space distortion caused by the \((k, \delta)\)-anonymity could not be reached by our methods no matter how much we increased the space threshold. Figure 2.3 explains this behavior by showing the values of total space distortion SwapLocations and ReachLocations minus the total space distortion of \((k, \delta)\)-anonymity. With almost every configuration, our methods have a total space distortion lower than the total space distortion of \((k, \delta)\)-anonymity. In the case of SwapLocations, the total space distortion is even much lower.

In general, SwapLocations does not reach high values of the total space distortion because it removes more locations than ReachLocations in order to achieve trajectory \(k\)-anonymity. Note that removing locations does not increase the total space distortion because we are considering \(\Omega = 0\). Tables 2.4 and 2.5 show in detail the percentage of removed trajectories and the percentage of removed locations for different values of \(k = \{2, 4, 6, 8, 10, 15\}\) and \(\delta = \{0, 1000, 2000, 3000, 4000, 5000\}\), for SwapLocations and ReachLocations, respectively.

As it can be seen in Table 2.4, in general SwapLocations removes less trajectories than \((k, \delta)\)-anonymity because SwapLocations can cluster non-overlapping trajectories. Indeed, with \((k, \delta)\)-anonymity 227 trajectories were discarded in the preprocessing step alone because their time span could not match the time span of other trajectories, and additional outlier trajectories were discarded during clustering, up to a total 24% of discarded trajectories. However, SwapLocations removed up to 84% of all locations in the worst cases and thus, it may not be suitable for applications where preserving the number of locations really matters. SwapLocations removes any location whose swapping set \(U\) contains less than \(k\) locations, which is a relatively frequent event when \(k\) trajectories with different lengths are clustered together. As the cluster size \(k\) increases, the length diversity tends to increase and the removal percentage increases. A simple way around the location removal problem is to create clusters that contain trajectories with roughly the same length, even though this may result in a higher total space distortion; higher space distortion is a natural consequence of clustering based on the trajectory length rather than the trajectory distance.

Table 2.5 shows that ReachLocations removes few trajectories when \(\delta\) is small and \(k\) is large. The reason is that, for those parameterizations, \((k, \delta)\)-anonymity introduces so much total space distortion that ReachLocations can afford taking the maximum space threshold \(R^* = 10^6\) without reaching that much distortion. Such a high space
Figure 2.3: Top: total space distortion of SwapLocations minus total space distortion of \((k, \delta)\)-anonymity for several parameter configurations. Bottom: total space distortion of ReachLocations minus total space distortion of \((k, \delta)\)-anonymity for several parameter configurations. The space thresholds defined in Tables 2.2 and 2.3 have been used, respectively.
Table 2.4: Percentage of trajectories (columns labeled with \( T \)) and locations (columns labeled \( L \)) removed by SwapLocations when using time threshold 100, \( k = \{2, 4, 6, 8, 10, 15\} \) and space thresholds that match the space distortion caused by \((k, \delta)\)-anonymity with the previous \( k \)'s and \( \delta = \{0, 1000, 2000, 3000, 4000, 5000\} \). Percentages have been rounded to integers for compactness.

<table>
<thead>
<tr>
<th>( \delta ) ( \setminus k )</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>L</td>
<td>T</td>
<td>L</td>
<td>T</td>
<td>L</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>2000</td>
<td>4</td>
<td>45</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>3000</td>
<td>11</td>
<td>62</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>4000</td>
<td>19</td>
<td>68</td>
<td>5</td>
<td>66</td>
<td>0</td>
<td>69</td>
</tr>
<tr>
<td>5000</td>
<td>32</td>
<td>78</td>
<td>20</td>
<td>73</td>
<td>4</td>
<td>72</td>
</tr>
</tbody>
</table>

threshold allows ReachLocations to easily swap spatial coordinates, so that very few locations need to be removed. Furthermore, the trajectories output by ReachLocations are consistent with the underlying city topology. As said above, the only drawback of this method is that in general it does not provide trajectory \( k \)-anonymity; rather, it provides location \( k \)-diversity.

Spatio-temporal range queries

As stated in Section 2.3.2, a typical use of trajectory data is to perform spatio-temporal range queries on them. That is why we report empirical results when performing the two query types described and motivated in Section 2.3.2: Sometime Definitively Inside (SI) and Always Definitely Inside (AI). We accumulate the number of trajectories in a set of trajectories \( T \) that satisfy the SI or AI range queries using the SQL style code below.

- Query \( Q_1(T, R, t_b, t_e) \):
  
  \[
  \text{SELECT COUNT (* FROM } T \text{ WHERE SI}(T.traj, R, t_b, t_e)
  \]

- Query \( Q_2(T, R, t_b, t_e) \):
  
  \[
  \text{SELECT COUNT (* FROM } T \text{ WHERE AI}(T.traj, R, t_b, t_e)
  \]

Then, we define two different range query distortions:

- \( \text{SID}(T, T^*) = \frac{1}{|T|} \sum_{\forall R, t_b, t_e} \max_{\xi} \left| Q_1(T, R, t_b, t_e) - Q_1(T^*, R, t_b, t_e) \right| \) where \( \xi \) is a set of SI queries as defined in Section 2.3.2 (definition of SI adapted to non-uncertain trajectories).
Table 2.5: Percentage of trajectories (columns labeled with T) and locations (columns labeled L) removed by ReachLocations when using time threshold 100, \( k = \{2, 4, 6, 8, 10, 15\} \) and space thresholds that match the space distortion caused by \((k, \delta)\)-anonymity with the previous \( k \)'s and \( \delta = \{0, 1000, 2000, 3000, 4000, 5000\} \). Percentages have been rounded to integers for compactness.

<table>
<thead>
<tr>
<th>( \delta ) ( k )</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T ) | ( T ) | ( T ) | ( T ) | ( T ) | ( T ) | ( T ) | ( T ) | ( T ) | ( T ) | ( T ) |</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2000</td>
<td>36</td>
<td>27</td>
<td>9</td>
<td>18</td>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>3000</td>
<td>74</td>
<td>38</td>
<td>39</td>
<td>18</td>
<td>28</td>
<td>6</td>
</tr>
<tr>
<td>4000</td>
<td>82</td>
<td>43</td>
<td>65</td>
<td>49</td>
<td>41</td>
<td>40</td>
</tr>
<tr>
<td>5000</td>
<td>84</td>
<td>60</td>
<td>84</td>
<td>53</td>
<td>60</td>
<td>52</td>
</tr>
</tbody>
</table>

- **AID**(\( T, T^* \)) = \( \frac{1}{|\xi|} \sum_{\forall R, t_b, t_e \in \xi} |Q_2(T, R, t_b, t_e) - Q_2(T^*, R, t_b, t_e)| \) max \( (Q_2(T, R, t_b, t_e), Q_2(T^*, R, t_b, t_e)) \) where \( \xi \) is a set of AI queries as defined in Section 2.3.2 (definition of AI adapted to non-uncertain trajectories).

For our experiments with the synthetic data set, we chose random time intervals \([t_b, t_e] \) such that \( 0 \leq t_e - t_b \leq 10 \). Also, we chose random uncertain trajectories with a randomly chosen radius \( 0 \leq \sigma \leq 750 \) as regions \( R \). Actually, 10 and 750 are, respectively, roughly a quarter of the average duration and distance of all trajectories. Note that we used uncertain trajectories only as regions \( R \); however, the methods we are considering in this chapter all release non-uncertain trajectories.

Armed with these settings, we ran 100,000 times both queries \( Q_1 \) and \( Q_2 \) on the original data set and the anonymized data sets provided by SwapLocations, ReachLocations, and \((k, \delta)\)-anonymity; that is, we took a set \( \xi \) with \( |\xi| = 100,000 \). The ideal range query distortion would be zero, which means that query \( Q_i \) for \( i \in 1, 2 \) yields the same result for both the original and the anonymized data sets; in practice, zero distortion is hard to obtain. Therefore, in order to compare our methods against \((k, \delta)\)-anonymity, we use the same parameters of the previous experiments (Tables 2.1, 2.2, and 2.3). We show in Tables 2.6 and 2.7 a comparison of SwapLocations, respectively ReachLocations, against \((k, \delta)\)-anonymity in terms of SID and AID.

It can be seen from Table 2.6 that SwapLocations performs significantly better than \((k, \delta)\)-anonymity for every cluster size and \( \delta \leq 3000 \). On the other hand, Table 2.7 shows that ReachLocations outperforms \((k, \delta)\)-anonymity only for \( \delta \) up to roughly 2000. Not surprisingly, SwapLocations offers better performance than ReachLocations, because the latter must deal with reachability constraints. It is also remarkable that ReachLocations performs much better in terms of SID than in terms of AID. The
Table 2.6: Range query distortion of SwapLocations compared to \((k, \delta)\)-anonymity for SID (columns labeled with S) and AID (columns labeled with A) when using \(k = \{2, 4, 6, 8, 10, 15\}\) and space thresholds that match the space distortion caused by \((k, \delta)\)-anonymity with the previous \(k\)'s and \(\delta = \{0, 1000, 2000, 3000, 4000, 5000\}\). In this table, a range query distortion \(x\) obtained with SwapLocations and a range query distortion \(y\) obtained with \((k, \delta)\)-anonymity are represented as the integer rounding of \((y - x) \times 100\). Hence, values in the table are positive if and only if SwapLocations outperforms \((k, \delta)\)-anonymity.

<table>
<thead>
<tr>
<th>(\delta) (k)</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>34</td>
<td>29</td>
<td>31</td>
<td>14</td>
<td>36</td>
<td>13</td>
</tr>
<tr>
<td>1000</td>
<td>24</td>
<td>20</td>
<td>24</td>
<td>8</td>
<td>28</td>
<td>9</td>
</tr>
<tr>
<td>2000</td>
<td>18</td>
<td>14</td>
<td>18</td>
<td>4</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>3000</td>
<td>8</td>
<td>3</td>
<td>11</td>
<td>-2</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>4000</td>
<td>-6</td>
<td>-7</td>
<td>6</td>
<td>-6</td>
<td>9</td>
<td>-5</td>
</tr>
<tr>
<td>5000</td>
<td>-22</td>
<td>-19</td>
<td>1</td>
<td>-9</td>
<td>3</td>
<td>-7</td>
</tr>
</tbody>
</table>

Table 2.7: Range query distortion of ReachLocations compared to \((k, \delta)\)-anonymity for SID (columns labeled with S) and AID (columns labeled with A) when using \(k = \{2, 4, 6, 8, 10, 15\}\) and space thresholds that match the space distortion caused by \((k, \delta)\)-anonymity with the previous \(k\)'s and \(\delta = \{0, 1000, 2000, 3000, 4000, 5000\}\). In this table, a range query distortion \(x\) obtained with ReachLocations and a range query distortion \(y\) obtained with \((k, \delta)\)-anonymity are represented as the integer rounding of \((y - x) \times 100\). Hence, values in the table are positive if and only if ReachLocations outperforms \((k, \delta)\)-anonymity.

<table>
<thead>
<tr>
<th>(\delta) (k)</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>34</td>
<td>25</td>
<td>28</td>
<td>12</td>
<td>33</td>
<td>10</td>
</tr>
<tr>
<td>1000</td>
<td>25</td>
<td>19</td>
<td>21</td>
<td>6</td>
<td>24</td>
<td>4</td>
</tr>
<tr>
<td>2000</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>-7</td>
<td>17</td>
<td>-3</td>
</tr>
<tr>
<td>3000</td>
<td>-4</td>
<td>2</td>
<td>0</td>
<td>-12</td>
<td>9</td>
<td>-12</td>
</tr>
<tr>
<td>4000</td>
<td>-11</td>
<td>-6</td>
<td>-6</td>
<td>-18</td>
<td>-2</td>
<td>-17</td>
</tr>
<tr>
<td>5000</td>
<td>-14</td>
<td>-5</td>
<td>-10</td>
<td>-22</td>
<td>-8</td>
<td>-25</td>
</tr>
</tbody>
</table>
explanation is that, while \((k, \delta)\)-anonymity and SwapLocations operate at the trajectory level, ReachLocations works at the location level.

We conclude that, according to these experiments, our methods perform better than \((k, \delta)\)-anonymity regarding range query distortion for values of \(\delta\) up to 2000. The performance for larger values of \(\delta\) is less and less relevant: indeed, when \(\delta \to \infty\), \((k, \delta)\)-anonymity means that no trajectory needs to be anonymized and hence the anonymized trajectories are the same as the original ones.

2.7.2 Results on real-life data

The San Francisco cab data set [82] we used consists of several files each of them containing the GPS information of a specific cab during May 2008. Each line within a file contains the space coordinates (latitude and longitude) of the cab at a given time. However, the mobility trace of a cab during an entire month can hardly be considered a single trajectory. We used big time gaps between two consecutive locations in a cab mobility trace to split that trace into several trajectories. All trajectory visualizations shown in this Section were obtained using Google Earth.

For our experiments we considered just one day of the entire month given in the real-life data set, but the empirical methodology described below could be extended to several days. In particular, we chose the day between May 25 at 12:04 hours and May 26 at 12:04 hours because during this 24-hour period there was the highest concentration of locations in the data set. We also defined the maximum time gap in a trajectory as 3 minutes; above 3 minutes, we assumed that the current trajectory ended and that the next location belonged to a different trajectory. This choice was based on the average time gap between consecutive locations in the data set, which was 88 seconds; hence, 3 minutes was roughly twice the average. In this way, we obtained 4582 trajectories and 94 locations per trajectory on average.

The next step was to filter out trajectories with strange features (outliers). These outliers could be detected based on several aspects like velocity, city topology, etc. We focused on velocity and defined 240 km/h as the maximum speed that could be reached by a cab. Consequently, the distance between two consecutive locations could not be greater than 12 km because the maximum within-trajectory time gap was 3 minutes. This allowed us to detect and remove trajectories containing obviously erroneous locations; Figure 2.4 shows one of these removed outliers where a cab appeared to have jumped far into the sea probably due to some error in recording its GPS coordinates. Altogether, we removed 45 outlier trajectories and we were left with a data set of 4547 trajectories with an average of 93 locations per trajectory. Figure 2.5 shows the ten longest trajectories (in number of locations) in the final data set that we used.

Experiments with the distance metric

We propose in this chapter a new distance metric designed specifically for clustering trajectories. Our distance metric considers both space and time, dealing even with non-overlapping or partially-overlapping trajectories. Contrary to the synthetic data where 10 trajectories had to be removed because the distances to them could not be
computed, in this real-life data set our distance function could be computed for every pair of trajectories.

Figure 2.6 shows two trajectories identified by our distance metric as the two closest ones in the data set. The two cabs moved around a parking lot and therefore stayed very close to one another in space. Also in time both trajectories were very close: one of them was recorded between 12:00:49 hours and 13:50:47 hours, while the other was recorded between 12:00:25 hours and 13:52:30 hours. Therefore, both trajectories were correctly identified by our distance metric as being close in time and space; they could be clustered together with minimum utility loss for anonymization purposes.

To compare, Figure 2.7 shows two trajectories identified by the Euclidean distance
as the two closest ones in the data set. These trajectories are located in a parking lot inside San Francisco Airport and, spatially, they are closer than the two trajectories shown in Figure 2.6. However, one of these trajectories was recorded between 24:42:55 hours and 24:55:59 hours, while the other was recorded between 19:05:29 hours and 19:06:15 hours. Hence, they should not be in the same cluster, because an adversary with time knowledge can easily distinguish them.

Figure 2.7: The two closest trajectories in the real-life data set according to the Euclidean distance

Experiments with the SwapLocations method

The ReachLocations method cannot be used when the graph of the city is not provided. Hence, in the experiments with the San Francisco real data we just considered the SwapLocations method. As in the experiments with synthetic data, we set $\Omega = 0$ during the computation of the total space distortion. Figure 2.8 shows the values of total space distortion given by the SwapLocations for different space thresholds and different cluster sizes.

Two other utility properties we are considering in this work are: percentage of removed trajectories and percentage of removed locations. Table 2.8 shows the values obtained with the SwapLocations method for both utility properties.

Finally, Table 2.9 reports the performance of SwapLocations regarding spatio-temporal range queries. We picked random time intervals of length at most 20 minutes. Also, random uncertain trajectories with uncertainty threshold of size at most 7 km were chosen as the regions. Analogously to the experiments with the synthetic data set, 20 and 7 are roughly a quarter of the average duration and distance of all trajectories, respectively. It can be seen that the SwapLocations method provides low range query distortion for every value of $k$ when the space threshold is small, i.e. when the total space distortion is also small. However, the smaller the space threshold, the larger
the number of removed trajectories and locations (see Table 2.8). This illustrates the trade-off between the utility properties considered.

2.8 Conclusions and future work

We have presented two permutation-based heuristic methods to anonymize trajectories with the common features that: i) places and times in the anonymized trajectories are true original places and times with full accuracy; ii) both methods can deal with trajectories with partial or no time overlap, thanks to a new distance also introduced in this chapter. The first method aims at trajectory $k$-anonymity while the second method takes reachability constraints into account, that is, it assumes a territory constrained by a network of streets or roads; to avoid removing too many locations, the second method changes its privacy ambitions from trajectory $k$-anonymity to location $k$-diversity.

Both methods use permutation of locations, and the first method uses also trajectory microaggregation. There are few counterparts in the literature comparable to the first method, and virtually none comparable to the second method. Experimental results show that, for most parameter choices and for similar privacy levels, our methods offer better utility than $(k, \delta)$-anonymity.

Future work will be directed towards designing trajectory anonymization methods aimed at achieving trajectory $p$-privacy (see Definition 6), but discarding less locations than the SwapLocations method. Also, finding trajectory anonymization methods for constrained territories with better utility than ReachLocations is an open challenge.
Table 2.8: Percentage of trajectories (columns labeled with $T$) and locations (columns labeled with $L$) removed by SwapLocations for several values of $k$ and several space thresholds $R^*$ on the real-life data set. Percentages have been rounded to integers for compactness.

<table>
<thead>
<tr>
<th>$R^*$ \ $k$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>L</td>
<td>T</td>
<td>L</td>
<td>T</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>43</td>
<td>40</td>
<td>64</td>
<td>49</td>
<td>71</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>29</td>
<td>34</td>
<td>47</td>
<td>54</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>17</td>
<td>27</td>
<td>35</td>
<td>35</td>
<td>40</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>19</td>
<td>19</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>17</td>
<td>22</td>
<td>20</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>7</td>
<td>2</td>
<td>15</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>64</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>128</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 2.9: Range query distortion caused by SwapLocations on the real-life data set for SID (columns labeled with $S$) and AID (columns labeled with $A$), for several values of $k$ and several space thresholds $R^*$. In this table, a range query distortion $x$ is represented as the integer rounding of $x \times 100$ for compactness.

<table>
<thead>
<tr>
<th>$R^*$ \ $k$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>A</td>
<td>S</td>
<td>A</td>
<td>S</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>22</td>
<td>18</td>
<td>27</td>
<td>20</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>24</td>
<td>25</td>
<td>34</td>
<td>26</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>25</td>
<td>30</td>
<td>37</td>
<td>33</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>27</td>
<td>34</td>
<td>40</td>
<td>38</td>
<td>34</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>26</td>
<td>36</td>
<td>42</td>
<td>42</td>
<td>47</td>
</tr>
<tr>
<td>32</td>
<td>21</td>
<td>26</td>
<td>39</td>
<td>44</td>
<td>45</td>
<td>49</td>
</tr>
<tr>
<td>64</td>
<td>20</td>
<td>25</td>
<td>39</td>
<td>44</td>
<td>46</td>
<td>50</td>
</tr>
<tr>
<td>128</td>
<td>21</td>
<td>26</td>
<td>39</td>
<td>44</td>
<td>48</td>
<td>50</td>
</tr>
</tbody>
</table>
Algorithm 4 ReachLocations($\mathcal{T}, G, R^t, R^s, k$)

Input: i) $\mathcal{T} = \{T_1, \ldots, T_N\}$ a set of original trajectories, ii) $G$ a graph describing the paths between locations, iii) $R^t$ is a time threshold and $R^s$ is a space threshold;

1: Let $TL = \{\lambda^i_j : T_i \in \mathcal{T}\}$ contain all locations from all trajectories, where $\lambda^i_j = (t^i_j, x^i_j, y^i_j)$ and the spatial coordinates $(x^i_j, y^i_j)$ are called a point;

2: Mark all locations in $TL$ as “unswapped”;

3: Let $\mathcal{T}^\star = \emptyset$ be an empty set of anonymized trajectories;

4: while there exist trajectories in $\mathcal{T}$ do

5: Let $T_i$ be a trajectory randomly chosen in $\mathcal{T}$;

6: for $j = 1$ to $j = |T_i|$ do

7: if $\lambda^i_j$ is “unswapped” then

8: Let $C^i_j = \{\lambda^1_j, \ldots, \lambda^{k-1}_j\}$ be a cluster of locations in $TL$ such that:

1. All locations in $C^i_j$ are “unswapped”, with points different from $(x^i_j, y^i_j)$ and no two equal points;

2. Points in $C^i_j$ belong to trajectories in $\mathcal{T} \setminus \{T_i\}$ and no two points belong to the same trajectory;

3. For any $\lambda \in C^i_j$, it holds that:

   (a) $|t^\lambda - t^i_j| \leq R^t$;

   (b) If $j > 1$ there is a path in $G$ between $(x^j_{j-1}, y^j_{j-1})$ and $(x^\lambda, y^\lambda)$;

   (c) If $j < |T_i|$ there is a path in $G$ between $(x^\lambda, y^\lambda)$ and $(x^j_{j+1}, y^j_{j+1})$;

   (d) The length of each path above is no more than $R^s$;

4. The sum of intra-cluster distances (following paths in $G$) in $C^i_j \cup \{\lambda^i_j\}$ is minimum among clusters of cardinality $k - 1$ meeting the previous conditions;

9: if such a cluster $C^i_j$ does not exist then

10: Remove $\lambda^i_j$ from $T_i$;

11: else

12: Mark $\lambda^i_j$ as “swapped”;

13: With probability $\frac{k-1}{k}$:

1. Pick a random location $\lambda \in C^i_j$ and mark it as “swapped”;

2. Swap the spatial coordinates $(x^i_j, y^i_j)$ of $\lambda^i_j$ with the spatial coordinates $(x^\lambda, y^\lambda)$ of $\lambda$;

14: end if

15: end if

16: end for

17: $\mathcal{T}^\star = \mathcal{T}^\star \cup \{T_i\}$;

18: Remove $T_i$ from $\mathcal{T}$;

19: end while

20: return $\mathcal{T}^\star$. 
Chapter 3

Anonymization methods for taxonomic microdata

3.1 Introduction

Statistical disclosure control (SDC, [25, 50, 51, 103, 30, 59]), a.k.a. data anonymization, or privacy-preserving data mining/publishing, aims at making possible the publication of statistical data in such a way that the individual responses of specific users cannot be inferred from the published data and background knowledge available to intruders. The anonymized data should still preserve enough analytical validity for their publication to be useful to potential users.

A microdata set can be defined as a file with a number of records, where each record contains a number of attributes on an individual, e.g. a patient. As an illustration consider the Patient Discharge Data Set 2010 that can be obtained as a public file on CD from California’s Office of Statewide Health Planning and Development [76]. Among other attributes, this data set contains the six following ones: AGE_YRS (age in years), SEX (male and female), RACE (six categories), LOS (length of stay from admission to discharge in days), CHARGE (in dollars) and DIAG_P (principal diagnosis).

Attributes can be classified depending on their range and the operations that can be performed on them:

1. **Numerical.** These are attributes on which arithmetical operations can be performed. Among the six attributes listed above, AGE_YRS, LOS and CHARGE are numerical.

2. **Categorical.** These are attributes taking values over a finite set and standard arithmetical operations on them do not make sense. There are two main types of categorical attributes:

   (a) **Ordinal.** These attributes take values in an ordered range of categories; e.g. “Wound classification” (slight, serious, very serious, critical).
(b) **Nominal.** These take values in an unordered range of categories. The only possible operator is comparison for equality. Nominal attributes can further be divided into two types:

i. **Hierarchical.** A hierarchical nominal attribute takes values in a hierarchical taxonomy, ontology or classification. *E.g.*, among the six attributes listed above, the principal diagnosis DIAG_P is coded according to the International Classification of Diseases (ICD-9-CM, [53]). The economic activity of a company according to the NACE classification ([69]) would be another example.

ii. **Non-hierarchical.** A non-hierarchical nominal attribute takes values in a flat hierarchy. *E.g.*, the SEX and RACE attributes are flat non-hierarchical.

Using existing data anonymization techniques with hierarchical nominal attributes while preserving the original categories (no loss of detail) is not easy. Consider the above mentioned Patient Discharge Data Set 2010. Even if we restrict to the subset of records for which DIAG_P is some form of neoplasm, the DIAG_P attribute in that subset alone takes as many as 542 different categories. For most anonymization techniques not generalizing original categories, it would be necessary to represent each possible category using a binary 0-1 attribute; in the proposed subset of neoplasm records, this would imply adding 541 binary attributes just to represent the DIAG_P attribute. For other anonymization techniques, such as PRAM [50], no binary attributes would be needed, but in the worst-case a Markov transition matrix of size $542 \times 542$ would have to be selected as a parameter to control the anonymization of DIAG_P (if the degrees of freedom of PRAM are reduced, many transition probabilities will be 0 and the matrix parameter will become certainly more manageable). Anyway, in either case, nominal hierarchical attributes greatly increase the computational complexity of anonymization.

Consider for example multiple imputation [85], a commonly used method for data anonymization that provides high data utility and low disclosure risk for the anonymized data [84]. Multiple imputation can be implemented using the sequential regression imputation method (SRMI), described in [83] and available from those authors in the SYNTHESIZE module of the free IVEware package [54]. In order to deal with nominal attributes, SRMI creates a binary 0-1 regressor attribute $B_{ij}$ for each nominal category $j$ of each nominal attribute $i$, so that $B_{ij} = 1$ for a record if and only if attribute $i$ takes category $j$ for that record.

When we attempt to use SRMI on the NEOPLASM data, we are confronted to a regression model where DIAG_P is replaced with 541 binary regressors. This model is far too large for IVEware to compute with reasonable resources. The result is that SRMI is a poor choice to anonymize the DIAG_P attribute. As mentioned earlier, this is true not only for multiple imputation, but for all anonymization techniques that rely on converting the hierarchical nominal data into binary 0-1 attributes.

Even if it was computationally feasible to create binary attributes, these would not capture the category hierarchy of the hierarchical nominal attribute. This is a serious shortcoming for anonymizing data sets which contain important hierarchical nominal attributes (like biomedical or economic data sets).
Other methods, like generalization or suppression, are indeed applicable to hierarchical data, but for some applications they cause an undesirable loss of information and/or detail (e.g., generalizing the values of the DIAG_P attribute would greatly diminish the analytical value of NEOPLASM).

3.1.1 Contribution and plan of this chapter

We show how anonymization methods originally designed for numerical microdata can be adapted for use with nominal and hierarchical nominal attributes thanks to a hierarchy-aware numerical mapping. In this way, the above-mentioned problems of computational complexity and loss of detail are avoided. Specifically, we use the mapping to adapt the data shuffling and microaggregation methods.

Section 3.2 gives background on data shuffling and microaggregation. Section 3.3 introduces the hierarchy-aware numerical mapping and shows how to combine it with data shuffling and microaggregation. Empirical results are presented and discussed in Section 3.4. Section 4.6 summarizes conclusions and lines for future research.

3.2 Background

We next review the two anonymization methods that will be used as building blocks of the adapted methods we propose.

3.2.1 Data shuffling

Data shuffling is a data masking procedure for numerical data introduced in [68]. It combines the best of perturbation and swapping. Data perturbation modifies (perturbs) the original values, which is a disadvantage, but it has the strength of providing high utility and low disclosure risk (if the perturbation is well-tuned). Data swapping by contrast does not modify the original values, but it swaps (exchanges) them between records. However, swapping cannot assure a disclosure risk as low as perturbation. By combining perturbation and swapping, data shuffling offers a versatile hybrid method that provides very high data utility and low disclosure risk.

Data shuffling can be described as follows. Let $X$ represent the set of confidential attributes and let $S$ represent the set of non-confidential attributes. Data shuffling models the joint distribution of $\{X, S\}$ as a multivariate normal (Gaussian) copula. In the copula approach, the joint distribution of $\{X, S\}$ is characterized by the rank order correlation matrix $\mathbf{R}$. The normalized values of the data set $\{X^*, S^*\}$ are characterized by the product moment correlation $\rho$, where the relationship between $\mathbf{R} = (r_{ij})$ and $\rho = (\rho_{ij})$ is $\rho_{ij} = 2\pi \sin(r_{ij})/6$ with $r_{ij}$ and $\rho_{ij}$ being, respectively, the rank order and product moment correlation between attributes $i$ and $j$. The normalized values $\{X^*, S^*\}$ are described by a multivariate normal distribution characterized by $\rho$. Using this information, the perturbed normalized values $Y^*$ are created using the conditional distribution $f(X^*|S^*)$. The generated values $Y^*$ are independent of $X^*$ (and $X$). Once the values of $Y^*$ have been generated in this manner, the original values
of $X$ are reverse mapped to $Y^*$ to obtain the perturbed values $Y$. For more details, see [68].

Data shuffling offers the following advantages:

1. The shuffled values $Y$ have the same marginal distribution as the original values $X$. Hence, the results of all univariate analyses using $Y$ provide exactly the same results as analyses using $X$.

2. The rank order correlation matrix of $\{Y, S\}$ is asymptotically the same as the rank order correlation matrix of $\{X, S\}$. In other words, for large data sets, data shuffling preserves linear and monotonic non-linear relationships. Hence, results of all analyses that involve linear relationship between attributes (such as regression analyses) or non-linear relationships using the shuffled data $\{Y, S\}$ will be very similar to results using the original data $\{X, S\}$. Furthermore, as the size of the data set increases, the difference in the results will also get smaller.

3. The shuffled values $Y$ are generated independently of the original values $X$, based only on the values of the non-confidential attributes $S$. In the absence of non-confidential attributes (that is, when $S$ is null), $X$ and $Y$ are statistically independent, that is, $Y$ is synthetic. Theoretically, this provides the lowest possible level of disclosure risk. Thus, data shuffling minimizes the risk of disclosure.

### 3.2.2 Microaggregation

Microaggregation is a family of perturbative SDC methods originally defined for numerical data [18, 23]. Microaggregation can be defined in terms of two steps:

**Partition:** The set of original records is partitioned into small groups in such a way that records in the same group are *similar* to each other and so that the number of records in each group is at least $k$.

**Aggregation:** An aggregation operator (typically the group centroid/mean) is computed for each group and is used to replace the original records. In other words, each record in a group is replaced by the group prototype.

Computational improvements of microaggregation can be found in [74, 20, 24, 27]. In [98, 24], extensions of microaggregation for categorical attributes were proposed: the former paper addressed only categorical ordinal attributes and proposed the median as an aggregation operator; the latter paper also considered nominal attributes (with no hierarchy) and proposed the modal value as an aggregation operator for them. Clearly, the modal value is a very coarse aggregation operator which may not even be uniquely defined, especially over a small group of values. Also, the two extensions mentioned fail to capture the semantics of hierarchical nominal data.

In [19], microaggregation-based hybrid generation for numerical data was proposed. The idea is to replace the aggregation step of microaggregation by a synthetic data generator preserving means and covariances, in such a way that the resulting anonymized data set exactly preserves the means and covariances of the original data set and approximately preserves them in random subdomains.
3.3 Our method

In this section, we first describe a hierarchy-aware numerical mapping of hierarchical nominal attributes. Then we show how this approach can be used effectively for anonymization techniques designed for numerical data. Consider a nominal attribute $X$ taking values from a hierarchical classification. Let $T_X$ be a sample of values of $X$. We propose the following algorithm to compute a new measure of the marginality (non-centrality) of the values in the sample $T_X$.

**Algorithm 1 (Marginality of nominal values).**

1. Given a sample $T_X$ of nominal categorical values drawn from $X$, place them in the tree representing the hierarchy of $X$. There is a one-to-one mapping between the set of tree nodes and the set of categories where $X$ takes values. Prune the sub-trees whose nodes have no associated sample values. If there are repeated sample values, there will be several nominal values associated to one or more nodes (categories) in the pruned tree.

2. Let $L$ be the depth of the pruned tree. Associate weight $2^{L-1}$ to edges linking the root of the hierarchy to its immediate descendants (depth 1), weight $2^{L-2}$ to edges linking the depth 1 descendants to their own descendants (depth 2), and so on, up to weight $2^0 = 1$ to the edges linking descendants at depth $L-1$ with those at depth $L$. In general, weight $2^{L-i}$ is assigned to edges linking nodes at depth $i - 1$ with those at depth $i$, for $i = 1$ to $L$.

3. For each nominal value $x_j$ in the sample, its marginality $m(x_j)$ is defined and computed as

$$m(x_j) = \sum_{x_l \in T_X - \{x_j\}} d(x_j, x_l)$$

where $d(x_j, x_l)$ is the sum of the edge weights along the path from the tree node corresponding to $x_j$ and the tree node corresponding to $x_l$.

Clearly, the greater $m(x_j)$, the more marginal (i.e. the less central) is $x_j$. Marginality constitutes a hierarchy-aware numerical mapping for nominal attributes. Note that marginality also takes sample frequencies into account: if the frequency of a value in $T_X$ increases, the marginality of that value decreases.

In addition to representing hierarchical nominal data using a numerical mapping, we need to describe the statistical characteristics of the resulting numerical attribute. Next, we provide derivations for these statistical characteristics (see [26] for proofs of correctness).

**Definition 14 (Marginality-based variance).** Given a sample $T_X$ of $n$ values drawn from a hierarchical nominal attribute $X$, the marginality-based sample variance is defined as

$$Var_M(T_X) = \frac{\sum_{x_j \in T_X} m(x_j)^2}{n}$$

47
In [26] it is shown that the above marginality-based variance is equivalent to the hierarchical variance defined in [28].

**Definition 15 (Marginality-based approximated mean).** Given a sample $T_X$ of a hierarchical nominal attribute $X$, the marginality-based approximated mean is defined as

$$\text{Mean}_M(T_X) = \arg \min_{x_j \in T_X} m(x_j)$$

if one wants the mean to be a nominal value, or

$$\text{Num\_mean}_M(T_X) = \min_{x_j \in T_X} m(x_j)$$

if one wants a numerical mean value.

**Definition 16 (S-distance).** The S-distance between two records $x_1$ and $x_2$ in a data set with $d$ attributes is

$$\delta(x_1, x_2) = \sqrt{\frac{(S^2)_1}{(S^2)_1} + \cdots + \frac{(S^2)_d}{(S^2)_d}}$$

where $(S^2)_l$ is the variance of the $l$-th attribute over the group formed by $x_1$ and $x_2$, and $(S^2)_d$ is the variance of the $l$-th attribute over the entire data set.

In [26] the above is shown to be a mathematical distance.

**Definition 17 (Marginality-based covariance).** Given a bivariate sample $T_{(X,Y)}$ consisting of $n$ ordered pairs of values $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from the ordered pair of nominal attributes $(X, Y)$, the marginality-based sample covariance is defined as

$$\text{Covar}_M(T_{(X,Y)}) = \frac{\sum_{j=1}^{n} \sqrt{m(x_j)m(y_j)}}{n}$$

The value of the above non-negative expression is higher when the marginalities of the values taken by $X$ and $Y$ are positively correlated.

The key feature of marginality is that it allows converting the hierarchical nominal attribute into a single numerical attribute which preserves the hierarchical structure. We have provided definitions of statistical measures that allow us to compare the original and anonymized data. Once converted in this manner, it is easy to apply selected numerical anonymization techniques without any additional computational complexity.

Note that not all numerical anonymization techniques can be applied after the marginality conversion. As an example, if multiple imputation or noise addition are used, the anonymized converted nominal attributes take values in a different set as the converted original nominal attributes. When this occurs, it becomes impossible to map the anonymized numerical values back to the original nominal categories. Hence, the anonymization procedure must return values for the attributes that belong to the marginality values to which original categories were mapped. We show below that shuffling fulfills this requirement and that microaggregation can be adapted to fulfill it.
3.3.1 Shuffling with marginality

For nominal data, data shuffling is typically implemented by converting the nominal attribute to binary 0-1 attributes. As discussed earlier, this results in much higher computational requirements. However, if the marginality mapping is performed, data shuffling can be implemented without any additional computational requirements. More importantly, marginality captures the hierarchical nature of the data set directly. Once represented in this manner, data shuffling has the capability to maintain the same structure in the masked data as was present in the original data. Converting marginalities back to nominal categories in the shuffled data set is feasible because shuffling does not modify the original nominal categories; hence, provided that the mapping from nominal categories to marginalities is one-to-one in the original data set (which happens with great probability), the mapping is the same in the shuffled data and can be inverted.

3.3.2 Hybrid microaggregation with marginality

In microaggregation of numerical data, the partition step seeks to form groups with high within-group similarity, that is, with low within-group variance. In order to microaggregate hierarchical nominal data, we need some measure of variance for such data. The intuitive idea behind variance in a hierarchy is that a sample of nominal values belonging to categories which are all children of the same parent category in the hierarchy has smaller variance than a sample with children from different parent categories. The average marginality of a sample turns out to capture this notion of variance (Definition 14 above).

Using the definitions of variance, mean and distance given earlier (Definitions 14, 15 and 16), we can easily adapt to nominal and hierarchical nominal attributes any microaggregation heuristic originally designed for numerical attributes (e.g. MDAV, [24]). To avoid the variance reduction caused by the aggregation step of microaggregation, in this chapter we will use only the partition step. Instead of replacing the records in a group by their centroid, we will replace them by a group of records generated with the following algorithm. The result are microaggregation-based hybrid data and we call the method marginality-based MicroHybrid. We proposed a parallel idea to generate hybrid numerical data in [19].

Algorithm 2.
For each record \( x_j \) in \( C_i \) and for each attribute \( X_l \) to be synthesized:

1. If \( X_l \) is hierarchical nominal, ordinal or numerical

   Pick a random value \( x'_{jl} \) among those that can be taken by \( X_l \) such that
   \[
   \delta(x_{jl}, x'_{jl}) \leq \delta_{\text{max},i}(x_{jl})
   \]

   where \( \delta(\cdot, \cdot) \) is a suitable distance (S-distance if \( X_l \) is hierarchical nominal, ordinal if \( X_l \) is ordinal and Euclidean if \( X_l \) is numerical), \( x_{jl} \) is the original value of the attribute in \( x_j \) and \( \delta_{\text{max},i}(x_{jl}) \) is the maximum distance between \( x_{jl} \) and the values taken by attribute \( X_l \) over group \( C_i \).
else \((X_i \text{ is flat nominal})\) randomly draw a value \(x'_{jl}\) with replacement from the set of values of \(X_i\) over \(C_i\);

2. Replace \(x_{jl}\) by \(x'_{jl}\).

3.4 Experimental results

From the entire data set, we selected the subset of records for which DIAG_P was some form of neoplasm. Among these we deleted the records with missing data and those for which CHARGE was $0 (a value 0 means that the charge for that discharge was unknown or invalid). This left us with 19502 records with the six attributes listed in Section 3.1 above. We will refer to this data subset as the NEOPLASM data.

We applied the marginality-based shuffling and the marginality-based \textit{MicroHybrid} to the NEOPLASM data. As observed earlier, SRMI is incapable of anonymizing the DIAG_P attribute in the NEOPLASM data. However, multiple imputation is considered to be one of the techniques offering a very high level of data utility and a low level of disclosure risk. In this sense, it represents a good benchmark to assess the effectiveness of the other two techniques. Hence, in addition to the results of using adapted shuffling and adapted \textit{MicroHybrid}, we also report results of using SRMI on the NEOPLASM data (without DIAG_P) as a benchmark.

One key aspect of the \textit{MicroHybrid} technique is the selection of the aggregation parameter \(k\). In order to conduct a fair comparison, we empirically determined a value of \(k\) for which the disclosure risk in \textit{MicroHybrid} was comparable to the disclosure risk of the SRMI-generated data and the shuffled data (note that SRMI and shuffling have no parameters, so we could only adjust \(k\)). Disclosure risk was measured by using distance-based record linkage. An anonymized record was linked to the original record whose values of the original attributes were at a shortest distance (using S-distance for nominal attributes and Euclidean distance for numerical attributes). The percentage of correct matches was taken as a measure of disclosure risk. We obtained a disclosure risk of 0.015382\% for SRMI (with five attributes) and of 0.005127\% for shuffling. We took \(k = 600\) for \textit{MicroHybrid}, which yielded a disclosure risk of 0.03589\%.

In order to assess the utility of the anonymized data sets obtained with the three methods, for several statistics \(\Theta\), we computed the variation \(\Delta(\Theta) = (\theta' - \theta)/\theta\) between the value \(\theta\) of the statistic over the original data set and its value \(\theta'\) over the anonymized data set. Specifically, we considered the statistics mean, variance and covariance. For nominal attributes, we used the definitions of mean, variance and covariance given in Section 3.3. While we were able to define covariance between nominal attributes, we do not have a definition of covariance between a nominal and a numerical attribute. Hence, this information is not reported. Table 3.1 shows the results of using the three methods on the NEOPLASM data set.

From the results in Table 3.1, we can see that marginality-based shuffling performs even better than the multiple imputation SRMI in most cases. Only for the covariances between \((\text{AGE}_\text{YRS}, \text{LOS})\) and between \((\text{AGE}_\text{YRS}, \text{CHARGE})\) does multiple imputation outperform data shuffling. And, as observed earlier, data shuffling also results in lower disclosure risk than multiple imputation. In summary, for the NEOPLASM data
Table 3.1: Variation $\Delta(\Theta)$ for means, variances and covariances between the original NEOPLASM data set and the data set anonymized using multiple imputation, marginality-based data shuffling and marginality-based MicroHybrid with $k = 600$.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Attribute</th>
<th>Multiple imputation</th>
<th>Marg.-based shuffling</th>
<th>Marg.-based MicroHybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>AGE_YRS</td>
<td>0.00042</td>
<td>0</td>
<td>0.03627</td>
</tr>
<tr>
<td></td>
<td>SEX</td>
<td>0.00373</td>
<td>0</td>
<td>0.01349</td>
</tr>
<tr>
<td></td>
<td>RACE</td>
<td>0.00033</td>
<td>0</td>
<td>0.01532</td>
</tr>
<tr>
<td></td>
<td>LOS</td>
<td>0.26345</td>
<td>0</td>
<td>0.57532</td>
</tr>
<tr>
<td></td>
<td>CHARGE</td>
<td>0.44923</td>
<td>0</td>
<td>0.91774</td>
</tr>
<tr>
<td></td>
<td>DIAG_P</td>
<td>N/A</td>
<td>0</td>
<td>0.00210</td>
</tr>
<tr>
<td>Variance</td>
<td>AGE_YRS</td>
<td>0.00785</td>
<td>0</td>
<td>0.02612</td>
</tr>
<tr>
<td></td>
<td>SEX</td>
<td>0.00124</td>
<td>0</td>
<td>0.00465</td>
</tr>
<tr>
<td></td>
<td>RACE</td>
<td>0.00011</td>
<td>0</td>
<td>0.01362</td>
</tr>
<tr>
<td></td>
<td>LOS</td>
<td>0.24603</td>
<td>0</td>
<td>7.54256</td>
</tr>
<tr>
<td></td>
<td>CHARGE</td>
<td>0.37275</td>
<td>0</td>
<td>19.81236</td>
</tr>
<tr>
<td></td>
<td>DIAG_P</td>
<td>N/A</td>
<td>0</td>
<td>0.00197</td>
</tr>
<tr>
<td>Covariance</td>
<td>(AGE_YRS, LOS)</td>
<td>5.56559</td>
<td>33.69468</td>
<td>162.38189</td>
</tr>
<tr>
<td></td>
<td>(AGE_YRS, CHARGE)</td>
<td>0.12300</td>
<td>1.60121</td>
<td>11.15957</td>
</tr>
<tr>
<td></td>
<td>(SEX, RACE)</td>
<td>0.00064</td>
<td>0.00059</td>
<td>0.00514</td>
</tr>
<tr>
<td></td>
<td>(SEX, DIAG_P)</td>
<td>N/A</td>
<td>0.00006</td>
<td>0.00205</td>
</tr>
<tr>
<td></td>
<td>(RACE, DIAG_P)</td>
<td>N/A</td>
<td>0.00005</td>
<td>0.00914</td>
</tr>
<tr>
<td></td>
<td>(LOS, CHARGE)</td>
<td>0.34455</td>
<td>0.11783</td>
<td>11.58553</td>
</tr>
</tbody>
</table>


set, data shuffling offers both high utility and low disclosure risk.

The performance of MicroHybrid is somewhat worse than both data shuffling and multiple imputation. However, MicroHybrid at least offers the ability to anonymize the data, while multiple imputation does not. Hence, in the presence of hierarchical nominal data, MicroHybrid would be preferred over multiple imputation.

In addition to evaluating the utility over the entire data set, we evaluated the utility loss when the user restricts her analysis to a random subdomain of the data set. Thus, for each method, we took 10% samples of the anonymized data set 100 times. We then computed the mean variation \( \Delta(\Theta) \) over the 100 samples and the corresponding samples of the original data set for the same statistics as above. The results are shown in Table 3.2. None of the three methods wins clearly for all statistics: shuffling no longer exactly preserves means and variances for subdomains and, for some statistics, it is slightly outperformed by one of the other two methods. However, SRMI is the clear loser to its inability to deal with DIAG_P and shuffling seems to behave somewhat better than MicroHybrid (the latter has quite large mean variations for four statistics).

In fact, the smaller \( k \) w.r.t. the sample size, the better MicroHybrid preserves statistics on random samples, because the generation of hybrid data is more constrained to being similar to the original data (in [19] we showed this for numerical data). However, we were forced to take \( k = 600 \) in order to match the low disclosure risk of data shuffling and multiple imputation, and \( k = 600 \) is not much smaller than the size of a 10% sample (1950 records).

### 3.5 Conclusions and future research

Hierarchical attributes are common in many data sets (for example biomedical or economic data), and they are often among the most important attributes (for example, “Diagnostic” or “Economic activity”). When used for secondary purposes, data containing hierarchical attributes must be anonymized, but, unfortunately, most existing anonymization techniques cannot be used. In this study, we have addressed this issue.

We have described a hierarchy-aware numerical mapping for hierarchical attributes, called marginality. With this mapping, any numerical anonymization procedure not perturbing original values (i.e., keeping the mapping reversible) can be employed to anonymize the data without any additional computational complexity. We have illustrated the application of this approach using data shuffling and microaggregation. The performance of these techniques was evaluated using the NEOPLASM data. Our results indicate that both techniques perform well.

Future research lines include:

- Extending the marginality-based mapping for anonymization techniques perturbing the input marginalities. One could think of an approximate reverse mapping for methods which do perturb input marginalities; that is, each numerical output marginality \( m \) could be mapped back to the hierarchical category having marginality closest to \( m \). However, approximate reverse mapping can lead to gross distortion if there are categories very distant within the hierarchy that have similar marginalities, because they could be unduly swapped. Hence, blocking
Table 3.2: Mean variation for means, variances and covariances between one hundred 10% original and anonymized samples of the NEOPLASM data set. Anonymized samples were created using multiple imputation, marginality-based data shuffling and marginality-based MicroHybrid with $k = 600$.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Attribute</th>
<th>Multiple imputation</th>
<th>Marg.-based shuffling</th>
<th>Marg.-based MicroHybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AGE_YRS</td>
<td>0.00654</td>
<td>0.00757</td>
<td>0.03644</td>
</tr>
<tr>
<td></td>
<td>SEX</td>
<td>0.03060</td>
<td>0.02450</td>
<td>0.03044</td>
</tr>
<tr>
<td></td>
<td>RACE</td>
<td>0.05581</td>
<td>0.05649</td>
<td>0.05370</td>
</tr>
<tr>
<td></td>
<td>LOS</td>
<td>0.26409</td>
<td>0.03139</td>
<td>0.57156</td>
</tr>
<tr>
<td></td>
<td>CHARGE</td>
<td>0.45352</td>
<td>0.03406</td>
<td>0.89759</td>
</tr>
<tr>
<td></td>
<td>DIAG_P</td>
<td>N/A</td>
<td>0.00573</td>
<td>0.00745</td>
</tr>
<tr>
<td><strong>Variance</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AGE_YRS</td>
<td>0.04109</td>
<td>0.03975</td>
<td>0.02969</td>
</tr>
<tr>
<td></td>
<td>SEX</td>
<td>0.01028</td>
<td>0.00822</td>
<td>0.01047</td>
</tr>
<tr>
<td></td>
<td>RACE</td>
<td>0.04925</td>
<td>0.04996</td>
<td>0.04747</td>
</tr>
<tr>
<td></td>
<td>LOS</td>
<td>0.26744</td>
<td>0.29376</td>
<td>7.86135</td>
</tr>
<tr>
<td></td>
<td>CHARGE</td>
<td>0.32469</td>
<td>0.40743</td>
<td>21.53582</td>
</tr>
<tr>
<td></td>
<td>DIAG_P</td>
<td>N/A</td>
<td>0</td>
<td>0.00296</td>
</tr>
<tr>
<td><strong>Covariance</strong></td>
<td>(AGE_YRS, LOS)</td>
<td>6.11472</td>
<td>11.01168</td>
<td>159.34892</td>
</tr>
<tr>
<td></td>
<td>(AGE_YRS, CHARGE)</td>
<td>1.50547</td>
<td>1.94719</td>
<td>20.94929</td>
</tr>
<tr>
<td></td>
<td>(SEX, RACE)</td>
<td>0.02942</td>
<td>0.02984</td>
<td>0.02772</td>
</tr>
<tr>
<td></td>
<td>(SEX, DIAG_P)</td>
<td>N/A</td>
<td>0.00546</td>
<td>0.00674</td>
</tr>
<tr>
<td></td>
<td>(RACE, DIAG_P)</td>
<td>N/A</td>
<td>0.02924</td>
<td>0.02821</td>
</tr>
<tr>
<td></td>
<td>(LOS, CHARGE)</td>
<td>0.35682</td>
<td>0.28189</td>
<td>12.32575</td>
</tr>
</tbody>
</table>
strategies or other mechanisms should be devised to avoid such undesirable effects.

- Using semantic metrics developed in artificial intelligence (e.g. [88]) as an alternative to assess the semantic distortion incurred by marginality-based anonymization.
Chapter 4

Probabilistic $k$-anonymity through microaggregation and data swapping

4.1 Introduction

A microdata file is composed of records that contain information specific to individuals (who may be citizens, companies, etc.) in the data set. These records contain, for each specific individual, the values corresponding to a list of attributes. Microdata files are the result of data collection processes carried out by national statistical offices, healthcare systems, electronic commerce, etc. They are a valuable resource for analysts and researchers, but also a threat to the individuals’ privacy. Direct publication of microdata files results in an unacceptable privacy breach for the individuals therein contained. Therefore, before being released, microdata files must undergo a process of anonymization that dissociates the identity of the individuals from specific records.

The two main aspects that any anonymization method must address are disclosure risk and information loss. There is an extensive literature about methods used to provide anonymity for microdata releases. Some good surveys on microdata anonymization are [25, 39, 17].

Among the several approaches to disclosure risk limitation for microdata files, we focus on $k$-anonymity. This is not really an anonymization method, but a privacy property that the published data set must satisfy. If $k$-anonymity is judged to be a sufficient guarantee for the privacy of the individuals in the data set, then the focus goes to the selection of a method that produces a data set satisfying $k$-anonymity with minimal information loss.

In a $k$-anonymous microdata set, for each combination of values of the quasi-identifier attributes present in the data set, there must be at least $k$ records sharing that combination. In other words, a record must be indistinguishable within a set of $k$ records as far as their quasi-identifier attributes are concerned. To fulfill this require-
ment, the data granularity of the quasi-identifiers is reduced, usually by generalization, suppression or micro-aggregation. The strict indistinguishability requirement of \( k \)-anonymity may lead to a substantial amount of information loss, especially if there is a large number of quasi-identifiers [5].

Our goal is to achieve the same level of disclosure risk limitation that \( k \)-anonymity provides, while improving the data quality of the released data set. Our proposal is based on a relaxation of the indistinguishability requirement of \( k \)-anonymity. Instead of requiring records to be indistinguishable within sets of \( k \) records in terms of the quasi-identifiers, we focus on the probability of re-identification. By requiring this probability to be at most \( 1/k \), we achieve the same level of protection against re-identification provided by \( k \)-anonymity, but the range of applicable methods is wider and hence the information loss can be reduced.

4.1.1 Contribution and plan of this paper

We introduce in this paper the concept of probabilistic \( k \)-anonymity, which like \( k \)-anonymity yields a re-identification probability at most \( 1/k \) but with much better data quality preservation. This is especially relevant when dealing with a data set that contains many quasi-identifier attributes.

Section 4.2 introduces some background concepts that are required for later sections. Section 4.3 presents probabilistic \( k \)-anonymity. Section 4.4 describes a computational procedure based on microaggregation and swapping to achieve probabilistic \( k \)-anonymity. Experimental results comparing the data quality loss caused by standard \( k \)-anonymity and probabilistic \( k \)-anonymity are reported in Section 4.5. Conclusions and future research are summarized in Section 4.6.

4.2 Background

A microdata set can be modeled as a table where each row refers to a different individual and each column contains information regarding one of the attributes collected. We use the notation \( T(A_1, \ldots, A_n) \) to denote a microdata set with information about attributes \( A_1, \ldots, A_n \).

The attributes in a microdata set are usually classified in the following non-exclusive categories according to the sensitiveness of the information they convey and the risk of record re-identification they imply:

- **Identifiers.** An attribute is an identifier if it provides unambiguous re-identification of the individual to which the record refers. Some examples of identifier attributes are the social security number, the passport number, etc. If a record contains an identifier, any sensitive information contained in other attributes may immediately be linked to a specific individual. To avoid direct re-identification of an individual, identifier attributes are usually removed or encrypted. We assume in the rest of this paper that the microdata set \( T(A_1, \ldots, A_n) \) does not contain any identifier attribute.
• **Quasi-identifiers.** Unlike an identifier, a quasi-identifier attribute alone does not lead to record re-identification. However, in combination with other quasi-identifier attributes, it may allow unambiguous re-identification of some individuals. For example, [94] shows that 87% of the population in the U.S. can be unambiguously identified by combining a 5-digit ZIP code, birth date and sex. Removing quasi-identifier attributes, as proposed for the identifiers, is not possible, because quasi-identifiers are required to perform any useful analysis on the data. Moreover, any attribute is potentially a quasi-identifier, depending on the external information available to the intruder; hence, to make sure all quasi-identifiers have been removed, one should remove all attributes (!).

• **Confidential attributes.** Confidential attributes hold sensitive information on the individuals that took part in the data collection process (e.g. salary, health condition, sex orientation, etc.). The primary goal of microdata protection techniques is to prevent intruders from learning confidential information about a specific individual. This goal involves not only preventing the intruder from determining the exact value a confidential attribute takes for some individual, but preventing inferences on the value of that attribute (like bounding it).

• **Non-confidential attributes.** Non-confidential attributes are those that do not belong to any of the previous categories. As they do not contain sensitive information about individuals and cannot be used for record re-identification, they do not affect our discussion on disclosure limitation for microdata sets. We assume for the rest of the paper that none of the attributes in $T(A_1, \ldots, A_n)$ belongs to this category.

When publishing a microdata file, the data collector must guarantee that no sensitive information about specific individuals is disclosed. To do so, the data collector does not publish the original microdata set $T(A_1, \ldots, A_n)$, but a modified version $T'(A_1, \ldots, A_n)$ where the quasi-identifiers and/or the confidential attributes have been masked. Disclosure can be classified in two categories [50]:

• **Identity disclosure.** The intruder is able to determine the true identity of the individual to which a record in the microdata file corresponds; the intruder can subsequently associate to this individual the values of the confidential attributes for that record.

• **Attribute disclosure.** Even if identity disclosure does not happen, it may be possible for an intruder to infer some information for a specific individual based on the published microdata set. For example, imagine that the salary is one of the confidential attributes and the job is a quasi-identifier attribute; if an intruder is interested in a specific individual whose job he knows to be “accountant” and there are several accountants in the data set (including the target individual), the intruder will be unable to re-identify the individual’s record based only on her job, but he will be able to lower-bound and upper-bound the individual’s salary (which lies between the minimum and the maximum salary of accountants in the data set).
An intruder who wants to re-identify an individual usually exploits some external information to perform a record linkage attack. A record linkage attack tries to link the records in an external non-anonymous data set back to the records in the published data set, thereby associating an identity to them. Assume that the intruder knows that some individual is in the published microdata set, and also knows a set of quasi-identifier attributes regarding this individual. To perform the record linkage attack, the intruder tries to match the quasi-identifier attributes he knows to some record in the published data set. If the intruder performs the right linkage to the published data set, the attack succeeds, and the intruder learns the confidential attribute values associated to that individual.

A possible approach towards avoiding identity disclosure is the one taken by \( k \)-anonymity, where each record in the published microdata set is made indistinguishable within a set of \( k \) records based on the quasi-identifiers. This way an intruder with access to an external non-anonymous data set that contains the quasi-identifiers in \( T(A_1, \ldots, A_n) \) is unable to perform a re-identification of the records in the published data set. Given a specific individual in the external data set, the intruder can at most determine a set of \( k \) records in the published data set that must contain that individual. The original proposal to achieve \( k \)-anonymity [87] was based on generalization and suppression of the information contained in the quasi-identifier attributes. Another proposal is based on micro-aggregation ([23, 24]).

\( k \)-Anonymity does not in general protect against attribute disclosure. If all the individuals within a group of indistinguishable records have the same value for a confidential attribute, then an intruder will learn the value of that confidential attribute for those individuals, even without re-identification. Further refinements on \( k \)-anonymity that try to address attribute disclosure have been proposed: \( l \)-diversity [63] requires the presence of \( l \) different values for the confidential attribute in every group of records sharing the same quasi-identifier values; \( t \)-closeness [61] requires the distribution of the confidential attribute in any group of records sharing the same quasi-identifier values to be close to its distribution in the overall data set.

To provide an accurate definition of \( k \)-anonymity, we first formalize and slightly generalize the definition of quasi-identifier. Usually a quasi-identifier is said to be a group of attributes that can be employed to unambiguously identify an individual. We note that any combination of attributes that may provide a level of re-identification beyond the admissible limits set by the data collector should also be treated as a quasi-identifier. For example, if we want to guarantee \( k \)-anonymity, any combination of externally available attributes that may be used to refer to a set containing less than \( k \) records must be considered as a quasi-identifier.

**Definition 18** (Quasi-identifier). A quasi-identifier \( QI \) of \( T \) is a subset of the set of attributes \( \{A_1, \ldots, A_n\} \) that satisfy the following two conditions: (i) the attributes in \( QI \) are available in an external, non-anonymous data set; (ii) the values of the attributes in \( QI \) may allow an intruder to determine the identity corresponding to a record in the published microdata set beyond an admissible level.

The goal of \( k \)-anonymity is to cloak the identity corresponding to a record by making each record indistinguishable within a set of \( k \) records. In other words, given a record in an external non-anonymous data set, an intruder must not be able to link it
with certainty to a set of records in the published data set with cardinality less than $k$. This determines the criterion used by $k$-anonymity to define what is an admissible level of re-identification, and hence what is a $k$-anonymous quasi-identifier.

For $k$-anonymity to provide the desired level of protection, an intruder must not be able to link an external record in a non-anonymous data set to a group of less than $k$ records in the published data set, no matter the quasi-identifier used. Note that, by adding more attributes to a quasi-identifier, we increase the level of certainty that the intruder may get. Therefore, to achieve protection against all possible quasi-identifiers $QI_1, \ldots, QI_m$, it suffices to achieve $k$-anonymity for the quasi-identifier that results from the union $QI_1 \cup \ldots \cup QI_m$.

**Definition 19** ($k$-Anonymity [87]). A microdata set $T'(A_1, \ldots, A_n)$ is said to satisfy $k$-anonymity if, for each record $t \in T'$, there are at least $k - 1$ other records sharing the same values for all the quasi-identifier attributes.

In order to apply the previous definition of $k$-anonymity, the data collector needs to know which attributes are available externally in a non-anonymous data set. Since assuming such knowledge by the data collector is a strong assumption, we will consider only two scenarios:

1. Uninformed intruder scenario. The intruder does not know the value that any confidential attribute takes for any individual in the data set.

2. Informed intruder scenario. An informed intruder may know some of the confidential attributes for some of the individuals. This may happen, for example, if the intruder is acquainted with an individual that is included in the microdata set. If the confidential attributes known by the intruder were not deemed quasi-identifiers, the intruder might exploit his knowledge to obtain a more accurate re-identification.

There may be multiple informed intruders, each of them knowing a different subset of confidential attributes over a different subset of records. For the informed intruder scenario we assume that the number of intruders and confidential attributes is the same, and that each of the intruders knows the values of all confidential attributes for all individuals, except for one confidential attribute whose values are completely unknown to the intruder for all individuals. We also assume that the intruders do not collude, as in that case they would be able to learn all the confidential attributes for all individuals even without seeing the published microdata set. Our intruders are not the strongest possible ones: a stronger intruder would be one with total knowledge of all confidential attributes except one, and partial knowledge of the remaining confidential attribute (whose values would be known to the intruder for some individuals). However, we judge the proposed intruders to be reasonably strong.

### 4.3 Probabilistic $k$-anonymity

$k$-Anonymity guarantees that, for any combination of values of quasi-identifier attributes in the published microdata set $T'(A_1, \ldots, A_n)$, there are at least $k$ records
sharing that combination of values. Therefore, given an individual in an external non-anonymous data set, the probability of performing the right linkage back to the corresponding record in the published microdata set, and thus the probability of learning its confidential attributes, is at most $1/k$. It is in this sense that probabilistic $k$-anonymity is defined.

A similar relaxation on the notion of $k$-anonymity was presented in [107], which partitioned the dataset and applied a permutation inside each of the partition components. We do the same in Section 4.4 to achieve probabilistic $k$-anonymity. However, ours is a more general framework, not limited to permutations (even if permutations are convenient choice to simplify probability calculations). Moreover [107] was limited to a single confidential attribute, whereas we handle multiple confidential attributes that can also be quasi-identifiers.

**Definition 20** (Probabilistic $k$-anonymity). Let $T'(A_1, \ldots, A_n)$ be a published data set generated from an original data set $T(A_1, \ldots, A_n)$ using an anonymization mechanism $M$. The data set $T'$ is said to satisfy probabilistic $k$-anonymity if, for any non-anonymous external data set $E$, the probability for an intruder $I$ knowing $T'$ and $E$ to correctly link any record $x \in E$ and its corresponding record (if any) in $T'$ is at most $1/k$.

Note than any method used to achieve $k$-anonymity also leads to probabilistic $k$-anonymity. In this sense, it may be said that $k$-anonymity provides a stronger guarantee. However, from the point of view of the probability of re-identification, both provide the same level of protection.

The advantage of probabilistic $k$-anonymity in comparison to $k$-anonymity is that, by relaxing the requirements on the indistinguishability within groups of $k$ records, the range of eligible methods to enforce it is wider, and therefore we may expect a reduction in the information loss.

As probabilistic $k$-anonymity is expressed in terms of probability of re-identification, it is natural to think of the released data set $T'(A_1, \ldots, A_n)$ as a perturbation of $T(A_1, \ldots, A_n)$. We use the notations in Figure 4.1. The records $x_i$ in $T$ have been split in two parts: the quasi-identifier attributes $q_i$, and the confidential attributes $c_i$. The records in $T'$ are obtained by applying a random perturbation to the corresponding record in $T$: $x'_i = X(x_i)$. This perturbation affects only the quasi-identifier attributes. For the sake of simplicity, we assume that the released records in $T'$ correspond to the first $|T'|$ records in $T$. If $|T| = |T'|$, then all the records are released. The data set $E$ links the quasi-identifiers $q_i$ to the identifier $id_i$. The functions $Id$ and $Rid$ assign a record in $T'$ to the records in $E$, thus performing the re-identification of the records.
Table 4.1: Data sets in Example 4

<table>
<thead>
<tr>
<th>T</th>
<th>T'</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>x₁ = (q₁₁, c₁)</td>
<td>x₁' = (q₁₁', c₁)</td>
<td>e₁ = (q₁₁', i₁)</td>
</tr>
<tr>
<td>x₂ = (q₁₂, c₂)</td>
<td></td>
<td>e₂ = (q₁₂', i₂)</td>
</tr>
</tbody>
</table>

in T'. The function \( \text{Rid} \) is the re-identification function used by the intruder, while \( \text{Id} \) is assumed to be the correct re-identification function. If there is no record in T' corresponding to the identity (i.e., the identified record) \( e_i \in E \), then \( \text{Id} \) returns the empty set.

The goal of probabilistic \( k \)-anonymity is to limit the probability of performing the right linkage to at most \( 1/k \). With the above notations this requirement can be stated as: for all \( e_i \in E \) and for all \( \text{Rid}(\cdot) \)

\[
P(\text{Rid}(e_i) = \text{Id}(e_i)) \leq \frac{1}{k}
\]

This formula catches the essence of the definition of probabilistic \( k \)-anonymity: the probability of performing the right re-identification is not greater than \( 1/k \). However, by having the intruder use any possible function \( \text{Rid}(\cdot) \) to perform the re-identification, the details on how a rational intruder will proceed are hidden. Given a record \( e_i \), a rational intruder selects the record \( x_r \) in T' that has the greatest probability given the knowledge of \( T', E \), and the mechanism \( M \) used to generate T' from T.

The probability that \( x_i' \) corresponds to \( e_i \) equals the probability of obtaining \( q_i' \) from \( q_i' \) over the total probability of obtaining \( q_i' \) from any other record in E:

\[
P(X'(q_i')) = q_i' | T', E, M)
\]

\[
= \frac{P(X'(q_i') = q_i'}{\sum_{(q_j', i_d, \cdot) \in E} P(X'(q_j') = q_i')/M)}
\]

The intruder selects \( e_1 \) as his guess if \( P(X'(q_1') = q_1'| T', E, M) \geq P(X'(q_2') = q_1'| T', E, M) \) and \( e_2 \) otherwise.

In the previous example we have seen that, given a record in E, the linkage is performed to the record in T' that has greatest probability. If that probability is smaller than \( 1/k \), then the probability of performing the right linkage will also be smaller than

Example 4. Let us assume that T contains two records, and that only the first one is included in the anonymized data set. This situation is shown in Table 4.1. From the intruder’s point of view, \( x_1' \) corresponds to either the individual in \( e_1 \) or \( e_2 \). The best the intruder can do is to select the one that has the greatest probability given the knowledge of \( T', E \), and the mechanism \( M \) used to generate \( T' \) from \( T \).

The probability that \( x_1' \) corresponds to \( e_i \) equals the probability of obtaining \( q_1' \) from \( q_1' \), over the total probability of obtaining \( q_i' \) from any other record in E:

\[
P(X'(q_i') = q_1'| T', E, M)
\]

\[
= \frac{P(X'(q_i') = q_1')}{\sum_{(q_j', i_d, \cdot) \in E} P(X'(q_j') = q_1')}
\]

The intruder selects \( e_1 \) as his guess if \( P(X'(q_1') = q_1'| T', E, M) \geq P(X'(q_1') = q_1'| T', E, M) \) and \( e_2 \) otherwise.

In the previous example we have seen that, given a record in E, the linkage is performed to the record in T’ that has greatest probability. If that probability is smaller than \( 1/k \), then the probability of performing the right linkage will also be smaller than
correspond to the other record in \( P \) the probabilities. For example, the probability one identity in \( k \) \( k \) to exactly contain the identities for the individuals in \( T \) Example 5. In this example the amount of information in \( M \) following section, we propose to use data swapping as probability in Inequality (4.1) for an arbitrary mechanism However, the previous examples show that the computation of the re-identification probability in Inequality (4.1) must hold. Example 5. In this example the amount of information in \( T' \) has been increased, by adding the record \( x'_2 \). The new data sets are shown in Table 4.2. As \( E \) is assumed to exactly contain the identities for the individuals in \( T \), the intruder knows that if one identity in \( E \) corresponds to a specific record in \( T' \), the other identity in \( E \) must correspond to the other record in \( T' \). This must be taken into account when computing the probabilities. For example, the probability \( P(X'(q_1^E) = q'_1 | T', E, M) \) that \( q_1^E \) corresponds to \( q'_1 \) equals \( P(X'(q_1^E) = q'_1, X'(q_1^E) = q'_2 | T', E, M) \), which can be computed as

\[
P(X'(q_1^E) = q'_1, X'(q_1^E) = q'_2 | M) \leq \frac{1}{k}
\]

Example 5. In this example the amount of information in \( T' \) has been increased, by adding the record \( x'_2 \). The new data sets are shown in Table 4.2. As \( E \) is assumed to exactly contain the identities for the individuals in \( T \), the intruder knows that if one identity in \( E \) corresponds to a specific record in \( T' \), the other identity in \( E \) must correspond to the other record in \( T' \). This must be taken into account when computing the probabilities. For example, the probability \( P(X'(q_1^E) = q'_1 | T', E, M) \) that \( q_1^E \) corresponds to \( q'_1 \) equals \( P(X'(q_1^E) = q'_1, X'(q_1^E) = q'_2 | T', E, M) \), which can be computed as

\[
P(X'(q_1^E) = q'_1, X'(q_1^E) = q'_2 | M) \leq \frac{1}{k}
\]

The next example shows how the correct re-identification probability would be computed in the most general case.

Example 6. Assume data sets \( T \), \( T' \) and \( E \) as in Table 4.3. Contrary to Example 5, fixing a correspondence between a record in \( T' \) and a record in \( E \) does not completely fix the rest of the correspondences. We still have to consider all the possible combinations. The probability \( P(X'(q_i^E) = q_{i'} | T', E, M) \) that \( q_i^E \) corresponds to \( q_{i'} \) equals \( \sum P(X'(q_i^E) = q_{i'}, X'(q_i^E) = q_{i_2'}, \ldots, X'(q_i^E) = q_{i_m'} | T', E, M) \), where \( 1 < i_2 < \ldots < i_m \leq N \), and \( \{j_2, \ldots, j_m\} = \{2, \ldots, M\} \). This probability can be computed as

\[
\frac{\sum P(X'(q_i^E) = q_{i'}, X'(q_i^E) = q_{i_2'}, \ldots, X'(q_i^E) = q_{i_m'}, | M)}{\sum P(X'(q_i^E) = q_{i_1'}, \ldots, X'(q_i^E) = q_{i_m'}, | M)}
\]

where \( 1 \leq r_2 < \ldots < r_m \leq N \), and \( \{s_2, \ldots, s_M\} = \{2, \ldots, M\} \).

We have said that, to have probabilistic \( k \)-anonymity, Inequality (4.1) must hold. However, the previous examples show that the computation of the re-identification probability in Inequality (4.1) for an arbitrary mechanism \( M \) may be complex. In the following section, we propose to use data swapping as \( M \), which has the advantage of making the computation of the re-identification probability very simple.
Table 4.3: Data sets in Example 6

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = (q_1, c_1)$</td>
<td>$x'_1 = (q'_1, c_1)$</td>
<td>$e_1 = (q_{1E}, id_1)$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$x_N = (q_i, c_N)$</td>
<td>$x'_M = (q'_M, c_M)$</td>
<td>$e_N = (q_{NE}, id_N)$</td>
</tr>
</tbody>
</table>

Table 4.4: Data sets in the uninformed intruder scenario

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = (q_1, c_1)$</td>
<td>$x'_1 = (q'_1, c_1)$</td>
<td>$e_1 = (q_{1E}, id_1)$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$x_N = (q_i, c_N)$</td>
<td>$x'_N = (q'_N, c_N)$</td>
<td>$e_N = (q_{NE}, id_N)$</td>
</tr>
</tbody>
</table>

4.4 Probabilistic $k$-anonymity via microaggregation and swapping

The proposed method consists of two main steps: (i) partition the records in $T$ into groups of size $k$ and (ii) apply a permutation to the quasi-identifier attributes within each of the groups. This method can accommodate many variations, depending on how the partition step (i) is done.

Note that, as the same permutation is applied to all quasi-identifier attributes, the identity of the individual is not masked. However, the quasi-identifier attributes are dissociated from the confidential attributes, and therefore intruders can only guess the actual values corresponding to a confidential attribute with probability at most $1/k$. If leaking the mere presence of an individual in the data set is itself disclosive, then some of the quasi-identifier attributes must be considered confidential, which takes us to the informed intruder scenario.

We introduce first the method that offers protection against uninformed intruders. In other words, we assume that the attributes may be quasi-identifier attributes or confidential attributes, but not both. Later we extend the method to the scenario with informed intruders proposed in Section 4.1.

4.4.1 Uninformed intruders

In presence of uninformed intruders there is a clear separation between quasi-identifier and confidential attributes. Assuming that all records in $T$ are masked and included in $T'$, we have the data sets in Table 4.4.

Selecting a random sample from $T$ to create $T'$ is a sensible approach, as it introduces uncertainty on whether an individual whose data was collected has been included in the published data set. However, by assuming that all the individuals in $T$ have been included in $T'$, we provide the intruder with the best information available. Therefore, if we achieve probabilistic $k$-anonymity in this scenario, then we will also achieve it in a scenario where a random sample from $T$ is selected.
It is easy to see that the partition and swapping method described above satisfies probabilistic $k$-anonymity because

$$P(X'(q_i^E) = q_i'|T', E, M) = \begin{cases} \frac{1}{k} & \text{if } q_i \in G(id(q_i^E)) \\ 0 & \text{otherwise} \end{cases}$$

where $G(id(q_i^E))$ is the group of records of $T$ that contains the record corresponding to $q_i^E$.

The key point in the method is the partition step. A first approach is to partition the data set $T$ into random groups. This leads indeed not only to probabilistic $k$-anonymity, but to probabilistic $|T|$-anonymity, as the quasi-identifiers of a record can be swapped with the quasi-identifiers of any other record. Moreover, the risk of attribute disclosure is small. However, the impact on data quality can be substantial, because very different records may be swapped.

To achieve better data quality, the groups of records must be selected to be as homogeneous as possible, although this increases the risk of attribute disclosure. Our proposal is to generate the groups using a microaggregation algorithm ([23, 24]) over the quasi-identifier attributes. Microaggregation is a cardinality-constrained form of clustering in which the number of clusters (groups) is not fixed beforehand but the minimum cardinality of each group is required to be $k$. In the section devoted to informed intruders, there are some experimental results obtained by using the MDAV microaggregation algorithm ([24, 52]); MDAV attempts to maximize intra-group homogeneity using the least squares criterion and it yields groups with size $k$, except perhaps one group which has size between $k$ and $2k - 1$.

Other options in the selection of the groups of records are possible. For example, a variant of MDAV, known as V-MDAV ([91, 92]), may be used that performs clustering in groups of variable size and that is known to reduce the information loss in clustered data sets. The $\mu$-Approx microaggregation heuristic [27] offers also variable-sized groups and is proven to yield a clustering within a bound of the optimal clustering. Another possibility is to select the groups of records in such a way that the risk of attribute disclosure is reduced, by ensuring a certain diversity in the values of the confidential attributes within each group.

### 4.4.2 MDAV microaggregation for informed intruders

In the scenario for informed intruders presented in Section 4.1 we assumed the number of informed intruders to be the same as the number of confidential attributes in the data set. To be more specific, we consider the attributes: $A_0, A_1, \ldots, A_n$, with $A_0$ being a non-confidential quasi-identifier attribute, and $A_1, \ldots, A_n$ being confidential quasi-identifier attributes. Intruder $I_i$, for $i = 1$ to $n$, is assumed to know the values of all attributes except $A_i$.

To achieve the desired level of protection against all informed intruders, we apply the method presented for uninformed intruders once for each informed intruder, in order to dissociate the value of the confidential attribute unknown to this intruder from the rest of attributes. For each informed intruder, we use the quasi-identifiers and the confidential attribute shown in Table 4.5.
Table 4.5: Quasi-identifiers and confidential attribute for each informed intruder

<table>
<thead>
<tr>
<th>Intruder</th>
<th>Quasi-identifier attributes</th>
<th>Confidential attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>$A_0, A_2, \ldots, A_n$</td>
<td>$A_1$</td>
</tr>
<tr>
<td>$I_2$</td>
<td>$A_0, A_1, A_3, \ldots, A_n$</td>
<td>$A_2$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$I_n$</td>
<td>$A_0, A_1, \ldots, A_{n-1}$</td>
<td>$A_n$</td>
</tr>
</tbody>
</table>

One difficulty that we face with the previous approach is that dealing with informed intruders in sequence requires applying different permutations over different but overlapping sets of attributes of the original data set $T$ (the quasi-identifiers for each informed intruder). To overcome this difficulty we take the reverse approach: instead of performing the permutation over the quasi-identifier attributes, we apply the reverse permutation to the single confidential attribute unknown to the current intruder. In this way, each permutation acts over a different attribute and there are no overlaps.

### 4.4.3 Individual ranking microaggregation for informed intruders

The above observation regarding the application of the inverse permutation on the single unknown confidential attribute leads to single-attribute microaggregation, also called individual ranking microaggregation. Instead of multivariate microaggregation of quasi-identifier attributes, we do individual ranking microaggregation on the unknown confidential attribute. By doing so, the data quality of the published data set is increased, as the confidential attributes are only swapped across records with similar values (see [22] on the low information loss caused by individual ranking microaggregation). It may be argued that there is an increase in the attribute disclosure risk; however, this increase can be mitigated by increasing $k$.

One extra benefit from this approach is that, since microaggregation is performed on a single attribute, there is no need to normalize attributes as required by multivariate microaggregation to avoid scale problems.

### 4.5 Experimental results

We have implemented the following three methods:

- **MDAV-ID.** MDAV microaggregation is run on the quasi-identifier attributes to partition the data set in groups of size $k$ records. Within each group, quasi-identifiers are replaced by the group centroid in order to have identical quasi-identifiers for all records in the group. This is the procedure suggested in [24] and it achieves the standard notion of $k$-anonymity proposed in [87] in the sense that all quasi-identifiers within a group are made indistinguishable.

- **MDAV-SWAP.** This is the method described in Section 4.4.1 for probabilistic $k$-anonymity: MDAV microaggregation on the quasi-identifier attributes plus swapping within groups.
Table 4.6: Correlations to the confidential attributes in the original “Census” data set

<table>
<thead>
<tr>
<th></th>
<th>(A_7)</th>
<th>(A_8)</th>
<th>(A_9)</th>
<th>(A_{10})</th>
<th>(A_{11})</th>
<th>(A_{12})</th>
<th>(A_{13})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A_1)</td>
<td>.0038</td>
<td>-.027</td>
<td>-.024</td>
<td>.031</td>
<td>.032</td>
<td>.039</td>
<td>.036</td>
</tr>
<tr>
<td>(A_2)</td>
<td>.98</td>
<td>.14</td>
<td>.2</td>
<td>.73</td>
<td>.71</td>
<td>.72</td>
<td>.7</td>
</tr>
<tr>
<td>(A_3)</td>
<td>.44</td>
<td>-.12</td>
<td>-.058</td>
<td>.56</td>
<td>.55</td>
<td>.56</td>
<td>.55</td>
</tr>
<tr>
<td>(A_4)</td>
<td>.98</td>
<td>.2</td>
<td>.28</td>
<td>.73</td>
<td>.69</td>
<td>.71</td>
<td>.69</td>
</tr>
<tr>
<td>(A_5)</td>
<td>.78</td>
<td>.27</td>
<td>.27</td>
<td>.9</td>
<td>.85</td>
<td>.88</td>
<td>.86</td>
</tr>
<tr>
<td>(A_6)</td>
<td>.79</td>
<td>.13</td>
<td>.22</td>
<td>.59</td>
<td>.57</td>
<td>.57</td>
<td>.56</td>
</tr>
<tr>
<td>(A_7)</td>
<td>1</td>
<td>.17</td>
<td>.23</td>
<td>.72</td>
<td>.7</td>
<td>.71</td>
<td>.69</td>
</tr>
<tr>
<td>(A_8)</td>
<td>1</td>
<td>.45</td>
<td>-.17</td>
<td>-.19</td>
<td>-.17</td>
<td>-.17</td>
<td></td>
</tr>
<tr>
<td>(A_9)</td>
<td>1</td>
<td>.072</td>
<td>.061</td>
<td>.70</td>
<td>.075</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(A_{10})</td>
<td>1</td>
<td>.96</td>
<td>.98</td>
<td>.96</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(A_{11})</td>
<td>1</td>
<td>.91</td>
<td>.89</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(A_{12})</td>
<td>1</td>
<td>.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(A_{13})</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **IR-SWAP.** This is the method described in Section 4.4.2 above for probabilistic \(k\)-anonymity: individual ranking microaggregation on each confidential attribute plus swapping within groups.

The above methods have been tested with the “Census” and “EIA” reference data sets proposed in the European project CASC [13].

4.5.1 **“Census” data set**

The “Census” data set contains 1080 records with 13 continuous attributes. Following the approach in [24] we consider the first 6 attributes in “Census” to be non-confidential quasi-identifiers, and the last 7 attributes to be confidential.

To assess the data quality, we evaluate the correlations from all attributes to the confidential attributes. As the proposed methods for probabilistic \(k\)-anonymity do not modify non-confidential attributes, correlations between the latter have the same value as in the original data set. Means and variances also remain unchanged for all attributes, because swapping does not change the values taken by each original attribute.

As an example, we computed the correlations for: i) the original data set (see Table 4.6); ii) the \(k\)-anonymous data set resulting from MDAV-ID with \(k = 12\) (see Table 4.7); iii) the probabilistically \(k\)-anonymous data set resulting from MDAV-SWAP with \(k = 12\) (see Table 4.8); and the probabilistically \(k\)-anonymous data set resulting from IR-SWAP with \(k = 12\) (see Table 4.9). The values in these tables must be taken with caution: they are results from a single execution of the algorithms, and may change in another execution. Despite these words of caution, we observe that MDAV-SWAP and IR-SWAP result in correlation values closer to the original data set than those obtained with MDAV-ID. The results of IR-SWAP are closest to the original correlations.
## Table 4.7: Correlations to the confidential attributes in the data set obtained using MDAV-ID with $k = 12$ ("Census" data set)

<table>
<thead>
<tr>
<th></th>
<th>$A_7$</th>
<th>$A_8$</th>
<th>$A_9$</th>
<th>$A_{10}$</th>
<th>$A_{11}$</th>
<th>$A_{12}$</th>
<th>$A_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>-.0035</td>
<td>-.035</td>
<td>-.055</td>
<td>.034</td>
<td>.035</td>
<td>.042</td>
<td>.04</td>
</tr>
<tr>
<td>$A_2$</td>
<td>1</td>
<td>.18</td>
<td>.39</td>
<td>.8</td>
<td>.81</td>
<td>.8</td>
<td>.78</td>
</tr>
<tr>
<td>$A_3$</td>
<td>.79</td>
<td>-.17</td>
<td>.084</td>
<td>.89</td>
<td>.9</td>
<td>.89</td>
<td>.89</td>
</tr>
<tr>
<td>$A_4$</td>
<td>.99</td>
<td>.23</td>
<td>.45</td>
<td>.82</td>
<td>.8</td>
<td>.81</td>
<td>.8</td>
</tr>
<tr>
<td>$A_5$</td>
<td>.86</td>
<td>.18</td>
<td>.4</td>
<td>.94</td>
<td>.92</td>
<td>.94</td>
<td>.93</td>
</tr>
<tr>
<td>$A_6$</td>
<td>.95</td>
<td>.2</td>
<td>.43</td>
<td>.77</td>
<td>.76</td>
<td>.76</td>
<td>.75</td>
</tr>
<tr>
<td>$A_7$</td>
<td>1</td>
<td>.2</td>
<td>.41</td>
<td>.8</td>
<td>.8</td>
<td>.79</td>
<td>.78</td>
</tr>
<tr>
<td>$A_8$</td>
<td>1</td>
<td>.68</td>
<td>-.15</td>
<td>-.18</td>
<td>-.15</td>
<td>-.16</td>
<td></td>
</tr>
<tr>
<td>$A_9$</td>
<td>1</td>
<td>.18</td>
<td>.14</td>
<td>.17</td>
<td>.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{10}$</td>
<td>1</td>
<td>.98</td>
<td>.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{11}$</td>
<td>1</td>
<td>.97</td>
<td>.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{12}$</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{13}$</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Table 4.8: Correlations to the confidential attributes in the probabilistically $k$-anonymous data set obtained using MDAV-SWAP with $k = 12$ ("Census" data set)

<table>
<thead>
<tr>
<th></th>
<th>$A_7$</th>
<th>$A_8$</th>
<th>$A_9$</th>
<th>$A_{10}$</th>
<th>$A_{11}$</th>
<th>$A_{12}$</th>
<th>$A_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>-.0011</td>
<td>-.028</td>
<td>-.034</td>
<td>.032</td>
<td>.033</td>
<td>.036</td>
<td>.032</td>
</tr>
<tr>
<td>$A_2$</td>
<td>.81</td>
<td>.089</td>
<td>.17</td>
<td>.69</td>
<td>.67</td>
<td>.69</td>
<td>.67</td>
</tr>
<tr>
<td>$A_3$</td>
<td>.42</td>
<td>-.020</td>
<td>.091</td>
<td>.48</td>
<td>.47</td>
<td>.48</td>
<td>.43</td>
</tr>
<tr>
<td>$A_4$</td>
<td>.77</td>
<td>.093</td>
<td>.18</td>
<td>.68</td>
<td>.65</td>
<td>.68</td>
<td>.67</td>
</tr>
<tr>
<td>$A_5$</td>
<td>.72</td>
<td>.086</td>
<td>.16</td>
<td>.80</td>
<td>.76</td>
<td>.79</td>
<td>.77</td>
</tr>
<tr>
<td>$A_6$</td>
<td>.64</td>
<td>.086</td>
<td>.14</td>
<td>.54</td>
<td>.52</td>
<td>.54</td>
<td>.52</td>
</tr>
<tr>
<td>$A_7$</td>
<td>1</td>
<td>.12</td>
<td>.17</td>
<td>.69</td>
<td>.67</td>
<td>.66</td>
<td>.65</td>
</tr>
<tr>
<td>$A_8$</td>
<td>1</td>
<td>.19</td>
<td>-.013</td>
<td>-.022</td>
<td>-.042</td>
<td>-.011</td>
<td></td>
</tr>
<tr>
<td>$A_9$</td>
<td>1</td>
<td>.11</td>
<td>.10</td>
<td>.10</td>
<td>.10</td>
<td>.13</td>
<td></td>
</tr>
<tr>
<td>$A_{10}$</td>
<td>1</td>
<td>.76</td>
<td>.81</td>
<td>.87</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{11}$</td>
<td>1</td>
<td>.72</td>
<td>.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{12}$</td>
<td>1</td>
<td>.77</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{13}$</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4.9: Correlations to the confidential attributes in the probabilistically $k$-anonymous data set obtained using IR-SWAP with $k = 12$ (“Census” data set)

<table>
<thead>
<tr>
<th></th>
<th>$A_7$</th>
<th>$A_8$</th>
<th>$A_9$</th>
<th>$A_{10}$</th>
<th>$A_{11}$</th>
<th>$A_{12}$</th>
<th>$A_{13}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>.0041</td>
<td>-.017</td>
<td>-.018</td>
<td>.031</td>
<td>.038</td>
<td>.039</td>
<td>.038</td>
</tr>
<tr>
<td>$A_2$</td>
<td>.98</td>
<td>.13</td>
<td>.20</td>
<td>.73</td>
<td>.71</td>
<td>.72</td>
<td>.70</td>
</tr>
<tr>
<td>$A_3$</td>
<td>.44</td>
<td>-.12</td>
<td>-.041</td>
<td>.56</td>
<td>.55</td>
<td>.56</td>
<td>.55</td>
</tr>
<tr>
<td>$A_4$</td>
<td>.98</td>
<td>.19</td>
<td>.27</td>
<td>.73</td>
<td>.68</td>
<td>.71</td>
<td>.69</td>
</tr>
<tr>
<td>$A_5$</td>
<td>.78</td>
<td>.26</td>
<td>.26</td>
<td>.90</td>
<td>.85</td>
<td>.88</td>
<td>.86</td>
</tr>
<tr>
<td>$A_6$</td>
<td>.79</td>
<td>-.12</td>
<td>.21</td>
<td>.59</td>
<td>.57</td>
<td>.57</td>
<td>.56</td>
</tr>
<tr>
<td>$A_7$</td>
<td>1</td>
<td>.16</td>
<td>.23</td>
<td>.72</td>
<td>.69</td>
<td>.71</td>
<td>.69</td>
</tr>
<tr>
<td>$A_8$</td>
<td>1</td>
<td>.42</td>
<td>-.17</td>
<td>-.17</td>
<td>-.17</td>
<td>-.17</td>
<td>-.17</td>
</tr>
<tr>
<td>$A_9$</td>
<td>1</td>
<td>.077</td>
<td>.063</td>
<td>.075</td>
<td>.080</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{10}$</td>
<td>1</td>
<td>.95</td>
<td>.98</td>
<td>.96</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{11}$</td>
<td>1</td>
<td>.91</td>
<td>.89</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{12}$</td>
<td>1</td>
<td>.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{13}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

To obtain results with more statistical significance, we ran MDV-ID, MDAV-SWAP and IR-SWAP 100 times. In Table 4.10 we report the mean and the standard deviation of the absolute value of the difference between the correlations to the confidential attributes in the anonymized data set and the original data set. The better the data quality of the anonymized data set, the closer the mean and standard deviation to zero. A value close to one for the mean means that most of the dependencies between attributes have been lost.

Table 4.10 confirms what had been observed from the previous tables based on a single run: MDAV-SWAP offers better quality than MDV-ID, but IR-SWAP clearly offers the best quality among the three methods compared. For example, for the data set tried, similar data quality is obtained using MDV-ID with $k = 11$, MDAV-SWAP with $k = 25$ and IR-SWAP with $k = 300$. Hence, probabilistic $k$-anonymity turns out to be much more information-preserving than $k$-anonymity.

### 4.5.2 “EIA” data set

Due to space constraints, empirical results for the “EIA” data set are more succinctly presented. Table 4.11 reports an evaluation for the “EIA” data set analogous to the one reported in Table 4.10 for the “Census” data set. Like before, we observe that MDAV-SWAP performs better than MDV-ID, but IR-SWAP is clearly the best of the three methods.

### 4.6 Conclusions and future research

$k$-Anonymity is a broadly used privacy property that focuses on protecting against identity disclosure. In a $k$-anonymous data set, for each record there are at least $k - 1$ other
Table 4.10: Mean and standard deviation of the absolute value of the difference between the correlations in the original and the anonymized data sets (“Census” data set)

<table>
<thead>
<tr>
<th>k</th>
<th>MDAV-ID mean</th>
<th>MDAV-ID st.dev.</th>
<th>MDAV-SWAP mean</th>
<th>MDAV-SWAP st.dev.</th>
<th>IR-SWAP mean</th>
<th>IR-SWAP st.dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>.055</td>
<td>.064</td>
<td>.037</td>
<td>.045</td>
<td>.0021</td>
<td>.0041</td>
</tr>
<tr>
<td>7</td>
<td>.062</td>
<td>.071</td>
<td>.048</td>
<td>.056</td>
<td>.0022</td>
<td>.0039</td>
</tr>
<tr>
<td>9</td>
<td>.069</td>
<td>.078</td>
<td>.055</td>
<td>.064</td>
<td>.0028</td>
<td>.0049</td>
</tr>
<tr>
<td>11</td>
<td>.078</td>
<td>.085</td>
<td>.061</td>
<td>.070</td>
<td>.0038</td>
<td>.0068</td>
</tr>
<tr>
<td>25</td>
<td>.11</td>
<td>.11</td>
<td>.091</td>
<td>.093</td>
<td>.0061</td>
<td>.012</td>
</tr>
<tr>
<td>100</td>
<td>.17</td>
<td>.15</td>
<td>.19</td>
<td>.17</td>
<td>.020</td>
<td>.030</td>
</tr>
<tr>
<td>200</td>
<td>.29</td>
<td>.27</td>
<td>.31</td>
<td>.28</td>
<td>.044</td>
<td>.047</td>
</tr>
<tr>
<td>300</td>
<td>.38</td>
<td>.39</td>
<td>.37</td>
<td>.34</td>
<td>.087</td>
<td>.071</td>
</tr>
</tbody>
</table>

Table 4.11: Mean and standard deviation of the absolute value of the difference between the correlations in the original and the anonymized data sets (“EIA” data set)

<table>
<thead>
<tr>
<th>k</th>
<th>MDAV-ID mean</th>
<th>MDAV-ID st.dev.</th>
<th>MDAV-SWAP mean</th>
<th>MDAV-SWAP st.dev.</th>
<th>IR-SWAP mean</th>
<th>IR-SWAP st.dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>.018</td>
<td>.017</td>
<td>.017</td>
<td>.035</td>
<td>.00064</td>
<td>.00075</td>
</tr>
<tr>
<td>7</td>
<td>.02</td>
<td>.017</td>
<td>.024</td>
<td>.05</td>
<td>.0012</td>
<td>.0018</td>
</tr>
<tr>
<td>9</td>
<td>.034</td>
<td>.031</td>
<td>.028</td>
<td>.053</td>
<td>.0015</td>
<td>.0018</td>
</tr>
<tr>
<td>11</td>
<td>.039</td>
<td>.036</td>
<td>.029</td>
<td>.052</td>
<td>.0019</td>
<td>.0023</td>
</tr>
<tr>
<td>25</td>
<td>.085</td>
<td>.078</td>
<td>.043</td>
<td>.081</td>
<td>.0063</td>
<td>.0072</td>
</tr>
<tr>
<td>50</td>
<td>.13</td>
<td>.12</td>
<td>.053</td>
<td>.089</td>
<td>.011</td>
<td>.011</td>
</tr>
<tr>
<td>100</td>
<td>.15</td>
<td>.14</td>
<td>.058</td>
<td>.092</td>
<td>.029</td>
<td>.037</td>
</tr>
<tr>
<td>200</td>
<td>.19</td>
<td>.18</td>
<td>.09</td>
<td>.11</td>
<td>.093</td>
<td>.074</td>
</tr>
<tr>
<td>300</td>
<td>.2</td>
<td>.18</td>
<td>.12</td>
<td>.13</td>
<td>.14</td>
<td>.091</td>
</tr>
</tbody>
</table>
records sharing the same values for all the quasi-identifier attributes. Hence, enforcing $k$-anonymity implies variability loss and therefore quality loss. This is especially serious in a scenario with informed intruders, who know the values of some confidential attributes: the confidential attributes known by the informed intruder can be viewed as additional quasi-identifiers. The more quasi-identifier attributes, the more data quality loss is caused by $k$-anonymity.

To mitigate the above problem, we have introduced the notion of probabilistic $k$-anonymity. Like standard $k$-anonymity, probabilistic $k$-anonymity guarantees that the probability of correct re-identification is at most $1/k$, but without explicitly requiring that the quasi-identifier attributes take identical values within each group of $k$ records. We have presented two computational methods to reach probabilistic $k$-anonymity, based on microaggregation and swapping. Experimental work shows that, for a fixed re-identification probability $1/k$, the new methods are much more quality-preserving than standard $k$-anonymity enforcement.

Future research will combine probabilistic $k$-anonymity with other properties like $l$-diversity or $t$-closeness in view of reducing the quality loss incurred to protect against attribute disclosure.
Chapter 5

Sensitivity-independent differential privacy via knowledge refinement

5.1 Introduction

Differential privacy [31, 32] is a privacy property of queryable databases that is normally implemented using output perturbation. The disclosure risk limitation offered by differential privacy is based on the limitation of the effect that any single individual has on a query response. If the influence of any single individual on the query response is small, publishing that response involves only a small disclosure risk for any individual.

Any mechanism used to achieve differential privacy may be seen as the application of a perturbation to the real value of the query response. The original proposal [31, 32] to attain differential privacy masks the query response by adding a Laplace-distributed noise whose magnitude is proportional to the global sensitivity of the query function (the global sensitivity is the maximum variation of the query function between any two data sets differing in one record, also known as neighbor data sets). The global sensitivity of the query function may be substantially higher than the local sensitivity at a certain data set (variation of the query function between that data set and its neighbors); hence, adding noise based on global sensitivity may overprotect most data sets. The approach in [73] tries to avoid this problem by adjusting for each data set the magnitude of the noise to a so-called smooth sensitivity based on the local sensitivities of the query function. Other mechanisms proposed to achieve differential privacy include the exponential mechanism [65], that introduces the concept of response utility, and some mechanisms designed for specific types of queries, such as [10, 105].

All the methods mentioned above are at some point concerned with the (global, smooth or local) sensitivity of the query function. Such mechanisms present two main problems: (i) deciding what amount of noise to add may require complex computations (such as computing the global sensitivity [31] or the smooth sensitivity [73] of the query...
function) and thus noise addition may be difficult to automate, and (ii) in some cases, the amount of noise that needs to be added is so large that the output may bear little resemblance to the real query response, and thus the response may be misleading.

Differential privacy was introduced as a query-response mechanism where the database is held by a trusted party. Users access the database by submitting queries to it. Queries are mediated by an access mechanism that answers them in a differentially private way. Sometimes this setting is simply too restrictive, because users may want access to the entire database. Hence, as a compromise one should allow for queries that are as flexible as possible. However, the literature on the generation of differentially private data sets is brief [10, 64, 1, 15, 51] and the available proposals are mainly centered on count queries. The reason for focusing on counts is that current methods to attain differential privacy behave well for count queries. However, even for count queries there may be a big impact on data utility, for example when the data are sparse [64]. Our proposal to reach differential privacy is intended to be one step ahead towards enabling more complex queries to more complex microdata sets.

5.1.1 Contribution and plan of this chapter

We introduce a mechanism to achieve differential privacy that works by refining the prior knowledge/beliefs of the database user as much as possible, given the constraints set by differential privacy. Our mechanism depends only on the prior knowledge and on the level of protection that we want to achieve. It is completely independent from the sensitivity of the actual query function, and thus no complex sensitivity computations are required.

This mechanism avoids problem (i) above, as no complex computations are required. Regarding problem (ii), the mechanism guarantees that the response yields increased utility over the prior knowledge that the user had.

Section 5.2 introduces the idea of knowledge refinement for differential privacy. Section 5.3 describes a general knowledge refinement mechanism, both for continuous and discrete responses. Section 5.4 evaluates the level of differential privacy that we get for multicomponent queries in terms of the components. Section 5.5 builds an interactive mechanism on top of knowledge refinement and shows that it is safe against malicious users. Section 5.6 compares the knowledge refinement approach with Laplace noise addition. Section 5.7 contains a discussion and Section 5.8 summarizes conclusions.

5.2 Refining prior knowledge to achieve differential privacy

The definition of differential privacy states that the probability that the response belongs to any subset of its range must be similar regardless of whether any specific individual is included or not in the data set.

Definition 21. A randomized function \( \kappa \) gives \( \varepsilon \)-differential privacy if, for all data sets \( D_1, D_2 \) such that one can be obtained from the other by adding or removing a single
record, and all \( S \subset \text{Range}(\kappa) \)

\[
P(\kappa(D_1) \in S) \leq \exp(\varepsilon) \times P(\kappa(D_2) \in S) \tag{5.1}
\]

A usual approach to satisfying the requirements of Definition 21 is noise addition: first, the real value of the query response is computed and, then, a random noise is added to mask it. A Laplace distribution with zero mean and a scale parameter that depends on the variability of the query function is commonly used for noise addition.

Our proposal is not based on masking the true value of the response by adding some noise, but on modifying the prior knowledge of the database user on the response. When a query is submitted to the database, the user submits at the same time her knowledge/beliefs about the response. We think of this prior knowledge as the probability distribution that the user expects for the response. For example, in case the user has absolutely no idea about the possible result for a query \( f \), the probability distribution to be used is the uniform distribution over the range of \( f \) (assuming that this range is bounded). The access mechanism modifies this prior knowledge to fit the real value of the response as much as possible given the constraints imposed by differential privacy.

Some users may be reluctant to provide detailed prior knowledge, because they regard doing so as giving information about themselves to the database. We should usually think of the prior knowledge as the information about the response that is publicly available. Providing the database with such a prior knowledge reveals nothing about the database user. If the database user has information that is not publicly available, she must decide whether to use it as prior knowledge or not; the more accurate the prior knowledge, the more accurate the response will be. We will see in Section 5.6 that, even when little prior knowledge is assumed, knowledge refinement may be superior, in terms of data quality, to noise addition approaches. Therefore, it may make sense to use knowledge refinement even if the database user is not willing to provide all her actual prior knowledge.

Definition 22. Given a query function \( f \), the prior knowledge about the response \( f(D) \) is the probability distribution \( P_f \), defined over \( \text{Range}(f) \), that the user expects for the response to \( f \).

The more concentrated the probability mass of \( P_f \) around the real value of the response to \( f \), the more accurate is the user’s prior knowledge. In general, as the user knows the query \( f \) and the set of possible databases \( D \), one may expect her to have some prior knowledge about the response \( f(D) \). The better the knowledge the user has on the actual database \( D \), the more accurate is the prior knowledge the user can provide to the response mechanism. If the user’s prior knowledge is wrong, the accuracy of the response may suffer. However, whatever the prior knowledge, the refinement procedure guarantees that the output distribution is more accurate than the prior knowledge.

If the query function \( f \) has multiple components (dimension \( n > 1 \)), the joint probability distribution must be provided. If the components of \( f \) are independent, specifying the marginal distribution for each component is enough to compute the joint distribution. This will also be the case if the components are not independent but the user has no knowledge about the relationship among them.

The access mechanism is run by the database holder as follows:
1. Receive the query $f$ and the prior knowledge $P_f$ from the database user.

2. Compute the actual value of the query response, $f(D)$.

3. Modify $P_f$ to adjust it to $f(D)$ as much as possible, given the constraints imposed by differential privacy.

4. Randomly sample the distribution resulting from the previous step, and return the sampled value as the response to $f$ evaluated at $D$.

Even though knowledge refinement works by adjusting the prior knowledge, the output is not the adjusted distribution but a sample from it. This is the usual approach in differential privacy; only a sample from the output distribution is returned. Returning the output distribution itself would leak too much information; in some cases, it could be used to determine the exact value of the query response.

Note that the user cannot pretend to have more knowledge than she actually has: sending a guess as $P_f$ will most likely be wrong and worsen the response quality. Also, we show in Section 5.5 that using several different (fake) prior knowledge distributions to mount adaptive attacks does not succeed in breaking $\varepsilon$-differential privacy.

The critical step is the adjustment of the prior knowledge to the real query response. To perform this adjustment, we distinguish two types of queries: statistical queries and individual queries. We call statistical queries those whose outcome depends on multiple individuals, while individual queries are those that depend on a single individual. It will be shown below that a finer adjustment of the prior knowledge is feasible for individual queries. We start by focusing on statistical queries, but, before formally specifying the response mechanism, we give an example to illustrate what we intend to do.

Example 7. Assume a query function $f$ that is known to return a value within the interval $[0, 1]$. Assume also that the database user has no further knowledge about the query response, i.e. her prior knowledge is $U[0, 1]$, the uniform distribution over $[0, 1]$.

To refine the prior knowledge, we modify its density by applying two multiplicative factors: $\alpha_u \geq 1$ to the points near $f(D)$, and $\alpha_d \leq 1$ to the points farther from $f(D)$. In this way, the probability of obtaining as the response a value near the actual response $f(D)$ is increased with respect to the prior knowledge, while the probability of obtaining a distant value is decreased. Figure 5.1 shows the probability distribution resulting from applying the procedure described above for a pair of neighbor data sets $D$ and $D'$.

To obtain $\varepsilon$-differential privacy, the density at a given point for the response to $f(D)$ must be a factor within the interval $[e^{-\varepsilon}, e^\varepsilon]$ of the density at the same point for the response to $f(D')$. Check, for example, the point 0.6 in Figure 5.1: on the left-hand side distribution, the point is far from the real response and thus a factor $\alpha_d$ is applied; on the right-hand side distribution, the point is near the real response and the factor applied is $\alpha_u$. For the $\varepsilon$-differential privacy condition to hold, it must be $\alpha_u / \alpha_d \leq e^\varepsilon$.

We can also think in the reverse way: given two constants $\alpha_u \geq 1$ and $\alpha_d \leq 1$, the level of differential privacy achieved by this response mechanism is $\varepsilon = \ln(\alpha_u / \alpha_d)$.

Note that, to obtain a valid density function from the above modification, the set of points over which each of the factors $\alpha_u$ and $\alpha_d$ are applied must be selected in such a
way that the total probability mass of the resulting distribution equals 1. If we denote by \( \mathcal{U}_u \) the set over which we apply the factor \( \alpha_u \), for the total probability mass of the adjusted distribution to be 1, we must have \( \alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1 \). If the prior knowledge is an absolutely continuous distribution, as in Example 7, for any pair of values \( \alpha_u \geq 1 \text{ and } \alpha_d \leq 1 \) it is possible to select a set \( \mathcal{U}_u \) in such a way that \( \alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1 \) is satisfied. The reason is that we can select the set \( \mathcal{U}_u \) to have any probability mass between 0 and 1. If the prior knowledge distribution is not absolutely continuous, it may not be possible to find a set \( \mathcal{U}_u \) with the required probability mass for the given values \( \alpha_u \) and \( \alpha_d \). This section assumes that such a set \( \mathcal{U}_u \) exists. In Section 5.3, we specify a general algorithm that works for any prior knowledge distribution.

The following proposition formalizes the ideas discussed in the previous example.

**Proposition 3.** Let \( f: \mathcal{D} \to \mathbb{R}^n \) be a query function and let \( P_f \) be the prior knowledge for \( f(D) \). Let \( \alpha_u \geq 1 \text{ and } \alpha_d \leq 1 \) be such that \( \alpha_u = e^\varepsilon \alpha_d \). Let \( \mathcal{U}_u \) be an environment of \( f(D) \) satisfying \( \alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1 \). The response mechanism that returns a value randomly sampled from the distribution obtained by modifying \( P_f \) through multiplication of the probability mass of the points in \( \mathcal{U}_u \) by \( \alpha_u \), and multiplication of the probability mass of the points outside \( \mathcal{U}_u \) by \( \alpha_d \), satisfies \( \varepsilon \)-differential privacy.

When the query \( f \) returns a value related to a single individual, the mechanism in Proposition 3 can be improved. In that case, there are only two possibilities for the response: (i) if the individual we are asking about is not in the database, the distribution of the response equals the prior knowledge distribution, and (ii) if the individual is in the database, the distribution for the response will be the result of refining the prior knowledge. To satisfy \( \varepsilon \)-differential privacy, we only need to guarantee that the distribution resulting from (i) and (ii) does satisfy the limitation on the knowledge gain imposed by differential privacy. In other words, the output distribution need only be compared to the prior knowledge. The conditions that must hold are \( 1 \leq \alpha_u \leq e^\varepsilon \) and \( e^{-\varepsilon} \leq \alpha_d \leq 1 \).

Note that, by choosing \( \alpha_u = e^\varepsilon \) and \( \alpha_d = e^{-\varepsilon} \), the level of differential privacy that we can guarantee for a statistical query function (depending on multiple individuals) is \( 2\varepsilon \), while for an individual query (whose outcome depends on a single individual), we double the guarantee to \( \varepsilon \).
Proposition 4. Let \( f: \mathcal{D} \to \mathbb{R}^n \) be an individual query in the above sense and let \( P_f \) be the prior knowledge distribution for \( f \). Let \( \alpha_u = e^\epsilon \) and \( \alpha_d = e^{-\epsilon} \). Let \( \mathcal{U}_u \) be an environment of \( f(D) \) satisfying \( \alpha_u P_f(\mathcal{U}_u) + \alpha_d(1 - P_f(\mathcal{U}_u)) = 1 \). The response mechanism that returns a value randomly sampled from the distribution obtained by modifying \( P_f \) through multiplication of the probability mass of the points in \( \mathcal{U}_u \) by \( \alpha_u \), and multiplication of the probability mass of the points outside \( \mathcal{U}_u \) by \( \alpha_d \), satisfies \( \epsilon \)-differential privacy.

5.3 A general algorithm for knowledge refinement

Propositions 3 and 4 above state that, given appropriate factors \( \alpha_u \) and \( \alpha_d \) and a set \( \mathcal{U}_u \) with the required probability mass, the knowledge refinement mechanism satisfies \( \epsilon \)-differential privacy. However, some details were left aside in the previous section: (i) how is the set \( \mathcal{U}_u \) selected?, and (ii) can we still apply knowledge refinement if a set \( \mathcal{U}_u \) with the required probability mass does not exist? This section gives a more detailed view of the knowledge refinement mechanism and answers the two aforementioned questions.

Knowledge refinement works by increasing the probability mass of the points near \( f(D) \), and by decreasing the probability mass of the rest of points in such a way that the total probability mass equals one. In Example 7 there was a natural way to determine the set \( \mathcal{U}_u \): the points closest to \( f(D) \) in absolute value. However, such a natural way does not always exist, as illustrated in the next example.

Example 8. To determine the form of the set \( \mathcal{U}_u \) for a query function with two components, say \( f = (f_1, f_2) \), we use a distance function defined over the range of \( f \), namely \( d: \text{Range}(f_1) \times \text{Range}(f_2) \to [0, \infty) \). If \( d \) does not treat \( f_1 \) and \( f_2 \) symmetrically, then one component is given priority over the other. In fact, there is no natural way to define \( d \) and hence \( \mathcal{U}_u \). Such definitions are application-dependent.

Table 5.1 shows some distance functions that are appropriate for a query with a single component in terms of the type of the result. We do not provide any distance for multivariate queries because such distances are very application-dependent, as pointed out in Example 8.

Table 5.1: Example distance functions for univariate query functions depending on the type of query result

<table>
<thead>
<tr>
<th>Query result</th>
<th>Range(( f ))</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>continuous</td>
<td>( \mathbb{R} )</td>
<td>( d(x, y) =</td>
</tr>
<tr>
<td>nominal</td>
<td>{( c_1, \ldots, c_n )}</td>
<td>( d(c_i, c_j) = 0 ) if ( i = j )</td>
</tr>
<tr>
<td>ordinal</td>
<td>{( c_1, \ldots, c_n )}</td>
<td>( d(c_i, c_j) =</td>
</tr>
</tbody>
</table>

Note that when we feed the knowledge refinement algorithm with a certain distance function, we are instructing it with the sets that we want to favor. Given a value \( f(D) \),
we modify the probability that the prior knowledge assigns to the points in $\text{Range}(f)$ according to the distance $d$. If a point at distance $r$ is being applied a factor $\alpha_1$, all points at distance $r$ must be applied the same factor, and points at a shorter distance must be applied a factor $\alpha_2$ with $\alpha_2 \geq \alpha_1$. Therefore, the set $U_u$ of points that has its probability increased must be of the form $U_{f(D),r}$, or $U_{f(D),r'}$, for some $r \in [0, \infty)$, where:

$$U_{f(D),r}^1 = \{ x \in \text{Range}(f) : d(f(D), x) \leq r \}$$

$$U_{f(D),r}^2 = \{ x \in \text{Range}(f) : d(f(D), x) < r \}$$

(5.2)

The set $U_d$ of points that has its probability decreased is the complement of $U_u$, that is, $U_d = \text{Range}(f) \setminus U_u$.

We want to choose two multiplicative factors $\alpha_u$ and $\alpha_d$ to modify the probability mass of $U_u$ and $U_d$, respectively. Factors $\alpha_u$ and $\alpha_d$ must be selected so that differential privacy holds and the total probability mass of the resulting modified distribution equals one.

Table 5.2 shows the form of factors $\alpha_u$ and $\alpha_d$ for the two types of queries considered in Section 5.2: individual and statistical. For the case of individual queries, the differential privacy condition need only hold between the distribution of the response and the prior knowledge. Any pair of values $\alpha_u \in [1, e^\varepsilon]$ and $\alpha_d \in [e^{-\varepsilon}, 1]$ yields $\varepsilon$-differential privacy; however, $\alpha_u = e^\varepsilon$ and $\alpha_d = e^{-\varepsilon}$ yield the greatest knowledge gain.

For statistical queries, the condition must hold for each pair of distributions for the response to the query over data sets that differ in a single record. Therefore, we must have $\alpha_u/\alpha_d \leq e^\varepsilon$. Same as for individual queries, the greatest knowledge gain is achieved when $\alpha_u/\alpha_d = e^\varepsilon$. The actual values of $\alpha_u$ and $\alpha_d$ must belong to the intervals $[1, e^\varepsilon]$ and $[e^{-\varepsilon}, 1]$, respectively, but they can be freely chosen, as long as $\alpha_u/\alpha_d \leq e^\varepsilon$ holds and the total probability mass is one:

$$\alpha_u P_f(U_u) + \alpha_d P_f(U_d) = 1$$

(5.3)

Table 5.2: Form of the factors $\alpha_u$ and $\alpha_d$ for individual and statistical queries

<table>
<thead>
<tr>
<th>Type of query</th>
<th>Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>individual</td>
<td>$\alpha_u = e^\varepsilon$, $\alpha_d = e^{-\varepsilon}$</td>
</tr>
<tr>
<td>statistical</td>
<td>$\alpha_u \in [1, e^\varepsilon]$, $\alpha_d \in [e^{-\varepsilon}, 1]$ with $\alpha_u/\alpha_d = e^\varepsilon$</td>
</tr>
</tbody>
</table>

For statistical queries, the specific values selected for $\alpha_u$ and $\alpha_d$ determine the maximum knowledge gain for the points in $U_u$ and $U_d$, where the gain is understood as the modification w.r.t. the prior knowledge $P_f$. Assuming that $\alpha_u/\alpha_d = e^\varepsilon$ holds, a greater value for $\alpha_u$ provides increased knowledge gain for the points in $U_u$, but it also results in a greater value for $\alpha_d$, because otherwise $\alpha_u/\alpha_d \leq e^\varepsilon$ would not be satisfied; this implies decreasing the knowledge gain for the points in $U_d$ with respect to the prior knowledge.
For fixed values of the factors $\alpha_u$ and $\alpha_d$, from Equation (5.3) and $P_f(U_d) = 1 - P_f(U_u)$, we have:

\[
\begin{align*}
P_f(U_u) &= \frac{\alpha_u - 1}{\alpha_u - \alpha_d} \\
P_f(U_d) &= \frac{1 - \alpha_d}{\alpha_u - \alpha_d}
\end{align*}
\]

For continuous prior knowledge, it is always possible to select sets $U_u$ and $U_d$ with the above probability masses. In this case, the knowledge refinement mechanism is very simple: apply factor $\alpha_u$ to $U_u$ and factor $\alpha_d$ to $U_d$, as stated in Propositions 3 and 4.

For other kinds of prior knowledge, the sets $U_u$ and $U_d$ with the required probability masses may not exist. In such cases, we still want to apply the factor $\alpha_u$ to the greatest possible set of points closest to $f(D)$, and the factor $\alpha_d$ to the greatest possible set of points farthest from $f(D)$, thus achieving the maximum knowledge gain at such points. We denote $U'_u$ the set that is applied factor $\alpha_u$, and $U'_d$ the set that is applied factor $\alpha_d$. For the remaining points we adjust their factor to have a total probability mass of one. See Algorithm 5 for a detailed description of the process; this algorithm is run by the database holder.

It is easy to check that the total probability mass of the distribution equals one, no matter whether the then or the else option of the if statement of Algorithm 5 is taken. Regarding the differential privacy condition, we have already seen that it holds for the then case. For the else case, differential privacy also holds, because $\alpha_{ud}$ belongs to the interval $[\alpha_d, \alpha_u]$.

Differential privacy is usually criticized for the low utility of the results it provides [89]. Several relaxations of $\varepsilon$-differential privacy have been proposed; in particular, the authors of [33] propose $(\varepsilon, \delta)$-differential privacy (a.k.a $(\varepsilon, \delta)$-indistinguishability), and $(\varepsilon, \delta)$-probabilistic differential privacy. The former property relaxes the strict requirement of differential privacy by adding a non-zero $\delta$. The latter property allows arbitrarily large knowledge gains within probability $\delta$. Let us briefly review $(\varepsilon, \delta)$-privacy and sketch how prior knowledge refinement can achieve it.

**Definition 23.** A randomized function gives $(\varepsilon, \delta)$-differential privacy if, for all data sets $D_1$, $D_2$ such that one can be obtained from the other by adding or removing a single record, and all $S \subset \text{Range}(\kappa)$

\[
P(\kappa(D_1) \in S) \leq \exp(\varepsilon) \times P(\kappa(D_2) \in S) + \delta \tag{5.4}
\]

As $\varepsilon$-differential privacy implies $(\varepsilon, \delta)$-differential privacy, Algorithm 5 can be used to obtain $(\varepsilon, \delta)$-differential privacy. However, a simple modification to Algorithm 5 can offer better data utility while still satisfying $(\varepsilon, \delta)$-differential privacy (but no longer $\varepsilon$-differential privacy). We do not provide a formal algorithm with the required modifications, but the idea is to use the extra margin $\delta$ to increase the probability at $f(D)$ and reduce it at the points farthest from $f(D)$.

Just like it happened for $\varepsilon$-differential privacy, the improvement of $(\varepsilon, \delta)$-privacy for individual queries is greater than for statistical queries. For an individual query, we only need to compare the distribution of the response with the prior knowledge (see Figure 5.2). As the prior knowledge is not modified, we can modify the response by adding $\delta$ to the probability mass of $f(D)$, and subtract $\delta$ from the tails of the distribution.
Algorithm 5 Knowledge refinement algorithm to respond to query \( f(D) \) for a general prior knowledge

Input parameters: query \( f \), prior knowledge \( P_f \) of the database user, distance function \( d \), factors \( \alpha_u \) and \( \alpha_d \) from the database holder.

1. Compute the actual value of the query response, \( f(D) \).

2. Modify \( P_f \) to adjust it to \( f(D) \) as much as possible, given the constraints imposed by differential privacy. This is done as follows:
   (a) Let \( p_u = (\alpha_u - 1)/(\alpha_u - \alpha_d) \).
   (b) Let \( p_d = (1 - \alpha_d)/(\alpha_u - \alpha_d) \).
   (c) if there exists a set \( U_u \) of the form \( U_{1f(D),r} \) or \( U_{2f(D),r} \) (see Expression 5.2) with \( P_f(U_u) = p_u \) then
      Build the distribution of the response to \( f(D) \) by applying the factor \( \alpha_u \) to \( U_u \), and \( \alpha_d \) to \( \text{Range}(f) \setminus U_u \).
   else
      i. Find the maximal set \( U_u \) of the form \( U_{1f(D),r} \) or \( U_{2f(D),r} \) with \( P_f(U_u) < p_u \).
      ii. Find the maximal set \( U_d \) of the form \( \text{Range}(f) \setminus U_{1f(D),r} \) or \( \text{Range}(f) \setminus U_{2f(D),r} \) with \( P_f(U_u) < p_d \).
      iii. Let \( p_{ud} = 1 - P_f(U_u) - P_f(U_d) \) be the probability of the points not in \( U_u \cup U_d \).
      iv. Let \( \alpha_{ud} = (1 - \alpha_u p_u - \alpha_d p_d)/(1 - p_u - p_d) \) be the factor to be applied to \( \text{Range}(f) \setminus (U_u \cup U_d) \).
      v. Build the distribution of the response to \( f(D) \) by applying:
         * factor \( \alpha_u \) to points in \( U_u \)
         * factor \( \alpha_d \) to points in \( U_d \)
         * factor \( \alpha_{ud} \) to points in \( \text{Range}(f) \setminus (U_u \cup U_d) \).

3. Randomly sample the distribution resulting from the previous step, and return the sampled value as the response to \( f \) evaluated at \( D \).
For a statistical query, we also want to increase the probability mass of the actual response $f(D)$, while reducing the probability mass of the set $S'_{f(D)}$ of points farthest from $f(D)$. Although other schemes are possible, a sensible choice is to have the probability mass of $f(D)$ increased by the same amount $\delta'$, whatever the data set $D$. As we have to keep the total probability mass equal to one, we must decrease the probability of $S'_{f(D)}$ by $\delta'$. Now, since we can select data sets $D_1$ and $D_2$ such that $f(D_1)$ belongs to $S'_{f(D_2)}$, for Inequality (5.4) to hold for $S'_{f(D_2)}$, it must be $\delta' = \delta/2$ (it can also be $\delta' < \delta/2$, but then we are not taking advantage of the whole $\delta$ margin).
5.4 Differential privacy in multicomponent queries

The knowledge refinement mechanism as introduced in Section 5.2 is independent of the number of components of the query function. However, for the case of multicomponent queries, we can relate the level of differential privacy for the multicomponent query to the level of differential privacy of the components. If we have a query \( f = (f_1, \ldots, f_n) \) and for each of the components, \( f_i \), we get an \( \varepsilon_i \)-differentially private response, then we get a \( \sum_{i=1}^{n} \varepsilon_i \)-differentially private response for \( f \). This is in fact a property of \( \varepsilon \)-differential privacy, hence a proof for our specific mechanism is not required (see [65]).

The above result on multicomponent queries can be improved when each of the queries refers to a disjoint set of individuals. For the noise addition mechanism, it easy to see that, when performing queries \( f_1, \ldots, f_n \) that refer each to a disjoint set of individuals, the global sensitivity equals the maximum of the sensitivities of the individual queries [31]. The reason is that, by adding or removing a single individual from the data set, only one of the queries is affected. This is a good property, as it guarantees \( \max(\varepsilon_i) \)-differential privacy instead of \( \sum \varepsilon_i \)-differential privacy. Our goal is to show that this property can also be achieved for our proposal. In fact, we will show further on that this is also a general property of differential privacy. We start with an example.

Example 9. Let \( D \) be a database with two attributes: an identifier \( ID \) and a Boolean attribute \( B \). Let \( f_1 \) and \( f_2 \) be queries that return the value of \( B \) for individuals 1 and 2, respectively. Let the prior knowledge for both queries be the independent uniform distribution over the set \( \{0, 1\} \), which assigns a prior probability 0.5 to each of the possible outcomes for each query. To respond to \( f_1 \) in an \( \varepsilon \)-differentially private way with \( \varepsilon = 1 \), we select factors \( \alpha_u = e^\varepsilon \) and \( \alpha_d = e^{-\varepsilon} \) that modify the prior knowledge. The same factors are selected for \( f_2 \). Now we want to check whether the combination of responses to \( f_1 \) and \( f_2 \) is still \( \varepsilon \)-differentially private.

For the sake of simplicity, we assume that both individuals are in \( D \), and that \( f_1(D) = 0 \) and \( f_2(D) = 0 \). For the rest of cases we would proceed in a similar way. Figure 5.4 shows the prior knowledge and the output distribution for both query functions \( f_1 \) and \( f_2 \). Indeed, by setting \( \alpha_d = e^{-\varepsilon} \) and adjusting the probability mass to one instead of setting \( \alpha_u = e^\varepsilon \), we have

\[
P(K_{f_1}(D) = 1|f_1(D) = 0) = P(K_{f_1}(D) = 1|f_2(D) = 0) = 0.5\alpha_d = 0.5e^{-1} = 0.1839
\]

\[
P(K_{f_1}(D) = 1|f_1(D) = 1) = P(K_{f_1}(D) = 1|f_2(D) = 1) = 1 - 0.5\alpha_d = 0.8161
\]

Table 5.3 shows the joint distribution for the output of \( (f_1, f_2) \), which is obtained by multiplying the output distributions for \( f_1 \) and \( f_2 \).

For \( \varepsilon \)-differential privacy to hold for the two-component query \( f = (f_1, f_2) \), the ratio of the response distribution at \( D \) and the response distribution at any \( D' \) that results from \( D \) by adding or removing a single individual must be within the range \([e^{-\varepsilon}, e^\varepsilon]\).
Figure 5.4: Prior knowledge about attribute B and distribution of the $\varepsilon$-differentially private response to query functions $f_1$ and $f_2$, assuming that the actual value for attribute B is 0.

Table 5.3: Distribution of the differentially private response to the two-component query $(f_1, f_2)$ when the true values are $f_1(D) = f_2(D) = 0$

<table>
<thead>
<tr>
<th>$K_{f_1}$</th>
<th>$f_1$</th>
<th>$f_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1 − $0.5\alpha_d$</td>
<td>$0.5\alpha_d$</td>
</tr>
<tr>
<td>1</td>
<td>$(1 − 0.5\alpha_d)^2$</td>
<td>$(1 − 0.5\alpha_d)0.5\alpha_d$</td>
</tr>
<tr>
<td></td>
<td>$(1 − 0.5\alpha_d)0.5\alpha_d$</td>
<td>$0.25\alpha_d^2$</td>
</tr>
</tbody>
</table>

As $f_1$ and $f_2$ are related to individuals 1 and 2, any modification to D that does not affect the records for those individuals leaves the distribution of responses unchanged. As we are assuming that individuals 1 and 2 are in D, the only modifications to be considered are the removal of one of these individuals. Table 5.4 shows the distributions of responses when individual 1 or 2 are removed. We use $K_f$ to denote the distribution of the response to query f. It can be seen that the respective ratios between the distribution in Table 5.3 and the ones in Table 5.4 are within $[e^{-\varepsilon}, e^{\varepsilon}] = [e^{-1}, e]$, specifically, the ratios take only two values, $\alpha_d = e^{-1}$ and $2 - \alpha_d = 2 - e^{-1}$.

We now state and prove in general the property illustrated in the previous example.

**Proposition 5.** Let D be a data set and let $(f_1, \ldots, f_n)$ be a set of query functions related to disjoint sets of individuals. Let $K_{f_i}$ be a random variable that provides $\varepsilon_i$-differential privacy for $f_i$, and assume that $K_{f_i}$ is independent from $K_{f_j}$ for any $i \neq j$. Then $(K_{f_1}, \ldots, K_{f_n})$ provides $\max\{\varepsilon_i\}$-differential privacy for $(f_1, \ldots, f_2)$.

**Proof.** Let $D'$ be a data set obtained from D by adding or removing a single user. We want to check that the following inequalities hold for any subset $S$ of the range of $(K_{f_1}, \ldots, K_{f_n})$:

$$e^{-\max\{\varepsilon_i\}} \leq \frac{P((K_{f_1}(D), \ldots, K_{f_n}(D)) \in S)}{P((K_{f_1}(D'), \ldots, K_{f_n}(D')) \in S)} \leq e^{\max\{\varepsilon_i\}}$$
Table 5.4: Distribution of the response to query $f = (f_1, f_2)$ when either individual 1 is missing (top) or individual 2 is missing (bottom), and when the attribute value for the non-missing individual is 0.

<table>
<thead>
<tr>
<th></th>
<th>$K_{f_1}$</th>
<th></th>
<th>$K_{f_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.5(1 - 0.5$\alpha_d$)</td>
<td>0.5(1 - 0.5$\alpha_d$)</td>
<td>0.25$\alpha_d$</td>
</tr>
</tbody>
</table>

It is easy to show that the above inequality holds for the case of $S$ being the Cartesian product of sets $S_i$, with $S_i$ a subset of the range of $K_{f_j}(D)$, or when the probability distribution of $(K_{f_1}, \ldots, K_{f_n})$ is absolutely continuous. For a general set $S$ and a non absolutely continuous distribution, the inequalities still hold. However, such a general proof requires the use of some concepts of measure theory and, for space reasons, we omit such details here. We will show that the inequalities hold for the case of $S = S_1 \times \ldots \times S_n$.

The probabilities $P((K_{f_1}(D), \ldots, K_{f_n}(D)) \in S)$ and $P((K_{f_1}(D'), \ldots, K_{f_n}(D')) \in S)$ can be written as the product of probabilities $\prod P(K_{f_j}(D) \in S_i)$ and $\prod P(K_{f_j}(D') \in S_i)$, respectively. By adding or removing a single individual, only one of the queries is affected. Say the affected query is $f_j$ for some $j \in \{1, \ldots, n\}$. By removing the factors that are both in the numerator and the denominator, the inequalities that we need to check become

$$e^{-\max\{\epsilon_i\} \leq \frac{P(K_{f_j}(D) \in S_j)}{P(K_{f_j}(D') \in S_j)} \leq e^{\max\{\epsilon_i\}},}$$

which holds because $K_{f_j}$ satisfies $\epsilon_j$-differential privacy, and $\epsilon_j \leq \max\{\epsilon_i\}$. ∎

### 5.5 Interactive queries and adaptive attacks

Differential privacy is usually presented as an interactive query-response mechanism where the data set is held by a trusted party to whom users send their queries. Despite this claimed interactivity, the formal definition of differential privacy (Definition 21) is based on a single query, thereby removing the complexities that interactivity would introduce. Malicious users may try to use interaction to exploit potential vulnerabilities of the access mechanism. When using Laplace noise addition the user can, for example, use the knowledge acquired from previous answers to forge the new
Protocol 1 Interactive Laplace noise addition mechanism

1. The database holder initializes the access mechanism with the following parameters:
   • $\varepsilon$, the maximum level of leakage allowed;
   • $\lambda$, the amount of noise to be added to every response ($\lambda$ is the parameter of the Laplace noise distribution); for fixed $\varepsilon$, the greater $\lambda$, the more queries the access mechanism will be able to answer.

2. Let $i := 1$.

3. while queries are answered by the access mechanism do
   (a) The user submits a query $f_i$ (for $i > 1$, $f_i$ may depend on responses to previous queries $(f_1, \cdots, f_{i-1})$).
   (b) if $\Delta(f_1, \cdots, f_i)/\lambda \leq \varepsilon$ then the access mechanism returns $f_i(D) + \text{Laplace}(\lambda)$ as response; else it returns nothing.
   (c) $i := i + 1$

query. For knowledge refinement the problem is even more compelling, since, besides the query function, the user also feeds the access mechanism with a prior knowledge distribution and optionally with a distance function.

5.5.1 Interactive access mechanisms

To implement interactivity, a protocol is built on top of the non-interactive access mechanism. The idea is quite simple; when a query is submitted, the access mechanism analyzes if answering the query is too disclosive, in which case the query is simply discarded. To determine if answering a new query is too disclosive, all the queries submitted by a user so far, including the new query, are treated as a single multicomponent query and $\varepsilon$-differential privacy is enforced for it. Protocol 1 describes the protocol for the interactive Laplace noise access mechanism introduced in [31]; in the protocol, $\Delta(\cdot)$ stands for sensitivity.

We now present an interactive knowledge refinement mechanism parallel to the Laplace-based one. As knowledge refinement does not depend on the sensitivity of the query function, our interactive mechanism does not need to compute sensitivities and is therefore simpler than the Laplace-based one. Also, we will allow the database user to select the amount of leakage $\varepsilon_i$ independently for each query $f_i$. The only requirement is that the access mechanism will refuse answering query $f_i$ (and successive queries) if the leakage of the multicomponent query $(f_1, \cdots, f_i)$ exceeds $\varepsilon$.

If the $d$-th query is the last query answered by the interactive mechanism of Protocol 2, by construction the user obtains at most a knowledge gain $\varepsilon$ for $(f_1, \cdots, f_d)$. This holds regardless of the prior knowledge distributions and distance functions chosen by the user for each query.
Protocol 2 Interactive mechanism for knowledge refinement

1. The database holder initializes the access mechanism with $\epsilon$, the maximum level of leakage allowed.

2. Let $i := 1$.

3. while queries are answered by the access mechanism do
   (a) The user submits a query $q_i = (f_i, P_{f_i}, d_i, \epsilon_i)$, where $f_i$ is the query function, $P_{f_i}$ is the prior knowledge distribution for the query, $d_i$ is the distance function to be used and $\epsilon_i$ is the desired level of leakage (for $i > 1$, $q_i$ may depend on responses to previous queries $(q_1, \ldots, q_{i-1})$).
   (b) if $\sum_{j=1}^{i} \epsilon_j \leq \epsilon$ then the access mechanism returns a response to $f_i$ resulting from applying knowledge refinement to $P_{f_i}$ with distance $d_i$ so that $\epsilon_i$-differential privacy is guaranteed; else it returns nothing.
   (c) $i := i + 1$

By submitting the desired level of leakage $\epsilon_i$ for each query, in Protocol 2 the database user is allowed to trade more accurate answers in some queries for less accurate answers in other queries. Protocol 1 could be modified to permit such flexibility as well: the user could be asked to choose the noise parameter $\lambda_i$ for the $i$-th query, and the condition checked by the access mechanism would become

$$\sum_{j=1}^{i} \Delta(f_j)/\lambda_j \leq \epsilon$$

Since $\Delta(f_1, \ldots, f_i) \leq \Delta(f_1) + \cdots + \Delta(f_i)$, when $\lambda_1 = \cdots = \lambda_i$ the modified condition above may result in less queries being answered than the condition in Protocol 1.

5.5.2 Adaptive attacks

The interactive mechanisms of Protocols 1 and 2 guarantee, respectively for Laplace noise and knowledge refinement, that the responses to any sequence of adaptive queries $(q_1, \ldots, q_d)$ will not violate $\epsilon$-differential privacy. However, the following question can be raised: is there any sequence of adaptive queries $(q_1, \ldots, q_d)$ and a way to combine the responses to this sequence that allows an attacker to obtain an estimator of $f(D)$ that does not satisfy $\epsilon$-differential privacy?

We show that such an attack cannot succeed. Our proof is completely general; it does not depend on the access mechanism used to attain differential privacy. Let $F : \mathbb{R}^d \to \mathbb{R}$ be the function used by the attacker to combine the responses to $q_1, \ldots, q_d$, let these responses be samples of the random vector $K_{f_1}(D), \ldots, K_{f_d}(D)$. The attacker computes $F(K_{f_1}(D), \ldots, K_{f_d}(D))$ and takes it as the response to $f(D)$. We are not interested in determining $F$ or even in determining whether $F(K_{f_1}(D), \ldots, K_{f_d}(D))$ is a good estimate for $f(D)$. The following result will suffice.
Proposition 6. For any function $F$, if $(K_{f_1}(D), \ldots, K_{f_d}(D))$ satisfies $\varepsilon$-differential privacy, then $F(K_{f_1}(D), \ldots, K_{f_d}(D))$ also satisfies $\varepsilon$-differential privacy.

Proof. We need to check that, for each pair of data sets $D$ and $D'$ that differ in a single individual and for each set $S \in \text{Range}(F(K_{f_1}, \ldots, K_{f_d}))$, it holds that

$$P(F(K_{f_1}(D), \ldots, K_{f_d}(D)) \in S) \leq e^\varepsilon$$

Since $P(F \circ X \in S) = P(X \in F^{-1}(S))$, we can express the previous inequality as

$$P((K_{f_1}(D'), \ldots, K_{f_d}(D') \in F^{-1}(S)) \leq e^\varepsilon$$

which holds because $(K_{f_1}(D), \ldots, K_{f_d}(D))$ satisfies $\varepsilon$-differential privacy. \qed

The following corollary follows from the previous proposition.

Corollary 1. Whatever the attacker’s strategy, her estimate for $f(D)$ always satisfies $\varepsilon$-differential privacy.

5.6 Quality of the response to individual queries

We have defined an individual query, $f$, to be one that depends on a single individual. We can think of it as a query that returns the value of some attribute for some specific individual.

Typical differential privacy mechanisms based on noise addition provide low data quality responses for individual queries. The reason is that, as any individual can take any value in $\text{Range}(f)$, the sensitivity of the query equals the length of $\text{Range}(f)$. When using knowledge refinement, the quality of the response depends to a great extent on the prior knowledge available.

In this section, we provide some data quality comparisons between Laplace noise addition and knowledge refinement for individual queries. Comparisons will be based of specific query functions. The first one is based on a query function that returns a Boolean value; we show how the distribution for the differentially private response gets closer to the real response by refining prior knowledge than by adding Laplace noise. The second comparison is based on a continuous function with range $[0, 1]$; we show that, even if we have no prior knowledge, knowledge refinement provides better data quality for individual queries.

5.6.1 Data quality for a Boolean attribute

Consider a simple database $D$ with two attributes: an identifier $ID$ and a Boolean attribute $B$ that may take values 0 and 1. We assume that $B$ is very sensitive and that, to limit the disclosure risk, access to the database must be mediated by a query-response mechanism satisfying differential privacy, with $\varepsilon = 1$. Let $f : D \rightarrow \{0, 1\}$ be a query that asks the value of attribute $B$ for a specific individual.
To achieve differential privacy via Laplace noise addition, we must first compute the sensitivity of function $f$. Assuming that $f$ returns $1/2$ if the individual is not in the database, the $L_1$-sensitivity of $f$ is $1/2$. Therefore, to achieve differential privacy for $\varepsilon = 1$, we must add a Laplace distribution $L(0, 1/2)$ to the true value of the query response. Figure 5.5 shows the distribution of the responses for both possible values of $B$, 0 and 1.

Assuming that the user is only interested in a 0/1 response, any value below $1/2$ is taken as 0, and any value above $1/2$ as 1. The distribution for the response thus obtained is:

$$K_f(D) = \begin{cases} 0 & f(D) + L(0, 1/2) < 0.5 \\ 1 & \text{otherwise} \end{cases}$$

If $f(D)$ equals 0, $K_f(D)$ follows a Bernoulli distribution with parameter 0.184. If $f(D)$ equals 1, the distribution of $K_f(D)$ is a Bernoulli with parameter 0.816. Note that this is completely independent from the true distribution of attribute $B$, and from any previous knowledge that the user might have on it. Hence, differential privacy via Laplace noise addition does not let the user exploit prior knowledge.

Let us assume that attribute $B$ is 1 only with probability 0.01. For a user with this information, using the response obtained from the differential privacy mechanism is actually misleading, as the result will be 1 with probability

$$P(K_f(D) = 1) = P(K_f(D) = 1|f(D) = 0)P(f(D) = 0) + P(K_f(D) = 1|f(D) = 1)P(f(D) = 1)$$

$$= 0.184 \cdot 0.99 + 0.816 \cdot 0.01 = 0.19$$

We could increase the parameter $\varepsilon$ to get a more accurate response. However, by doing so we would be reducing the privacy guarantees.

Now, we turn to the refinement mechanism and, same as before, we assume that the user knows that $B$ equals 1 with probability 0.01. Take $\alpha_u = e^{\varepsilon} = e$ and $\alpha_d = e^{-\varepsilon} = e^{-1}$. Hence,
Figure 5.6: Response distribution with prior knowledge refinement

\[
P(K_f(D) = 1|f(D) = 0) = P(f(D) = 1) \cdot \alpha_d = 0.003678
\]
\[
P(K_f(D) = 0|f(D) = 0) = 1 - 0.003678 = 0.996322
\]
\[
P(K_f(D) = 1|f(D) = 1) = P(f(D) = 1) \cdot \alpha_u = 0.027182
\]
\[
P(K_f(D) = 0|f(D) = 1) = 1 - 0.027182 = 0.972817
\]

Note that, as this is not an absolutely continuous distribution, we had to do some adjustment to have a total probability mass equal to one: instead of adjusting \(\alpha_u\) and \(\alpha_d\), we directly adjusted \(P(K_f(D) = 0|f(D) = 0)\) and \(P(K_f(D) = 0|f(D) = 1)\).

Figure 5.6 depicts the distribution of the response for both possible values of attribute \(B\) and for the prior knowledge.

Now, the probability of obtaining a response 1 is

\[
P(K_f(D) = 1) = P(K_f(D) = 1|f(D) = 0)P(f(D) = 0) + P(K_f(D) = 1|f(D) = 1)P(f(D) = 1)
\]
\[
= 0.003678 \cdot 0.99 + 0.02182 \cdot 0.01 = 0.003912
\]

As 0.003912 is much closer to 0.01 than 0.19, we conclude that, despite both mechanisms providing the same level of privacy, the output distribution is much closer to the actual distribution of the attribute when using the mechanism based on knowledge refinement. Therefore, knowledge refinement outperforms Laplace noise addition for Boolean attributes released under differential privacy.

### 5.6.2 Data quality for a continuous attribute

Let \(f : D \rightarrow [0, 1]\) be a query function that returns a value in the interval \([0, 1]\). We have fixed the range of \(f\) to be able to obtain some numerical results, but a similar comparison can be done for other ranges. We compare the response obtained by using Laplace noise addition and knowledge refinement with a uniform \(U[0, 1]\) prior knowledge.

When using Laplace noise addition, the response to \(f(D)\) is \(K_f(D) = f(D) + Laplace(0, 1/\varepsilon)\). When using knowledge refinement, the prior knowledge is modified by increasing the probability of the set \(U_u\) containing the points closer to \(f(D)\) by
a factor \( \alpha_u \), and decreasing the probability of the rest by a factor \( \alpha_d \). We saw in Section 5.3 that \( U_u \) must satisfy \( P_f(U_u) = \frac{(\alpha_u - 1)}{(\alpha_u - \alpha_d)} \), which in the case of a uniform prior knowledge within the interval \([0, 1]\) coincides with the size of \( U_u \). We also saw (Table 5.2) that, for an individual query, the factors are \( \alpha_u = e^\epsilon \) and \( \alpha_d = e^{-\epsilon} \).

Table 5.5 shows a comparison of the distribution for the response to \( f(D) \) for several values of \( \epsilon \) when \( f(D) = 0.5 \). For Laplace noise addition, we have computed the variance of the response, as well as the probability for the response to be within the range \([0, 1]\). For knowledge refinement, we have computed the variance of the response, the size of \( U_u \), and the probability for the response to be in \( U_u \). The results in the table show that knowledge refinement behaves much better than Laplace noise addition, but perhaps this is better observed by comparing the actual distributions. Figure 5.7 shows the distributions for the response when using Laplace noise addition and knowledge refinement with the same values of \( \epsilon \) used in the table.

Table 5.5: Comparison between the distribution of the response to \( f(D) \) for Laplace noise addition and knowledge refinement for several values of \( \epsilon \) when \( f(D) = 0.5 \)

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>Variance</th>
<th>( P(K_f(D) \in [0, 1]) )</th>
<th>Variance</th>
<th>( \text{size}(U_u) )</th>
<th>( P(K_f(D) \in U_u) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>200</td>
<td>0.476</td>
<td>0.077</td>
<td>0.475</td>
<td>0.525</td>
</tr>
<tr>
<td>( \ln(2) )</td>
<td>4.16</td>
<td>0.549</td>
<td>0.046</td>
<td>0.333</td>
<td>0.667</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.607</td>
<td>0.034</td>
<td>0.269</td>
<td>0.731</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>0.684</td>
<td>0.012</td>
<td>0.119</td>
<td>0.881</td>
</tr>
</tbody>
</table>

5.7 Discussion

In previous sections we have highlighted that the knowledge refinement mechanism lets the database user exploit her prior knowledge to obtain a more accurate response. In Section 5.6 we saw that, for the case of individual queries, knowledge refinement provides a much more accurate response even when there is no prior knowledge.

Other advantages of prior knowledge refinement are:

- **Simplicity.** Mechanisms such as Laplace noise addition are based on the addition of a random noise whose magnitude depends on the variation of the query function across neighbor data sets, also known as sensitivity. To calibrate the random noise, the sensitivity of the function must be computed, which may be quite complex. The mechanism based on the refinement of the prior knowledge only depends on the prior knowledge (it is independent from the sensitivity of the query function), and thus it is easier to implement, especially in a non-supervised environment.

- **Generality.** As said above, Laplace noise addition requires computing the sensitivity of the query function, and this can only be done if the query function takes
values in a metric space. This introduces some complexities when the function returns categorical information. The mechanism based on prior knowledge refinement does not impose any requirement on the query function, and thus it can be applied without extra overhead to functions returning categorical information.

- **Consistency.** Knowledge refinement lets the database user easily restrict the response to a set of values consistent with the query function, by having the prior knowledge assign a probability mass of zero to the set of inconsistent values. For example, in Table 5.5 we saw that Laplace noise sends the response outside the query function range $[0, 1]$ with great probability, while knowledge refinement always keeps the response within range. Querying categorical attributes is another example. It is usual to have some combinations of categories that do not make sense. For example, if the attributes are “employed” (Y/N), and “unemployment benefits” (Y/N), a response Y for both attributes does not make sense. When using a noise addition mechanism, there is no way to avoid that combination of values, while, when using knowledge refinement, to avoid that combination we only have to use a prior knowledge distribution that assigns zero probability mass to it.

Despite the advantages listed above, there are some situations for which the proposed mechanism is not appropriate. If the range of values that the function may return is large compared to the variability between neighbor data sets, and the database user does not have precise knowledge of the response, then a method based on noise addition produces better data quality. This may be the case of statistical queries where the
user has no prior knowledge of the result. However, when querying about a specific individual, the proposed method results in much greater response quality.

5.8 Conclusions

We have introduced a novel mechanism to attain differential privacy. This mechanism is based on refining the prior knowledge that the user may have about the query response. This refinement is performed taking into account the constraints imposed by differential privacy.

The refinement mechanism presents several advantages over the usual noise addition mechanism. It is easier to implement, especially in a non-supervised environment, as it does not require potentially complex computations (such as determining the sensitivity of the query function). The fact that it lets users exploit their prior knowledge may lead to a level of data quality not reachable by mechanisms independent of the user knowledge. For example, we showed in the examples of Section 5.6 that the distribution of the response was closer to the real distribution when using the refinement mechanism. For query functions with great sensitivity, the amount of noise added by noise addition mechanisms, such as [31], may render the response useless. In contrast, the data quality that results from our proposal is independent from the sensitivity of the query function; yet this has the drawback that, for small sensitivities, our approach may be inferior to noise addition.

We have also analyzed the behavior of our approach for multicomponent queries. A generic property of differential privacy guarantees that, if a $\varepsilon_i$-differentially private response is provided for a query $f_i$, for $i = 1$ to $n$, a $\sum \varepsilon_i$-differentially private response is provided for the query $(f_1, \ldots, f_n)$. We have seen that this can be improved if each query $f_i$ refers to a disjoint set of individuals. In this case, we achieve $\max\{\varepsilon_i\}$-differential privacy, instead of $\sum \varepsilon_i$-differential privacy.

Interactive mechanisms for Laplace noise addition and knowledge refinement have also been described. Such interactive mechanisms take as input parameter the maximum level of leakage $\varepsilon$ allowed by the database holder, and queries are answered until that level of leakage is reached. The knowledge refinement interactive mechanism is superior to the Laplace noise interactive mechanism in that it does not need to compute sensitivities. We have shown that any interactive mechanism providing $\varepsilon$-differential privacy is safe against adaptive attacks; whatever the strategy used by an attacker to combine query responses, $\varepsilon$-differential privacy holds.
Chapter 6

A methodology for direct and indirect discrimination prevention

Automated data collection and data mining techniques such as classification rule mining have paved the way to making automated decisions, like loan granting/denial, insurance premium computation, etc. If the training datasets are biased in what regards discriminatory attributes like gender, race, religion, etc., discriminatory decisions may ensue. Discrimination can be either direct or indirect. Direct discrimination occurs when decisions are made based on discriminatory attributes. Indirect discrimination occurs when decisions are made based on non-discriminatory attributes which are strongly correlated with biased sensitive ones. In this chapter, we tackle discrimination prevention in data mining and propose new techniques applicable for direct or indirect discrimination prevention individually or both at the same time. We discuss how to clean training datasets and outsourced datasets in such a way that direct and/or indirect discriminatory decision rules are converted to legitimate (non-discriminatory) classification rules. We also propose new metrics to evaluate the utility of the proposed approaches and we compare these approaches. The experimental evaluations demonstrate that the proposed techniques are effective at removing direct and/or indirect discrimination biases in the original dataset while preserving data quality.

6.1 Contributions

Discrimination prevention methods based on pre-processing published so far [56] present some limitations, which we next highlight:

- They attempt to detect discrimination in the original data only for one discriminatory item and based on a single measure. This approach cannot guarantee that the transformed dataset is really discrimination-free, because it is known
that discriminatory behaviors can often be hidden behind several discriminatory items, and even behind combinations of them.

- They only consider direct discrimination.
- They do not include any measure to evaluate how much discrimination has been removed and how much information loss has been incurred.

In this chapter, we propose pre-processing methods which overcome the above limitations. Our new data transformation methods (i.e. rule protection and rule generalization) are based on measures for both direct and indirect discrimination and can deal with several discriminatory items. Also, we provide utility measures. Hence, our approach to discrimination prevention is broader than in previous work.

In our earlier work [42], we introduced the initial idea of using rule protection and rule generalization for direct discrimination prevention, but we gave no experimental results. In [43], we introduced the use of rule protection in a different way for indirect discrimination prevention and we gave some preliminary experimental results. In this thesis, we present a unified approach to direct and indirect discrimination prevention, with finalized algorithms and all possible data transformation methods based on rule protection and/or rule generalization that could be applied for direct or indirect discrimination prevention. We specify the different features of each method.

As part of this effort, we have developed metrics that specify which records should be changed, how many records should be changed and how those records should be changed during data transformation. In addition, we propose new utility measures to evaluate the different proposed discrimination prevention methods in terms of data quality and discrimination removal for both direct and indirect discrimination. Based on the proposed measures, we present extensive experimental results for two well-known datasets and compare the different possible methods for direct or indirect discrimination prevention to find out which methods could be more successful in terms of low information loss and high discrimination removal.

6.1.1 Basic Definitions

Let \( \mathcal{I} = \{i_1, \ldots, i_n\} \) be a set of items, where each item \( i_j \) has the form \( \text{attribute}=\text{value} \) (e.g., \( \text{Sex=female} \)). An itemset \( X \subseteq \mathcal{I} \) is a collection of one or more items, e.g., \{\text{Sex=female, Credit\_history=no-taken}\}. A database is a collection of data objects (records) and their attributes; more formally, a (transaction) database \( \mathcal{D} = \{r_1, \ldots, r_m\} \) is a set of data records or transactions where each \( r_i \subseteq \mathcal{I} \). Civil rights laws [9, 35] explicitly identify the groups to be protected against discrimination, such as minorities and disadvantaged people, e.g., women. In our context, these groups can be represented as items, e.g., \( \text{Sex=female} \), which we call potentially discriminatory (PD) items; a collection of PD items can be represented as an itemset, e.g., \{\text{Sex=female, Foreign\_worker=yes}\}, which we call PD itemset or protected-by-law (or protected for short) groups, denoted by \( \mathcal{DI}_b \). An itemset \( X \) is PND if \( X \cap \mathcal{DI}_b \neq \emptyset \), e.g., \{\text{credit\_history=no-taken}\} is a PND itemset where \( \mathcal{DI}_b:\{\text{Sex=female}\} \). PD attributes are those that can take PD items as values; for instance, \( \text{Race} \) and \( \text{Gender} \) where \( \mathcal{DI}_b:\{\text{Sex=female, Race=black}\} \). A decision (class) attribute is one taking as values
yes or no to report the outcome of a decision made on an individual; an example is the attribute credit_approved, which can be yes or no. A class item is an item of class attribute, e.g., Credit_approved=no. The support of an itemset \( X \) in a database \( D \) is the number of records that contain \( X \), i.e. \( \text{supp}_D(X) = |\{r_i \in D | X \subseteq r_i\}| \), where \( \cdot \) is the cardinality operator. From patterns, it is possible to derive association rules. An association rule is an expression \( X \rightarrow Y \), where \( X \) and \( Y \) are itemsets. We say that \( X \rightarrow Y \) is a classification rule if \( Y \) is a class item and \( X \) is an itemset containing no class item, e.g. \( \text{Sex=female, Credit\_history=no-taken} \rightarrow \text{Credit\_approved=no} \). The itemset \( X \) is called the premise of the rule. We say that a rule \( X \rightarrow C \) is completely supported by a record if both \( X \) and \( C \) appear in the record. The confidence of a classification rule, \( \text{conf}_D(X \rightarrow C) \), measures how often the class item \( C \) appears in records that contain \( X \). Hence, if \( \text{supp}_D(X) > 0 \) then

\[
\text{conf}_D(X \rightarrow C) = \frac{\text{supp}_D(X, C)}{\text{supp}_D(X)} \tag{6.1}
\]

Confidence ranges over \([0, 1]\). We omit the subscripts in \( \text{supp}_D(\cdot) \) and \( \text{conf}_D(\cdot) \) when there is no ambiguity. A frequent classification rule is a classification rule with support and confidence greater than respective specified lower bounds. The negated itemset, i.e. \( \neg X \) is an itemset with the same attributes as \( X \), but the attributes in \( \neg X \) take any value except those taken by attributes in \( X \). In this chapter, we use the \( \neg \) notation for itemsets with binary or non-binary categorical attributes. For a binary attribute, e.g. \{Foreign worker=Yes/No\}, if \( X \) is \{Foreign worker=Yes\}, then \( \neg X \) is \{Foreign worker=No\}. If \( X \) is binary, it can be converted to \( \neg X \) and vice versa, that is, the negation works in both senses. In the previous example, we can select the records in \( DB \) so that the value of the Foreign worker attribute is “Yes” and change that attribute’s value to “No”, and conversely. However, for a non-binary categorical attribute, e.g. \{Race=Black/White/Indian\}, if \( X \) is \{Race=Black\}, then \( \neg X \) is \{Race=White\} or \{Race=Indian\}. In this case, \( \neg X \) can be converted to \( X \) without ambiguity, but the conversion of \( X \) into \( \neg X \) is not uniquely defined. In the previous example, we can select the records in \( DB \) such that the Race attribute is “White” or “Indian” and change that attribute’s value to “Black”; but if we want to negate \{Race=Black\}, we do not know whether to change it to \{Race=White\} or \{Race=Indian\}. In this thesis, we use only non-ambiguous negations.

### 6.1.2 Measures of Discrimination

The legal principle of under-representation has inspired existing approaches for discrimination discovery based on rule/pattern mining.

Given \( DI_b \) and starting from a dataset \( D \) of historical decision records, the authors of [77] propose to extract frequent classification rules of the form \( A, B \rightarrow C \), called PD rules, to unveil contexts \( B \) of possible discrimination, where the non-empty protected group \( A \subseteq DI_b \) suffers from over-representation with respect to the negative decision \( C \) (\( C \) is a class item reporting a negative decision, such as credit denial, application rejection, job firing, and so on). In other words, \( A \) is under-represented w.r.t. the corresponding positive decision \( \neg C \). As an example, rule \( \text{Sex=female, Job=veterinarian} \)
→ Credit-approved=no is a PD rule about denying credit (the decision C) to women (the protected group A) among those who are veterinarians (the context B), where DI_{v;[Sex=female]}. And a classification rule of the form X → C is called PND rule if X is a PND itemset. As an example, rule Credit_history=paid-delay, Job=veterinarian → Credit_approved=no is a PND rule, where DI_{v;[Sex=female]}. Then, the degree of under-representation should be measured over each PD rule by one of the legally grounded measures introduced in Pedreschi et al. [78].

**Definition 24.** Let A, B → C be a PD classification rule extracted from D with conf(¬A, B → C) > 0. The selection lift (slift) of the rule is

\[ \text{slift}(A, B \rightarrow C) = \frac{\text{conf}(A, B \rightarrow C)}{\text{conf}(\neg A, B \rightarrow C)} \]  

(6.2)

In fact, slift is the ratio of the proportions of benefit denial, e.g., credit denial, between the protected and unprotected groups, e.g. women and men resp., in the given context, e.g. new applicants. A special case of slift occurs when we deal with non-binary attributes, for instance, when comparing the credit denial ratio of blacks with the ratio for other groups of the population. This yields a third measure called contrasted lift (clift) which, given A as a single item \( a = v_1 \) (e.g. black race), compares it with the most favored item \( a = v_2 \) (e.g. white race).

**Definition 25.** Let \( a = v_1, B \rightarrow C \) be a PD classification rule extracted from D, and \( v_2 \in \text{dom}(a) \) with \( \text{conf}(a = v_2, B \rightarrow C) \) minimal and non-zero. The contrasted lift (clift) of the rule is

\[ \text{clift}(a = v_1, B \rightarrow C) = \frac{\text{conf}(a = v_1, B \rightarrow C)}{\text{conf}(a = v_2, B \rightarrow C)} \]  

(6.3)

**Definition 26.** Let A, B → C be a PD classification rule extracted from D with \( \text{conf}(B \rightarrow C) > 0 \). The extended lift (elift) of the rule is

\[ \text{elift}(A, B \rightarrow C) = \frac{\text{conf}(A, B \rightarrow C)}{\text{conf}(B \rightarrow C)} \]  

(6.4)

In fact, elift is the ratio of the proportions of benefit denial, e.g. credit denial, between the protected groups and all people who were not granted the benefit in the given context, e.g. women versus all men and women who were denied credit, in the given context, e.g. those who live in NYC.

The last ratio measure is the odds lift (olift), the ratio between the odds of the proportions of benefit denial between the protected and unprotected groups.

**Definition 27.** Let A, B → C be a PD classification rule extracted from D with \( \text{conf}(\neg A, B \rightarrow C) > 0 \) and \( \text{conf}(A, B \rightarrow C) < 1 \). The odds lift (olift) of the rule is

\[ \text{olift}(A, B \rightarrow C) = \frac{\text{odds}(A, B \rightarrow C)}{\text{odds}(\neg A, B \rightarrow C)} \]  

(6.5)

---

1Discrimination on the basis of an attribute happens if a person with an attribute is treated less favorably than a person without the attribute.

2Discrimination occurs when a higher proportion of people not in the group is able to comply.
where
\[
\text{odds}(A, B \to C) = \frac{\text{conf}(A, B \to C)}{\text{conf}(A, B \to \neg C)} \tag{6.6}
\]

Although the measures introduced so far are defined in terms of ratios, measures based on the difference of confidences have been considered on the legal side as well.

**Definition 28.** Let \( A, B \to C \) be a PD classification rule extracted from \( D \). The difference measures are defined as
\[
\text{slift}_d(A, B \to C) = \text{conf}(A, B \to C) - \text{conf}(\neg A, B \to C) \tag{6.7}
\]
\[
\text{elift}_d(A, B \to C) = \text{conf}(A, B \to C) - \text{conf}(B \to C) \tag{6.8}
\]

Difference-based measures range over \([-1, 1]\). Lastly, the following measures are also defined in terms of ratios and known as chance measures.

**Definition 29.** Let \( A, B \to C \) be a PD classification rule extracted from \( D \). The chance measures are defined as
\[
\text{slift}_c(A, B \to C) = \frac{1 - \text{conf}(A, B \to C)}{1 - \text{conf}(\neg A, B \to C)} \tag{6.9}
\]
\[
\text{elift}_c(A, B \to C) = \frac{1 - \text{conf}(A, B \to C)}{1 - \text{conf}(B \to C)} \tag{6.10}
\]

For \( \text{slift}, \text{elift} \) and \( \text{olift} \), the values of interest (potentially indicating discrimination) are those greater than 1; for \( \text{slift}_d \) and \( \text{elift}_d \), they are those greater than 0; and for \( \text{slift}_c \) and \( \text{elift}_c \), they are those less than 1. On the legal side, different measures are adopted worldwide. For example, UK law mentions mostly \( \text{slift}_d \). The EU court of justice has made more emphasis in \( \text{slift} \), and US laws courts mainly refer to \( \text{slift}_c \).

**Classification rule: \( c = A, B \to C \)**

<table>
<thead>
<tr>
<th></th>
<th>( A )</th>
<th>( \neg A )</th>
<th>( C )</th>
<th>( \neg C )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( a_1 )</td>
<td>( a_2 )</td>
<td>( n_1 - a_1 )</td>
<td>( n_1 )</td>
</tr>
<tr>
<td></td>
<td>( n_2 - a_2 )</td>
<td>( n_2 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( p_1 = a_1/n_1 \), \( p_2 = a_2/n_2 \), \( p = (a_1 + a_2)/(n_1 + n_2) \)

\[
\text{elift}(c) = \frac{p_1}{p}, \quad \text{slift}(c) = \frac{p_1}{p_2}, \quad \text{olift}(c) = \frac{p_1(1 - p_2)}{p_2(1 - p_1)}
\]
\[
\text{elift}_d(c) = p_1 - p, \quad \text{slift}_d(c) = p_1 - p_2
\]
\[
\text{elift}_c(c) = \frac{1 - p_1}{1 - p}, \quad \text{slift}_c(c) = \frac{1 - p_1}{1 - p_2}
\]

**Figure 6.1: Discrimination measures**
An alternative view of the measures introduced so far can be given starting from the contingency table of \( c : A, B \rightarrow C \) shown in Fig. 6.1. Each cell in the table is filled in with the number of records in the data table \( D \) satisfying \( B \) and the coordinates (i.e., their absolute support). Using the notation of the figure, confidence of \( c : A, B \rightarrow C \) is \( p_1 = a_1/n_1 \). Similarly, other measures can be defined as shown in Fig. 6.1. Confidence intervals and tests of statistical significant of the above measures are discussed in [78]. Here, we only mention that statistical tests will rank the rules according to how unlikely it is that they would be observed if there was equal treatment, not according to the severity of discrimination. The rankings imposed by the discrimination measures in Fig. 6.1 are investigated by Pedreschi et al. [80]: the choice of the reference measure critically affects the rankings of PD rules, with the \( slift_c \) and the \( slift \) measures exhibiting the largest differences.

### 6.2 Direct and Indirect Discrimination Measurement

Let \( FR \) be the database of frequent classification rules extracted from \( D \). Whether a PD rule in \( FR \) has to be considered discriminatory or not can be assessed by thresholding one of the measures in Fig. 6.1.

**Definition 30.** Let \( f \) be one of the measures in Fig. 6.1. Given protected groups \( DI_b \) and \( \alpha \in \mathbb{R} \), a fixed threshold\(^3\), a PD classification rule \( r : A, B \rightarrow C \), where \( C \) denies some benefit and \( A \subseteq DI_b \), is \( \alpha \)-protective w.r.t. \( f \) if \( f(r) < \alpha \). Otherwise, \( c \) is \( \alpha \)-discriminatory.

The purpose of direct discrimination discovery is to identify \( \alpha \)-discriminatory rules. In fact, \( \alpha \)-discriminatory rules indicate biased rules that are directly inferred from discriminatory items (e.g. Foreign worker = Yes). We call these rules direct \( \alpha \)-discriminatory rules.

The purpose of indirect discrimination discovery is to identify redlining rules. In fact, redlining rules indicate biased rules that are indirectly inferred from non-discriminatory items (e.g. Zip = 10451) because of their correlation with discriminatory ones. To determine the redlining rules, Pedreschi et al. in [77] stated the theorem below which gives a lower bound for \( \alpha \)-discrimination of PD classification rules, given information available in PND rules (\( \gamma, \delta \)) and information available from background rules (\( \beta_1, \beta_2 \)). They assume that background knowledge takes the form of association rules relating a PND itemset \( D \) to a PD itemset \( A \) within the context \( B \).

**Theorem 3.** Let \( r : D, B \rightarrow C \) be a PND classification rule, and let

\[
\gamma = \text{conf}(r : D, B \rightarrow C) \quad \delta = \text{conf}(B \rightarrow C) > 0.
\]

Let \( A \) be a PD itemset, and let \( \beta_1, \beta_2 \) such that

\[
\text{conf}(r_{b1} : A, B \rightarrow D) \geq \beta_1
\]

\(^3\)\( \alpha \) states an acceptable level of discrimination according to laws and regulations. For example, the U.S. Equal Pay Act [101] states that "a selection rate for any race, sex, or ethnic group which is less than four-fifths of the rate for the group with the highest rate will generally be regarded as evidence of adverse impact". This amounts to using \( elift \) with \( \alpha = 1.25 \).
\[ \text{conf}(\text{rb}_2 : D, B \rightarrow A) \geq \beta_2 > 0. \]

**Call**

\[ f(x) = \frac{\beta_1}{\beta_2} (\beta_2 + x - 1) \]

\[ \text{elb}(x, y) = \begin{cases} 
    f(x)/y & \text{if } f(x) > 0 \\
    0 & \text{otherwise}
\end{cases} \]

It holds that, for \( \alpha \geq 0 \), if \( \text{elb}(\gamma, \delta) \geq \alpha \), the PD classification rule \( r' : A, B \rightarrow C \) is \( \alpha \)-discriminatory.

Based on the above theorem, the following formal definitions of redlining and non-redlining rules are presented:

**Definition 31.** A PND classification rule \( r : D, B \rightarrow C \) is a redlining rule if it could yield an \( \alpha \)-discriminatory rule \( r' : A, B \rightarrow C \) in combination with currently available background knowledge rules of the form \( \text{rb}_1 : A, B \rightarrow D \) and \( \text{rb}_2 : D, B \rightarrow A \), where \( A \) is a PD itemset. For example \{Zip=10451, City=NYC\} \( \rightarrow \) Hire=No.

**Definition 32.** A PND classification rule \( r : D, B \rightarrow C \) is a non-redlining or legitimate rule if it cannot yield any \( \alpha \)-discriminatory rule \( r' : A, B \rightarrow C \) in combination with currently available background knowledge rules of the form \( \text{rb}_1 : A, B \rightarrow D \) and \( \text{rb}_2 : D, B \rightarrow A \), where \( A \) is a PD itemset. For example \{Experience=Low, City=NYC\} \( \rightarrow \) Hire=No.

We call \( \alpha \)-discriminatory rules that ensue from redlining rules **indirect \( \alpha \)-discriminatory rules**.

### 6.3 The Approach

In this section, we present our approach, including the data transformation methods that can be used for direct and/or indirect discrimination prevention. For each method, its algorithm and its computational cost are specified. Our approach for direct and indirect discrimination prevention can be described in terms of two phases:

- **Discrimination Measurement.**
  
  Direct and indirect discrimination discovery includes identifying \( \alpha \)-discriminatory rules and redlining rules. To this end, first, based on predetermined discriminatory items in \( D \), frequent classification rules in \( FR \) are divided in two groups: PD and PND rules. Second, direct discrimination is measured by identifying \( \alpha \)-discriminatory rules among the PD rules using a direct discrimination measure (e.g., \( \text{elift} \)) and a discriminatory threshold \( (\alpha) \). Third, indirect discrimination is measured by identifying redlining rules among the PND rules combined with background knowledge, using an indirect discriminatory measure (\( \text{elb} \)) and a discriminatory threshold \( (\alpha) \). Let \( MR \) be the database of direct \( \alpha \)-discriminatory rules obtained with the above process. In addition, let \( RR \) be the database of redlining rules and their respective indirect \( \alpha \)-discriminatory rules obtained with the above process.
3

Fig. 13.1. The process of extracting biased and unbiased decision rules.

Despite the wide deployment of information systems based on data mining technology in decision making, the issue of anti-discrimination in data mining did not receive much attention until 2008 (Pedreschi et al. 2008). After that, some proposals have addressed the discovery and measure of discrimination. Others deal with the prevention of discrimination. The discovery of discriminatory decisions was first proposed by Pedreschi et al. (2008) and Ruggieri et al. (2010). The approach is based on mining classification rules (the inductive part) and reasoning on them (the deductive part) on the basis of quantitative measures of discrimination that formalize legal definitions of discrimination. For instance, the U.S. Equal Pay Act (United States Congress 1963) states that: “a selection rate for any race, sex, or ethnic group which is less than four-fifths of the rate for the group with the highest rate will generally be regarded as evidence of adverse impact.”

Discrimination can be either direct or indirect (also called systematic, see Pedreschi et al. (2008)). Direct discriminatory rules indicate biased rules that are directly inferred from discriminatory items (e.g., Foreign worker = Yes). Indirect discriminatory rules (redlining rules) indicate biased rules that are indirectly inferred from non-discriminatory items (e.g., Zip = 10451) because of their correlation with discriminatory ones. Indirect discrimination could happen because of the availability of some background knowledge (rules), for example, indicating that a certain zipcode corresponds to a deteriorating area or an area with a mostly black population. The background knowledge might be accessible from publicly available data (e.g., census data) or might be obtained from the original dataset itself because of the existence of non-discriminatory attributes that are highly correlated with the sensitive ones in the original dataset.

### Data Transformation

Figure 6.2: The process of extracting biased and unbiased decision rules

- **Data Transformation.** Transform the original data $D$ in such a way to remove direct and/or indirect discriminatory biases, with minimum impact on the data and on legitimate decision rules, so that no unfair decision rule can be mined from the transformed data. In the following subsections, we present the data transformation methods that can be used for this purpose.

Figure 6.2 illustrates that if the original biased dataset $D$ goes through an anti-discrimination process including discrimination measurement and data transformation, the rules extracted from transformed dataset $D'$ could lead to automated unfair decisions.

As mentioned before, background knowledge might be obtained from the original dataset itself because of the existence of PND attributes that are highly correlated with the PD ones in the original dataset. Let $BK$ be a database of background rules that is defined as

$$BK = \{r_{b2}: D, B \rightarrow A | A \text{ discriminatory itemset and } \text{supp}(D, B \rightarrow A) \geq ms\}$$

In fact, $BK$ is the set of classification rules $D, B \rightarrow A$ with a given minimum support $ms$ that shows the correlation between the PD itemset $A$ and the PND itemset $D$ with context $B$. Although rules of the form $r_{b1}: A, B \rightarrow D$ (in Theorem 3) are not included in $BK$, $\text{conf}(r_{b1}: A, B \rightarrow D)$ could be obtained as $\text{supp}(r_{b2}: D, B \rightarrow A)/\text{supp}(B \rightarrow A)$.

#### 6.4 Data Transformation for Direct Discrimination

The proposed solution to prevent direct discrimination is based on the fact that the dataset of decision rules would be free of direct discrimination if it only contained
PD rules that are $\alpha$-protective or are instances of at least one non-redlining PND rule. Therefore, a suitable data transformation with minimum information loss should be applied in such a way that each $\alpha$-discriminatory rule either becomes $\alpha$-protective or an instance of a non-redlining PND rule. We call the first procedure direct rule protection and the second one rule generalization.

6.4.1 Direct Rule Protection (DRP)

In order to convert each $\alpha$-discriminatory rule into an $\alpha$-protective rule, based on the direct discriminatory measure $elift$ (see Figure 6.1), we should enforce the following inequality for each $\alpha$-discriminatory rule $r': A, B \rightarrow C$ in $\mathcal{MR}$, where $A$ is a PD itemset:

\[
elift(r') < \alpha \tag{6.11}\]

By using the statement of the $elift$ Definition, Inequality (6.11) can be rewritten as

\[
\frac{\text{conf}(r': A, B \rightarrow C)}{\text{conf}(B \rightarrow C)} < \alpha \tag{6.12}\]

Let us rewrite Inequality (6.12) in the following way

\[
\text{conf}(r': A, B \rightarrow C) < \alpha \cdot \text{conf}(B \rightarrow C) \tag{6.13}\]

So, it is clear that Inequality (6.11) can be satisfied by decreasing the confidence of the $\alpha$-discriminatory rule $r': A, B \rightarrow C$ to a value less than the right-hand side of Inequality (6.13), without affecting the confidence of its base rule $B \rightarrow C$. A possible solution for decreasing

\[
\text{conf}(r': A, B \rightarrow C) = \frac{\text{supp}(A, B, C)}{\text{supp}(A, B)} \tag{6.14}\]

is to perturb the discriminatory itemset from $\neg A$ to $A$ in the subset $D_c$ of all records of the original dataset which completely support the rule $\neg A, B \rightarrow \neg C$ and have minimum impact on other rules; doing so increases the denominator of Expression (6.14) while keeping the numerator and $\text{conf}(B \rightarrow C)$ unaltered.

There is also another way to provide direct rule protection. Let us rewrite Inequality (6.12) in the following different way

\[
\text{conf}(B \rightarrow C) > \frac{\text{conf}(r': A, B \rightarrow C)}{\alpha} \tag{6.15}\]

It is clear that Inequality (6.11) can be satisfied by increasing the confidence of the base rule $(B \rightarrow C)$ of the $\alpha$-discriminatory rule $r': A, B \rightarrow C$ to a value higher than the right-hand side of Inequality (6.15), without affecting the value of $\text{conf}(r': A, B \rightarrow C)$. A possible solution for increasing Expression

\[
\text{conf}(B \rightarrow C) = \frac{\text{supp}(B, C)}{\text{supp}(B)} \tag{6.16}\]

is to perturb the class item from $\neg C$ to $C$ in the subset $DB_c$ of all records of the original dataset which completely support the rule $\neg A, B \rightarrow \neg C$ and have minimum impact on
Table 6.1: Data transformation methods (DTMs) for different measures

<table>
<thead>
<tr>
<th>Measures</th>
<th>DTMs for DRP</th>
<th>Transformation requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>elift</td>
<td>¬A, B → ¬C ⇒ A, B → ¬C</td>
<td>elift(A, B → C) &lt; α</td>
</tr>
<tr>
<td></td>
<td>¬A, B → ¬C ⇒ ¬A, B → C</td>
<td></td>
</tr>
<tr>
<td>slift</td>
<td>A, B → C ⇒ A, B → ¬C</td>
<td>slift(A, B → C) &lt; α</td>
</tr>
<tr>
<td></td>
<td>¬A, B → ¬C ⇒ ¬A, B → C</td>
<td></td>
</tr>
<tr>
<td>olift</td>
<td>¬A, B → ¬C ⇒ A, B → ¬C</td>
<td>olift(A, B → C) &lt; α</td>
</tr>
<tr>
<td></td>
<td>¬A, B → ¬C ⇒ ¬A, B → C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A, B → C ⇒ A, B → ¬C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A, B → C ⇒ ¬A, B → C</td>
<td></td>
</tr>
</tbody>
</table>

other rules; doing so increases the numerator of Expression (6.16) while keeping the denominator and \( \text{conf}(r': A, B \rightarrow C) \) unaltered.

Therefore, there are two methods that could be applied for direct rule protection. One method (Method 1) changes the PD itemset in some records (e.g., gender changed from male to female in the records with granted credits) and the other method (Method 2) changes the class item in some records (e.g., from grant credit to deny credit in the records with male gender). Thus, a suitable data transformation with minimum information loss should be applied in such a way that each \( \alpha \)-discriminatory rule \( r \) becomes \( \alpha \)-protective by ensuring that \( f(r) < \alpha \), where \( f(r) \) could be one of the measures in Fig. 6.1. Then the general idea of our proposed method DRP for all these measures is the same. However, in the details (i.e., which records should be changed, how many records should be changed and how those records should be changed during data transformation), there could be some differences because of the different definition of these measures. For instance, in Table 6.1 we present all the possible data transformation methods for measures elift, slift and olift based on our proposed direct rule protection method.

As shown in Table 6.1, two possible data transformation methods for elift are the same as two possible ones for olift and one of them is the same as the possible ones for slift. Another point is that there is a method (the second one for each measure) that is the same for all three measures. Then we can conclude that for DRP not only the general idea is the same for different measures, but also in the details, there is a data transformation method applicable for all the measures.

### 6.4.2 Rule Generalization (RG)

Rule generalization is another data transformation method for direct discrimination prevention. It is based on the fact that if each \( \alpha \)-discriminatory rule \( r': A, B \rightarrow C \) in the database of decision rules was an instance of at least one non-redlining (legitimate) PND rule \( r : D, B \rightarrow C \), the dataset would be free of direct discrimination.

In rule generalization, we consider the relation between rules instead of discrimination measures. The following example illustrates this principle. Assume that a complainant claims discrimination against foreign workers among applicants for a job position. A classification rule \{Foreign worker=Yes, City=NYC\} → Hire=No with
high else supports the complainant’s claim. However, the decision maker could argue that this rule is an instance of a more general rule \{Experience=Low, City=NYC\} \rightarrow Hire=No. In other words, foreign workers are rejected because of their low experience, not just because they are foreign. The general rule rejecting low-experienced applicants is a legitimate one, because experience can be considered a genuine/legitimate requirement for some jobs. To formalize this dependency among rules (i.e. \( r' \) is an instance of \( r \)), Pedreschi et al. in [79] say that a PD classification rule \( r' : A, B \rightarrow C \) is an instance of a PND rule \( r : D, B \rightarrow C \) if rule \( r \) holds with the same or higher confidence, namely \( \text{conf}(r : D, B \rightarrow C) \geq \text{conf}(r' : A, B \rightarrow C) \), and a case (record) satisfying discriminatory itemset \( A \) in context \( B \) satisfies legitimate itemset \( D \) as well, namely \( \text{conf}(A, B \rightarrow D) = 1 \). The two conditions can be relaxed as follows:

**Definition 33.** Let \( d \in [0, 1] \). A classification rule \( r' : A, B \rightarrow C \) is a \( d \)-instance of \( r : D, B \rightarrow C \) if both conditions below are true:

- **Condition 1:** \( \text{conf}(r) \geq d \cdot \text{conf}(r') \)
- **Condition 2:** \( \text{conf}(r'' : A, B \rightarrow D) \geq d \).

Then, if \( r' \) is a \( d \)-instance of \( r \) (where \( d \) is 1 or a value near 1), \( r' \) is free of direct discrimination. Based on this concept, we propose a data transformation method (i.e. rule generalization) to transform each \( \alpha \)-discriminatory \( r' \) in \( MR \) into a \( d \)-instance of a legitimate rule. An important issue to perform rule generalization is to find a suitable PND rule \( r : D, B \rightarrow C \) or, equivalently, to find a suitable \( D \) (e.g. Experience=Low). Although choosing non-redlining rules, as done in this chapter, is a way to obtain legitimate PND rules, sometimes it is not enough and a semantic hierarchy is needed to find the most suitable legitimate itemset.

At any rate, rule generalization can be attempted for \( \alpha \)-discriminatory rules \( r' \) for which there is at least one non-redlining PND rule \( r \) satisfying at least one of the two conditions of Definition 33. If any of the two conditions does not hold, the original data should be transformed for it to hold. Let us assume that Condition (2) is satisfied but Condition (1) is not. Based on the definition of \( d \)-instance, to satisfy the first condition of Definition 33, we should enforce for each \( \alpha \)-discriminatory rule \( r' : A, B \rightarrow C \) in \( MR \) the following inequality, with respect to its PND rule \( r : D, B \rightarrow C \):

\[
\text{conf}(r : D, B \rightarrow C) \geq d \cdot \text{conf}(r' : A, B \rightarrow C)
\]  
(6.17)

Let us rewrite Inequality (6.17) in the following way

\[
\text{conf}(r' : A, B \rightarrow C) \leq \frac{\text{conf}(r : D, B \rightarrow C)}{d}
\]  
(6.18)

So, it is clear that Inequality (6.17) can be satisfied by decreasing the confidence of the \( \alpha \)-discriminatory rule \( r' : A, B \rightarrow C \) to values less than the right-hand side of Inequality (6.18), without affecting the confidence of rule \( r : D, B \rightarrow C \) or the satisfaction of Condition (2) of Definition 33. The confidence of \( r' \) was previously specified in Expression (6.14). A possible solution to decrease this confidence is to perturb the class item from \( C \) to \( \neg C \) in the subset \( D_c \) of all records in the original dataset which completely support the rule \( A, B, \neg D \rightarrow C \) and have minimum impact.
on other rules; doing so decreases the numerator of Expression (6.14) while keeping its denominator, \( \text{conf}(r : D, B \rightarrow C) \) and also \( \text{conf}(r' : A, B \rightarrow D) \) (Condition (2) for rule generalization) unaltered.

Let us see what happens if if Condition (1) of Definition 33 is satisfied but Condition (2) is not. In this case, based on the definition of \( d \)-instance, to satisfy Condition (2) we should enforce the following inequality for each \( \alpha \)-discriminatory rule \( r' : A, B \rightarrow C \) in \( MR \) with respect to its PND rule \( r : D, B \rightarrow C \):

\[
\text{conf}(r'' : A, B \rightarrow D) \geq d \tag{6.19}
\]

Inequality (6.19) must be satisfied by increasing the confidence of rule \( r'' : A, B \rightarrow D \) to a value higher than \( d \), without affecting the satisfaction of Condition (1). However, any effort at increasing the confidence of \( r'' \) impacts on the confidence of the \( r \) or \( r' \) rules and might threaten the satisfaction of Condition (1) of Definition 33; indeed, in order to increase the confidence of \( r'' \) we must either decrease \( \text{supp}(A, B) \) (which increases \( \text{conf}(r') \)) or change \( \sim D \) to \( D \) for those records satisfying \( A \) and \( B \) (which decreases \( \text{conf}(r) \)). Hence, rule generalization can only be applied if Condition (2) is satisfied without any data transformation.

To recap, we see that rule generalization can be achieved provided that Condition (2) is satisfied, because Condition (1) can be reached by changing the class item in some records (e.g. from “Hire no” to “Hire yes” in the records of foreign and high-experienced people in NYC city).

### 6.4.3 Direct Rule Protection and Rule Generalization

Since rule generalization might not be applicable for all \( \alpha \)-discriminatory rules in \( MR \), rule generalization cannot be used alone for direct discrimination prevention and must be combined with direct rule protection. When applying both rule generalization and direct rule protection, \( \alpha \)-discriminatory rules are divided into two groups:

- \( \alpha \)-discriminatory rules \( r' \) for which there is at least one non-redlining PND rule \( r \) satisfying Condition (2) of Definition 33. For these rules, rule generalization is performed unless direct rule protection requires less data transformation (in which case direct rule protection is used).
- \( \alpha \)-discriminatory rules such that there is no such PND rule. For these rules, direct rule protection is performed.

We propose Algorithm 6 to select the most appropriate discrimination prevention approach for each \( \alpha \)-discriminatory rule. First, for each \( \alpha \)-discriminatory rule in \( MR \) of type \( r' : A, B \rightarrow C \), a collection \( D_{pn} \) of non-redlining PND rules of type \( r : D, B \rightarrow C \) is found (Step 2). Then, the conditions of Definition 33 are checked for each rule in \( D_{pn} \), for \( d \geq 0.8 \) (Steps 4-18). Three cases arise depending on whether Conditions (1) and (2) hold:

- **Case 1:** There is at least one rule \( r \in D_{pn} \) such that both Conditions (1) and (2) of Definition 33 hold. In this case \( r' \) is a \( d \)-instance of \( r \) for \( d \geq 0.8 \) and no transformation is required (Steps 19-20).
• Case 2: There is no rule in \( D_{pn} \) satisfying both Conditions (1) and (2) of Definition 33, but there is at least one rule satisfying Condition (2). In this case (Step 23), the PND rule \( r_b \) in \( D_{pn} \), should be selected (Step 24) which requires the minimum data transformation to fulfill Condition (1). A smaller difference between the values of the two sides of Condition (1) for each \( r \) in \( D_{pn} \) indicates a smaller required data transformation. In this case the \( \alpha \)-discriminatory rule is transformed by rule generalization (Step 25).

• Case 3: No rule in \( D_{pn} \) satisfies Condition (2) of Definition 33. In this case (Step 21), rule generalization is not possible and direct rule protection should be performed (Step 22).

Algorithm 6 DETERMINING DISCRIMINATION PREVENTION APPROACHES

\begin{verbatim}
Input: DB, MR, FR, d \geq 0, \alpha
1: for each \( r' : A, B \rightarrow C \in MR \) do
2: \( D_{pn} \leftarrow \) Collection of non-redlining PND rules \( r : D, B \rightarrow C \) from FR
3: if \(|D_{pn}| \neq 0\) then
4: for each \( r \in D_{pn} \) do
5: // Assess conditions of p-instance
6: Compute \( \text{conf}(r') \), where \( r' : A, B \in D \)
7: if \( \text{conf}(r) \geq d \cdot \text{conf}(r') \) then
8: \( C_1 \leftarrow \) true
9: else
10: \( C_1 \leftarrow \) false
11: \( \text{dif}_1 \leftarrow d \cdot \text{conf}(r') - \text{conf}(r) \)
12: end if
13: if \( \text{conf}(r') \geq d \) then
14: \( C_2 \leftarrow \) true
15: else
16: \( C_2 \leftarrow \) false
17: end if
18: end for
19: if \( \exists r \in D_{pn} \) s.t. \( C_1 = \) true \& \( C_2 = \) true then
20: \( TR_{r'} \leftarrow \) Not \# No transformation needed
21: else if for all \( r \in D_{pn} \), \( C_2 = \) false then
22: \( TR_{r'} \leftarrow \) DRP \# Direct Rule Protection
23: else if \( \exists r \in D_{pn} \) s.t. \( C_2 = \) true then
24: \( r_b \leftarrow r \in D_{pn} \) with minimum \( \text{dif}_1 \)
25: \( TR_{r'} \leftarrow \) RG \# Rule Generalization
26: end if
27: else
28: // |D_{pn}| = 0
29: \( TR_{r'} \leftarrow \) DRP \# Direct Rule Protection
30: end if
31: if \( TR_{r'} = \) RG then
32: \( \text{dif}'_1 \leftarrow \text{conf}(r') - \alpha \cdot \text{conf}(B \rightarrow C) \)
33: if \( \text{dif}'_1 < \text{dif}_1 \) then
34: \( TR_{r'} \leftarrow \) DRP
35: end if
36: end if
37: end for
Output: \( TR \) containing all \( r' \in MR \) and their respective \( TR_{r'} \) and \( r_b \)
\end{verbatim}

For the \( \alpha \)-discriminatory rules to which rule generalization can be applied, it is possible that direct rule protection can be achieved with a smaller data transformation. For these rules the algorithm should select the approach with minimum transformation (Steps 31-36). The algorithm yields as output a database \( TR \) with all \( r' \in MR \), their respective rule \( r_b \) and their respective discrimination prevention approaches (\( TR_{r'} \)).
6.5 Data Transformation for Indirect Discrimination

The proposed solution to prevent indirect discrimination is based on the fact that the dataset of decision rules would be free of indirect discrimination if it contained no redlining rules. To achieve this, a suitable data transformation with minimum information loss should be applied in such a way that redlining rules are converted to non-redlining rules. We call this procedure indirect rule protection.

6.5.1 Indirect Rule Protection (IRP)

In order to turn a redlining rule into a non-redlining rule, based on the indirect discriminatory measure (i.e. \( \text{elb} \) in Theorem 3), we should enforce the following inequality for each redlining rule \( r \):

\[
\frac{\text{conf}(r_{b_1})}{\text{conf}(r_{b_2})} \left( \text{conf}(r_{b_2}) + \text{conf}(r : D, B \rightarrow C) - 1 \right) < \alpha \tag{6.20}
\]

By using the definitions stated when introducing \( \text{elb} \) in Theorem 3, Inequality (6.20) can be rewritten as

\[
\frac{\text{conf}(r_{b_1} : A, B \rightarrow D)}{\text{conf}(r_{b_2})} < \alpha \cdot \frac{\text{conf}(B \rightarrow C) \cdot \text{conf}(r_{b_2})}{\text{conf}(r_{b_2}) + \text{conf}(r : D, B \rightarrow C) - 1} \tag{6.21}
\]

Note that the discriminatory itemset (i.e. \( A \)) is not removed from the original database \( D \) and the rules \( r_{b_1} : A, B \rightarrow D \) and \( r_{b_2} : D, B \rightarrow A \) are obtained from \( D \), so that their confidences might change as a result of data transformation for indirect discrimination prevention. Let us rewrite Inequality (6.21) in the following way

\[
\text{conf}(r_{b_1} : A, B \rightarrow D) < \alpha \cdot \frac{\text{conf}(B \rightarrow C) \cdot \text{conf}(r_{b_2})}{\text{conf}(r_{b_2}) + \text{conf}(r : D, B \rightarrow C) - 1} \tag{6.22}
\]

Clearly, in this case Inequality (6.20) can be satisfied by decreasing the confidence of rule \( r_{b_1} : A, B \rightarrow D \) to values less than the right-hand side of Inequality (6.22) without affecting either the confidence of the redlining rule or the confidence of the \( B \rightarrow C \) and \( r_{b_2} \) rules. Since the values of both inequality sides are dependent, a transformation is required that decreases the left-hand side of the inequality without any impact on the right-hand side. A possible solution for decreasing

\[
\text{conf}(A, B \rightarrow D) = \frac{\text{supp}(A, B, D)}{\text{supp}(A, B)} \tag{6.23}
\]

in Inequality (6.22) to the target value is to perturb the discriminatory itemset from \( \neg A \) to \( A \) in the subset \( D_c \) of all records of the original dataset which completely support the rule \( \neg A, B, \neg D \rightarrow \neg C \) and have minimum impact on other rules; this increases the denominator of Expression (6.23) while keeping the numerator and \( \text{conf}(B \rightarrow C) \), \( \text{conf}(r_{b_2} : D, B \rightarrow A) \), and \( \text{conf}(r : D, B \rightarrow C) \) unaltered.

\[4\] It is worth noting that \( \beta_1 \) and \( \beta_2 \) are lower bounds for \( \text{conf}(r_{b_1}) \) and \( \text{conf}(r_{b_2}) \), respectively, so it is correct if we replace \( \beta_1 \) and \( \beta_2 \) with \( \text{conf}(r_{b_1}) \) and \( \text{conf}(r_{b_2}) \) in the \( \text{elb} \) formulation.
There is another way to provide indirect rule protection. Let us rewrite Inequality (6.21) as Inequality (6.24), where the confidences of \( r_{b1} \) and \( r_{b2} \) rules are not constant.

\[
\text{conf}(B \rightarrow C) > \frac{\text{conf}(r_{b1}) (\text{conf}(r_{b2}) + \text{conf}(r : D, B \rightarrow C) - 1)}{\alpha}
\]

Clearly, in this case Inequality (6.20) can be satisfied by increasing the confidence of the base rule \( (B \rightarrow C) \) of the redlining rule \( r : D, B \rightarrow C \) to values greater than the right-hand side of Inequality (6.24) without affecting either the confidence of the redlining rule or the confidence of the \( r_{b1} \) and \( r_{b2} \) rules. A possible solution for increasing Expression (6.16) in Inequality (6.24) to the target value is to perturb the class item from \( \neg C \) to \( C \) in the subset \( D_{c} \) of all records of the original dataset which completely support the rule \( \neg A, B, \neg D \rightarrow \neg C \) and have minimum impact on other rules; this increases the numerator of Expression (6.16) while keeping the denominator and \( \text{conf}(r_{b1} : A, B \rightarrow D) \), \( \text{conf}(r_{b2} : D, B \rightarrow A) \), and \( \text{conf}(r : D, B \rightarrow C) \) unaltered.

Hence, like in direct rule protection, there are also two methods that could be applied for indirect rule protection. One method (Method 1) changes the discriminatory itemset in some records (e.g. from non-foreign worker to foreign worker in the records of hired people in NYC city with Zip\#10451) and the other method (Method 2) changes the class item in some records (e.g. from “Hire yes” to “Hire no” in the records of non-foreign worker of people in NYC city with Zip\#10451).

6.6 Data Transformation for Both Direct and Indirect Discrimination

We deal here with the key problem of transforming data with minimum information loss to prevent at the same time both direct and indirect discrimination. We will give a pre-processing solution to simultaneous direct and indirect discrimination prevention. First, we explain when direct and indirect discrimination could simultaneously occur. This depends on whether the original dataset \( (D) \) contains discriminatory itemsets or not. Two cases arise:

- PD itemsets (i.e. \( A \)) did not exist in the original database \( D \) or have previously been removed from it due to privacy constraints or for preventing discrimination. However, if background knowledge from publicly available data (e.g. census data) is available, indirect discrimination remains possible. In fact, in this case, only PND rules are extracted from \( D \) so only indirect discrimination could happen.

- At least one PD itemset (i.e. \( A \)) is not removed from the original database \( (D) \). So it is clear that PD rules could be extracted from \( D \) and direct discrimination could happen. However, in addition to direct discrimination, indirect discrimination might occur because of background knowledge obtained from \( D \) itself due to the existence of PND items that are highly correlated with the sensitive
Table 6.2: Direct and indirect rule protection methods

<table>
<thead>
<tr>
<th></th>
<th>Method 1</th>
<th>Method 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Rule Protection</td>
<td>¬A, B → ¬C ⇒ A, B → ¬C</td>
<td>¬A, B → ¬C ⇒ ¬A, B → ¬C</td>
</tr>
<tr>
<td>Indirect Rule Protection</td>
<td>¬A, B, ¬D → ¬C ⇒ A, B, ¬D → ¬C</td>
<td>¬A, B, ¬D → ¬C ⇒ ¬A, B, ¬D → ¬C</td>
</tr>
</tbody>
</table>

(discriminatory) ones. Hence, in this case both direct and indirect discrimination could happen.

To provide both direct rule protection (DRP) and indirect rule protection (IRP) at the same time, an important point is the relation between the data transformation methods. Any data transformation to eliminate direct $\alpha$-discriminatory rules should not produce new redlining rules or prevent the existing ones from being removed. Also any data transformation to eliminate redlining rules should not produce new direct $\alpha$-discriminatory rules or prevent the existing ones from being removed.

For subsequent use in this section, we summarize in Table 6.2 the methods for DRP and IRP described in Sections 6.4.1 and 6.5.1 above. We can see in Table 6.2 that DRP and IRP operate the same kind of data transformation: in both cases Method 1 changes the PD itemset, whereas Method 2 changes the class item. Therefore, in principle any data transformation for DRP (resp. IRP) not only does not need to have a negative impact on IRP (resp. DRP), but both kinds of protection could even be beneficial to each other.

However, there is a difference between DRP and IRP: the set of records chosen for transformation. As shown in Table 6.2, in IRP the chosen records should not satisfy the $D$ itemset (chosen records are those with $¬A, B, ¬D → ¬C$), whereas DRP does not care about $D$ at all (chosen records are those with $¬A, B → ¬C$). The following interactions between direct and indirect rule protection become apparent.

**Lemma 2.** Method 1 for DRP cannot be used if simultaneous DRP and IRP are desired.

**Proof:** Method 1 for DRP might undo the protection provided by Method 1 for IRP, as we next justify. Method 1 for DRP decreases $conf(A, B → D)$ until the direct rule protection requirement (Inequality (6.13)) is met and Method 1 for IRP needs to decrease $conf(A, B → D)$ until the indirect rule protection requirement is met (Inequality (6.22)). Assume that decreasing $conf(A, B → D)$ to meet the direct rule protection requirement is achieved by changing $y$ (how $y$ is obtained will be discussed in Section 6.8) number of records with $¬A, B, ¬C$ to records with $A, B, ¬C$ (as done by Method 1 for DRP). This actually could increase $conf(A, B → D)$ if $z$ among the changed records, with $z ≤ y$, turn out to satisfy $D$. This increase can undo the protection provided by Method 1 for IRP (i.e. $conf(A, B → D) < IRP_{req1}$, where $IRP_{req1} = \frac{\alpha conf(B → C) - conf(r_{r12})}{\alpha conf(B → C) - conf(r_{r12}) + conf(r_{D,B → C}) - 1}$) if the new value $conf(A, B → D) = \frac{supp(A,B,D) + z}{supp(A,B) + y}$ is greater than or equal to $IRP_{req1}$, which happens if $z ≥ IRP_{req1} \cdot (supp(A, B) + Y) - supp(A, B, D)$.

**Lemma 3.** Method 2 for IRP is beneficial for Method 2 for DRP. On the other hand, Method 2 for DRP is at worst neutral for Method 2 for IRP.
Proof: Method 2 for DRP and Method 2 for IRP are both aimed at increasing $\text{conf}(B \rightarrow C)$. In fact, Method 2 for IRP changes a subset of the records changed by Method 2 for DRP. This proves that Method 2 for IRP is beneficial for Method 2 for DRP. On the other hand, let us check that, in the worst case, Method 2 for DRP is neutral for Method 2 for IRP: such a worst case is the one in which all changed records satisfy $D$, which could result in increasing both sides of Inequality (6.24) by an equal amount (due to increasing $\text{conf}(B \rightarrow C)$ and $\text{conf}(D, B \rightarrow C)$); even in this case, there is no change in whatever protection is achieved by Method 2 for IRP. □

Thus, we conclude that Method 2 for DRP and Method 2 for IRP are the only methods among those described that can be applied to achieve simultaneous direct and indirect discrimination prevention. In addition, in the cases where either only direct or only indirect discrimination exist, there is no interference between the described methods: Method 1 for DRP, Method 2 for DRP and Rule Generalization can be used to prevent direct discrimination; Method 1 for IRP and Method 2 for IRP can be used to prevent indirect discrimination. In what follows, we propose algorithms based on the described methods that cover direct and/or indirect discrimination prevention.

6.7 The Algorithms

We describe in this section our algorithms based on the direct and indirect discrimination prevention methods proposed in Sections 6.4, 6.5 and 6.6. There are some assumptions common to all algorithms in this section. First, we assume the class attribute in the original dataset $D$ to be binary (e.g. denying or granting credit). Second, we consider classification rules with negative decision (e.g. denying credit) to be in $\mathcal{FR}$. Third, we assume the PD itemsets (i.e. $A$) and the PND itemsets (i.e. $D$) to be binary or non-binary categorical.

6.7.1 Direct Discrimination Prevention Algorithms

We start with direct rule protection. Algorithm 7 details Method 1 for DRP. For each direct $\alpha$-discriminatory rule $r \in \mathcal{MR}$ (Step 3), after finding the subset $D_c$ (Step 5), records in $D_c$ should be changed until the direct rule protection requirement (Step 10) is met for each respective rule (Steps 10-14).

Among the records of $D_c$, one should change those with lowest impact on the other ($\alpha$-protective or non-redlining) rules. Hence, for each record $db_c \in D_c$, the number of rules whose premise is supported by $db_c$ is taken as the impact of $db_c$ (Step 7), that is $\text{impact}(db_c)$; the rationale is that changing $db_c$ impacts on the confidence of those rules. Then the records $db_c$ with minimum $\text{impact}(db_c)$ are selected for change (Step 9), with the aim of scoring well in terms of the utility measures proposed in the next section. We call this procedure (Steps 6-9) impact minimization and we re-use it in the pseudocodes of the rest of algorithms specified in this chapter.

Algorithm 8 details Method 2 for DRP. The parts of Algorithm 8 to find subset $D_c$ and perform impact minimization (Step 4) are the same as in Algorithm 7. However, the transformation requirement that should be met for each $\alpha$-discriminatory rule in $\mathcal{MR}$ (Step 5) and the kind of data transformation are different (Steps 5-9).
Algorithm 7 DIRECT RULE PROTECTION (METHOD 1)

1: Inputs: $D$, $FR$, $MR$, $\alpha$, $DI_b$
2: Output: $D'$ (transformed dataset)
3: for each $r' : A, B \rightarrow C \in MR$ do
4:    $FR \leftarrow FR - \{r'\}$
5:    $D_c \leftarrow$ All records completely supporting $\neg A, B \rightarrow \neg C$
6:    for each $db_c \in D_c$ do
7:        Compute $impact(db_c) = |\{r_a \in FR| db_c supports the premise of r_a\}|$
8:    end for
9:    Sort $D_c$ by ascending impact
10: while $conf(r') \geq \alpha \cdot conf(B \rightarrow C)$ do
11:    Select first record in $D_c$
12:    Modify PD itemset of $db_c$ from $\neg A$ to $A$ in $D$
13:    Recompute $conf(r')$
14: end while
15: end for
16: Output: $D' = D$

Algorithm 8 DIRECT RULE PROTECTION (METHOD 2)

1: Inputs: $D$, $FR$, $MR$, $\alpha$, $DI_b$
2: Output: $D'$ (transformed dataset)
3: for each $r' : A, B \rightarrow C \in MR$ do
4:    Steps 4-9 Algorithm 7
5:    while $conf(B \rightarrow C) \leq \frac{conf(r')}{\alpha}$ do
6:        Select first record in $D_c$
7:        Modify the class item of $db_c$ from $\neg C$ to $C$ in $D$
8:        Recompute $conf(B \rightarrow C)$
9: end while
10: end for
11: Output: $D' = D$

As mentioned in Section 6.4.3, rule generalization cannot be applied alone for solving direct discrimination prevention, but it can be used in combination with Method 1 or Method 2 for DRP. In this case, after specifying the discrimination prevention method (i.e. direct rule protection or rule generalization) to be applied for each $\alpha$-discriminatory rule based on the algorithm in Section 6.4.3, Algorithm 9 should be run to combine rule generalization and one of the two direct rule protection methods.

Algorithm 9 takes as input $TR$, which is the output of the algorithm in Section 6.4.3, containing all $r' \in MR$ and their respective $TR_{r'}$ and $r_b$. For each $\alpha$-discriminatory rule $r'$ in $TR$, if $TR_{r'}$ shows that rule generalization should be performed (Step 5), after determining the records that should be changed for impact minimization (Steps 7-8), these records should be changed until the rule generalization requirement is met (Steps 9-13). Also, if $TR_{r'}$ shows that direct rule protection should be performed (Step 15), based on either Method 1 or Method 2, the relevant sections of Algorithm 7 or 8 are called, respectively (Step 17).
Algorithm 9 DIRECT RULE PROTECTION AND RULE GENERALIZATION

1: Inputs: \(D, FR, TR, p \geq 0.8, \alpha, DI_b\)
2: Output: \(D'\) (transformed dataset)
3: for each \(r' : A, B \rightarrow C \in TR\) do
4: \(FR \leftarrow FR - \{r'\}\)
5: if \(TR_{r'} = RG\) then
6: // Rule Generalization
7: \(D_c \leftarrow \) All records completely supporting \(A, B, \neg D \rightarrow C\)
8: // Steps 6-9 Algorithm 7
9: while \(conf(r') > conf((r_c : D, B \rightarrow C))\) do
10: Select first record in \(D_c\)
11: Modify class item of \(db_c\) from \(C\) to \(\neg C\) in \(D\)
12: Recompute \(conf(r')\)
13: end while
14: end if
15: if \(TR_{r'} = DRP\) then
16: // Direct Rule Protection
17: // Steps 5-14 Algorithm 7 or Steps 4-9 Algorithm 8
18: end if
19: end for
20: Output: \(D' = D\)

6.7.2 Indirect Discrimination Prevention Algorithms

A detailed algorithm implementing Method 2 for IRP is provided in [43], from which an algorithm implementing Method 1 for IRP can be easily derived. Due to similarity with the previous algorithms, we do not recall those two algorithms for IRP here.

6.7.3 Direct and Indirect Discrimination Prevention Algorithms

Algorithm 10 details our proposed data transformation method for simultaneous direct and indirect discrimination prevention. The algorithm starts with redlining rules. From each redlining rule \((r : X \rightarrow C)\), more than one indirect \(\alpha\)-discriminatory rule \((r' : A, B \rightarrow C)\) might be generated because of two reasons: 1) existence of different ways to group the items in \(X\) into a context itemset \(B\) and a PND itemset \(D\) correlated to some PD itemset \(A\); and 2) existence of more than one item in \(DI_b\). Hence, as shown in Algorithm 10 (Step 5), given a redlining rule \(r\), proper data transformation should be conducted for all indirect \(\alpha\)-discriminatory rules \(r' : (A \subseteq DI_b), (B \subseteq X) \rightarrow C\) ensuing from \(r\).

If some rules can be extracted from \(D\) as both direct and indirect \(\alpha\)-discriminatory rules, it means that there is overlap between \(\mathcal{MR}\) and \(\mathcal{RR}\); in such case, data transformation is performed until both the direct and the indirect rule protection requirements are satisfied (Steps 13-18). This is possible because, as we showed in Section 6.6, the same data transformation method (Method 2 consisting of changing the class item) can provide both DRP and IRP. However, if there is no overlap between \(\mathcal{MR}\) and \(\mathcal{RR}\), the data transformation is performed according to Method 2 for IRP, until the indirect discrimination prevention requirement is satisfied (Steps 19-23) for each indirect
Algorithm 10 DIRECT AND INDIRECT DISCRIMINATION PREVENTION

1: Inputs: \( \mathcal{D}, \mathcal{F}, \mathcal{R}, \mathcal{M}, \mathcal{R}, \alpha, \mathcal{D}_0 \)
2: Output: \( \mathcal{D}' \) (transformed dataset)
3: for each \( r: X \rightarrow C \in \mathcal{R} \), where \( D, B \subseteq X \) do
4: \( \gamma = \text{conf}(r) \)
5: for each \( r': (A \subseteq \mathcal{D}_0), (B \subseteq X) \rightarrow C \in \mathcal{R} \) do
6: \( \beta_2 = \text{conf}(r_{b2}: X \rightarrow A) \)
7: \( \Delta_1 = \text{supp}(r_{b2}: X \rightarrow A) \)
8: \( \delta = \text{conf}(B \rightarrow C) \)
9: \( \Delta_2 = \text{supp}(B \rightarrow A) \)
10: \( \beta_1 = \frac{\Delta_1}{\Delta_2} \cdot \text{conf}(r_{b1}: A, B \rightarrow \mathcal{D}) \)
11: Find \( \mathcal{D}_c \): all records in \( \mathcal{D} \) that completely support \( \neg A, B, \neg D \rightarrow \neg C \)
12: Steps 6-9 Algorithm 7
13: if \( r' \in \mathcal{M} \) then
14: while \( (\delta \leq \frac{\Delta_1}{\Delta_2} (\beta_2 + \frac{\gamma - 1}{\alpha})) \) and \( (\delta \leq \frac{\text{conf}(r')}{\alpha}) \) do
15: Select first record \( d_{bc} \) in \( \mathcal{D}_c \)
16: Modify the class item of \( d_{bc} \) from \( \neg C \) to \( C \) in \( \mathcal{D} \)
17: Recompute \( \delta = \text{conf}(B \rightarrow C) \)
18: end while
19: else
20: while \( \delta \leq \frac{\Delta_1}{\Delta_2} (\beta_2 + \frac{\gamma - 1}{\alpha}) \) do
21: Steps 15-17 Algorithm 10
22: end while
23: end if
24: end for
25: for each \( r': (A, B \rightarrow C) \in \mathcal{M} \setminus \mathcal{R} \) do
26: \( \delta = \text{conf}(B \rightarrow C) \)
27: while \( (\delta \leq \frac{\text{conf}(r')}{\alpha}) \) do
28: Find \( \mathcal{D}_c \): all records in \( \mathcal{D} \) that completely support \( \neg A, B \rightarrow \neg C \)
29: Step 12
30: end while
31: Steps 15-17 Algorithm 10
32: end for
33: end for
34: Output: \( \mathcal{D}' = \mathcal{D} \)

\( \alpha \)-discriminatory rule ensuing from each redlining rule in \( \mathcal{R} \); this can be done without any negative impact on direct discrimination prevention, as justified in Section 6.6. Then, for each direct \( \alpha \)-discriminatory rule \( r' \in \mathcal{M} \setminus \mathcal{R} \) (that is only directly extracted from \( \mathcal{D} \)), data transformation for satisfying the direct discrimination prevention requirement is performed (Steps 26-33), based on Method 2 for DRP; this can be done without any negative impact on indirect discrimination prevention, as justified in Section 6.6. Performing rule protection or generalization for each rule in \( \mathcal{M} \) by each of Algorithms 7-10 has no adverse effect on protection for other rules (i.e. rule protection at Step \( i + x \) to make \( r^i \) protective cannot turn into discriminatory a rule \( r \) made protective at Step \( i \)) because of the following two reasons: the kind of data transformation for each rule is the same (change the PD itemset or the class item of records) and there are no two \( \alpha \)-discriminatory rules \( r \) and \( r' \) in \( \mathcal{M} \) such that \( r = r' \).

6.8 Computational Cost

The computational cost of Algorithm 7 can be broken down as follows:

- Let \( m \) be the number of records in \( \mathcal{D} \). The cost of finding subset \( \mathcal{D}_c \) (Step 5) is
Let \( k \) be the number of rules in \( FR \) and \( h \) the number of records in subset \( D_c \). The cost of computing \( \text{impact}(d_{bc}) \) for all records in \( D_c \) (Steps 6-8) is \( O(hk) \).

The cost of sorting \( D_c \) by ascending impact (Step 9) is \( O(h \log h) \). Then, the cost of the impact minimization procedure (Steps 6-9) in all algorithms is \( O(hk + h \log h) \).

During each iteration of the inner loop (Step 10), the number of records supporting the premise of rule \( r' : A, B \rightarrow C \) is increased by one. After \( i \) iterations, the confidence of rule \( r' : A, B \rightarrow C \) is \( \text{conf}(r' : A, B \rightarrow C)^{(i)} = \frac{N_{ABC}}{N_{AB+i}} \), where \( N_{ABC} \) is the number of records supporting rule \( r' \) and \( N_{AB} \) is the number of records supporting the premise of rule \( r' \). If we let \( DRP_{req1} = \alpha \cdot \text{conf}(B \rightarrow C) \), the inner loop (Step 10) is iterated until \( \text{conf}(r' : A, B \rightarrow C)^{(i)} < DRP_{req1} \) or equivalently \( \frac{N_{ABC}}{N_{AB+i}} < DRP_{req1} \). This inequality can be rewritten as \( i > \left( \frac{N_{ABC}}{DRP_{req1}} - N_{BC} \right) \). From this last inequality we can derive that \( i = \left\lceil \frac{N_{ABC}}{DRP_{req1}} - N_{BC} \right\rceil \). Hence, iterations in the inner loop (Step 10) will stop as soon as the first integer value greater than (or equal) \( \frac{N_{ABC}}{DRP_{req1}} - N_{BC} \) is reached. Then, the cost spent on the inner loop to satisfy the direct rule protection requirement (Steps 10-14) will be \( O(m \ast \left\lceil \frac{N_{ABC}}{DRP_{req1}} - N_{BC} \right\rceil) \).

Therefore, assuming \( n \) is the number of \( \alpha \)-discriminatory rules in \( MR \) (Step 3), the total computational time of Algorithm 7 is bounded by \( O(n \ast \{m + hk + h \log h + im\}) \), where \( d = \left\lceil \frac{N_{ABC}}{DRP_{req1}} - N_{BC} \right\rceil \).

The impact minimization procedure substantially increases the complexity. Without computing the impact, the time complexity of Algorithm 7 decreases to \( O(n \ast \{m + im\}) \). In addition, it is clear that the execution time of Algorithm 7 increases linearly with the number \( m \) of original data records as well as the number \( n \) of frequent classification rules and the number \( n \) of \( \alpha \)-discriminatory rules.

The computational cost of the other algorithms can be computed similarly, with some small differences. In summary, the total computational time of Algorithm 8 is also bounded by \( O(n \ast \{m + hk + h \log h + im\}) \), where \( i = \left\lceil (N_B + DRP_{req2}) - N_{BC} \right\rceil \), \( N_{BC} \) is the number of records supporting rule \( B \rightarrow C \), \( N_B \) is the number of records supporting itemset \( B \) and \( DRP_{req2} = \frac{\text{conf}(r')}{\alpha} \). The computational cost of Algorithm 9 is the same as the last ones with the difference that \( i = \left\lceil N_{ABC} - (RG_{req} \ast N_{AB}) \right\rceil \), where \( RG_{req} = \frac{\text{conf}(r_s)}{\alpha} \), or \( d = \left\lceil (N_B + DRP_{req2}) - N_{BC} \right\rceil \), depending on whether rule generalization or direct rule protection is performed.

Finally, assuming \( f \) is the number of indirect \( \alpha \)-discriminatory rules in \( RR \) and \( n \) is the number of direct \( \alpha \)-discriminatory rules in \( MR \) that no longer exist in \( RR \), the total computational time of Algorithm 10 is bounded by \( O((f+n) \ast \{m + hk + h \log h + im\}) \), where \( i = \left\lceil (N_B \ast \text{max}_{\text{req}}) - N_{BC} \right\rceil \) and \( \text{max}_{\text{req}} = \max\left(\frac{2f(\beta_2^{\gamma+1} - 1) + \text{conf}(r')}{\beta_2^{\gamma+1}}, \frac{\text{conf}(r')}{\alpha}\right) \).
6.9 Experiments

This section presents the experimental evaluation of the proposed direct and/or indirect discrimination prevention approaches and algorithms. To obtain $\mathcal{FR}$ and $\mathcal{BK}$ we used the Apriori algorithm [6], which is a common algorithm to extract frequent rules. All algorithms and utility measures were implemented using the C# programming language. The tests were performed on an 2.27 GHz Intel® Core™ i3 machine, equipped with 4 GB of RAM, and running under Windows 7 Professional.

First, we describe the datasets used in our experiments. Then, we introduce the new utility measures we propose to evaluate direct and indirect discrimination prevention methods in terms of their success at discrimination removal and impact on data quality. Finally, we present the evaluation results of the different methods and also the comparison between them.

6.9.1 Datasets

Adult dataset: We used the Adult dataset from [38], also known as Census Income, in our experiments. This dataset consists of 48,842 records, split into a “train” part with 32,561 records and a “test” part with 16,281 records. The dataset has 14 attributes (without class attribute). We used the “train” part in our experiments. The prediction task associated to the Adult dataset is to determine whether a person makes more than 50K$ a year based on census and demographic information about people. The dataset contains both categorical and numerical attributes.

For our experiments with the Adult dataset, we set $\mathcal{DI}_b = \{ \text{Sex} = \text{Female}, \text{Age} = \text{Young} \}$. Although the Age attribute in the Adult dataset is numerical, we converted it to categorical by partitioning its domain into two fixed intervals: Age $\leq 30$ was renamed as Young and Age $> 30$ was renamed as old.

German Credit dataset: We also used the German Credit dataset from [38]. This dataset consists of 1,000 records and 20 attributes (without class attribute) of bank account holders. This is a well-known real-life dataset, containing both numerical and categorical attributes. It has been frequently used in the anti-discrimination literature [77, 56]. The class attribute in the German Credit dataset takes values representing good or bad classification of the bank account holders. For our experiments with this dataset, we set $\mathcal{DI}_b = \{ \text{Foreign worker} = \text{Yes}, \text{Personal Status} = \text{Female and not Single}, \text{Age} = \text{Old} \}$ (cut-off for Age=Old: 50 years old).

6.9.2 Utility Measures

Our proposed techniques should be evaluated based on two aspects. On the one hand, we need to measure the success of the method in removing all evidence of direct and/or indirect discrimination from the original dataset; on the other hand, we need to measure the impact of the method in terms of information loss (i.e. data quality loss). To measure discrimination removal, four metrics were used:

- **Direct Discrimination Prevention Degree (DDPD).** This measure quantifies the percentage of $\alpha$-discriminatory rules that are no longer $\alpha$-discriminatory in
the transformed dataset. We define DDPD as
\[
DDPD = \frac{|MR| - |MR'|}{|MR|}
\]
where \(MR\) is the database of \(\alpha\)-discriminatory rules extracted from \(D\) and \(MR'\) is the database of \(\alpha\)-discriminatory rules extracted from the transformed dataset \(D'\). Note that \(|·|\) is the cardinality operator.

**Direct Discrimination Protection Preservation (DDPP).** This measure quantifies the percentage of the \(\alpha\)-protective rules in the original dataset that remain \(\alpha\)-protective in the transformed dataset. It is defined as
\[
DDPP = \frac{|PR \cap PR'|}{|PR|}
\]
where \(PR\) is the database of \(\alpha\)-protective rules extracted from the original dataset \(D\) and \(PR'\) is the database of \(\alpha\)-protective rules extracted from the transformed dataset \(D'\).

**Indirect Discrimination Prevention Degree (IDPD)** This measure quantifies the percentage of redlining rules that are no longer redlining in the transformed dataset. It is defined like DDPD but substituting \(MR\) and \(MR'\) with the database of redlining rules extracted from \(D\) and \(D'\), respectively.

**Indirect Discrimination Protection Preservation (IDPP)** This measure quantifies the percentage of non-redlining rules in the original dataset that remain non-redlining in the transformed dataset. It is defined like DDPP but substituting \(PR\) and \(PR'\) with the database of non-redlining extracted from \(D\) and \(D'\), respectively.

Since the above measures are used to evaluate the success of the proposed method in direct and indirect discrimination prevention, ideally their value should be 100%. To measure data quality, we use two metrics proposed in the literature as information loss measures in the context of rule hiding for privacy-preserving data mining (PPDM) [102].

**Misses Cost (MC).** This measure quantifies the percentage of rules among those extractable from the original dataset that cannot be extracted from the transformed dataset (side-effect of the transformation process).

**Ghost Cost (GC).** This measure quantifies the percentage of the rules among those extractable from the transformed dataset that were not extractable from the original dataset (side-effect of the transformation process).

MC and GC should ideally be 0%. However, MC and GC may not be 0% as a side-effect of the transformation process.
6.9.3 Empirical Evaluation

We implemented the algorithms for all proposed methods for direct and/or indirect discrimination prevention, and we evaluated them in terms of the proposed utility measures. We report the performance results in this section.

Tables 6.3 and 6.4 show the utility scores obtained by our methods on the Adult dataset and the German Credit dataset, respectively. Within each table, the first row relates to the simple approach of deleting discriminatory attributes, the next four rows relate to direct discrimination prevention methods, the next two ones relate to indirect discrimination prevention methods and the last one relates to the combination of direct and indirect discrimination.

Table 6.3 shows the results for minimum support 2% and minimum confidence 10% for all the methods. Value “n.a.” denotes that the respective measure is not applicable.

Table 6.4: German Credit dataset: Utility measures for minimum support 5% and confidence 10% for all methods. Value “n.a.” denotes that the respective measure is not applicable.

Table 6.3: Adult dataset: Utility measures for minimum support 2% and confidence 10% for all the methods. Value “n.a.” denotes that the respective measure is not applicable.

### Table 6.3: Adult dataset

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Removing Disc. Attributes</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>DRP (Method 1)</td>
<td>1.2</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>274</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 2)</td>
<td>1.2</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>274</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 1) + RG</td>
<td>1.2</td>
<td>0.9</td>
<td>n.a.</td>
<td>n.a.</td>
<td>274</td>
<td>91.58</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 2) + RG</td>
<td>1.2</td>
<td>0.9</td>
<td>n.a.</td>
<td>n.a.</td>
<td>274</td>
<td>91.58</td>
<td>100</td>
</tr>
<tr>
<td>IRP (Method 1)</td>
<td>1.1</td>
<td>n.a.</td>
<td>21</td>
<td>n.a.</td>
<td>n.a.</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>IRP (Method 2)</td>
<td>1.1</td>
<td>n.a.</td>
<td>30</td>
<td>n.a.</td>
<td>n.a.</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

No of Freq. Class. Rules: 5,092
No of Back. Know. Rules: 2089

### Table 6.4: German Credit dataset

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Removing Disc. Attributes</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>DRP (Method 1)</td>
<td>1.2</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>991</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 2)</td>
<td>1.2</td>
<td>n.a.</td>
<td>n.a.</td>
<td>n.a.</td>
<td>991</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 1) + RG</td>
<td>1.2</td>
<td>0.9</td>
<td>n.a.</td>
<td>n.a.</td>
<td>991</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 2) + RG</td>
<td>1.2</td>
<td>0.9</td>
<td>n.a.</td>
<td>n.a.</td>
<td>991</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>IRP (Method 1)</td>
<td>1</td>
<td>n.a.</td>
<td>37</td>
<td>n.a.</td>
<td>n.a.</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>IRP (Method 2)</td>
<td>1</td>
<td>n.a.</td>
<td>42</td>
<td>n.a.</td>
<td>n.a.</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 2) + IRP (Method 2)</td>
<td>1</td>
<td>n.a.</td>
<td>37</td>
<td>n.a.</td>
<td>499</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

No of Freq. Class. Rules: 32,340
No of Back. Know. Rules: 22,763

6.9.3 Empirical Evaluation

We implemented the algorithms for all proposed methods for direct and/or indirect discrimination prevention, and we evaluated them in terms of the proposed utility measures. We report the performance results in this section.

Tables 6.3 and 6.4 show the utility scores obtained by our methods on the Adult dataset and the German Credit dataset, respectively. Within each table, the first row relates to the simple approach of deleting discriminatory attributes, the next four rows relate to direct discrimination prevention methods, the next two ones relate to indirect discrimination prevention methods and the last one relates to the combination of direct and indirect discrimination.

Table 6.3 shows the results for minimum support 2% and minimum confidence 10%. Table 6.4 shows the results for minimum support 5% and minimum confidence 10%. In Tables 6.3 and 6.4, the results of direct discrimination prevention methods are reported for discriminatory threshold $\alpha = 1.2$ and, in the cases where direct rule protection is applied in combination with rule generalization, we used $d = 0.9$, and $DI_b = \{\text{Sex=Female, Age=Young}\}$ in the Adult dataset, and $DI_b = \{\text{Foreign worker=Yes, Personal Status=Female and not Single, Age=Old}\}$ in the German Credit dataset. In addition, in Table 6.3, the results of the indirect discrimination prevention methods and both direct and indirect discrimination prevention are reported for discriminatory threshold $\alpha = 1.1$ and $DI_b = \{\text{Sex=Female, Age=Young}\}$; in Table 6.4, these results are reported for $\alpha = 1$ and $DI_b = \{\text{Foreign worker=Yes}\}$.

115
We selected the discriminatory threshold values and $DI_b$ for each dataset in such a way that the number of redlining rules and $\alpha$-discriminatory rules extracted from $D$ could be suitable to test all our methods. In addition to the scores of utility measures, the number of redlining rules, the number of indirect $\alpha$-discriminatory rules and the number of direct $\alpha$-discriminatory rules are also reported in Tables 6.3 and 6.4. These tables also show the number of frequent classification rules found, as well as the number of background knowledge rules related to this experiment.

As shown in Tables 6.3 and 6.4, we get very good results for all methods in terms of discrimination removal: DDPD, DDPP, IDPD, IDPP are near 100% for both datasets. In terms of data quality, the best results for direct discrimination prevention are obtained with Method 2 for DRP or Method 2 for DRP combined with Rule Generalization. The best results for indirect discrimination prevention are obtained with Method 2 for IRP. This shows that lower information loss is obtained with the methods changing the class item (i.e. Method 2) than with those changing the discriminatory itemset (i.e. Method 1). As mentioned above, in direct discrimination prevention, rule generalization cannot be applied alone and must be applied in combination with direct rule protection; however, direct rule protection can be applied alone. The results in the last row of the above tables (i.e. Method 2 for DRP + Method 2 for IRP) based on Algorithm 10 for the case of simultaneous direct and indirect discrimination demonstrate that the proposed solution achieves a high degree of simultaneous direct and indirect discrimination removal with very little information loss.

For all methods, Tables 6.3 and 6.4 show that we obtained lower information loss in terms of MC and GC in the Adult dataset than in the German Credit dataset. In terms of discrimination removal, results on both datasets were almost the same. In addition, the highest value of information loss is obtained by the simple approach of removing discriminatory attributes (first row of each table): as it could be expected, entirely suppressing the PD attributes is much more information-damaging than modifying the values of these attributes in a few records.

After the above general results and comparison between methods, we now present...
more specific results on each method for different parameters $\alpha$ and $d$. Figure 6.3 shows on the left the degree of information loss (as average of MC and GC) and on the right the degree of discrimination removal (as average of DDPD and DDPP) of direct discrimination prevention methods for the German Credit dataset when the value of the discriminatory threshold $\alpha$ varies from 1.2 to 1.7, $d$ is 0.9, the minimum support is 5% and the minimum confidence is 10%. The number of direct $\alpha$-discriminatory rules extracted from the dataset is 991 for $\alpha = 1.2$, 415 for $\alpha = 1.3$, 207 for $\alpha = 1.4$, 120 for $\alpha = 1.5$, 63 for $\alpha = 1.6$ and 30 for $\alpha = 1.7$, respectively. As shown in Figure 6.3, the degree of discrimination removal provided by all methods for different values of $\alpha$ is also 100%. However, the degree of information loss decreases substantially as $\alpha$ increases; the reason is that, as $\alpha$ increases, the number of $\alpha$-discriminatory rules to be dealt with decreases. In addition, as shown in Figure 6.3, the lowest information loss for most values of $\alpha$ is obtained by Method 2 for DRP.

In addition, to demonstrate the impact of varying $d$ on the utility measures in the methods using Rule Generalization, Figure 6.4 (left) shows the degree of information loss and Figure 6.4 (right) shows the degree of discrimination removal for different values of $d$ ($0.8, 0.85, 0.9, 0.95$) and $\alpha=1.2$ for the German Credit dataset. Although the values of DDPD and DDPP achieved for different values of $p$ remain almost the same, increasing the value of $d$ leads to an increase of MC and GC because, to cope with the rule generalization requirements, more data records must be changed.

Tables 6.5 and 6.6 show the utility measures obtained by running Algorithm 10 to achieve simultaneous direct and indirect discrimination prevention (i.e. Method 2 for DRP+ Method 2 for IRP) on the Adult and German credit datasets, respectively. In Table 6.5 the results are reported for different values of $\alpha \in [1, 1.5]$; in Table 6.6 different values of $\alpha \in [1, 1.4]$ are considered. We selected these $\alpha$ intervals in such a way that, with respect to the predetermined discriminatory items in this experiment for the Adult dataset (i.e. $DI_{\alpha} =$ {Sex=Female, Age=Young}) and the German Credit dataset (i.e. $DI_{\alpha} =$ {Foreign worker=Yes}), both direct $\alpha$-discriminatory and redlining rules could be extracted. The reason is that we need to detect some cases with both direct and indirect discrimination to be able to test our method. Moreover, we restricted
the lower bound to limit the number of direct $\alpha$-discriminatory and redlining rules. In addition to utility measures, the number of redlining rules, the number of indirect $\alpha$-discriminatory rules and the number of direct $\alpha$-discriminatory rules are also reported for different values of $\alpha$. Value “n.a.” denotes that the respective measure is not applicable.

### Table 6.5: Adult dataset: Utility measures for minimum support 2% and confidence 10% for direct and indirect rule protection; columns show the results for different values of $\alpha$. Value “n.a.” denotes that the respective measure is not applicable.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>No. of Direct $\alpha$-Disc. rules</th>
<th>No. of Indirect $\alpha$-Disc. rules</th>
<th>Discrimination Removal</th>
<th>Data Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct DDP</td>
<td>DDPP</td>
</tr>
<tr>
<td>$\alpha=1$</td>
<td>43</td>
<td>71</td>
<td>89.45</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.1$</td>
<td>21</td>
<td>30</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.2$</td>
<td>9</td>
<td>14</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.3$</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.4$</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.5$</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

No of Freq. Class. Rules: 5,092

### Table 6.6: German Credit dataset: Utility measures for minimum support 5% and confidence 10% for direct and indirect rule protection; columns show the results for different values of $\alpha$. Value “n.a.” denotes that the respective measure is not applicable.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>No. of Direct $\alpha$-Disc. rules</th>
<th>No. of Indirect $\alpha$-Disc. rules</th>
<th>Discrimination Removal</th>
<th>Data Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Direct DDP</td>
<td>DDPP</td>
</tr>
<tr>
<td>$\alpha=1$</td>
<td>37</td>
<td>42</td>
<td>99.97</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.1$</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.2$</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.3$</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$\alpha=1.4$</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

No of Freq. Class. Rules: 32,340

The values of both direct discrimination removal measures (i.e. DDPD and DDPP) and indirect discrimination removal measures (i.e. IDPD and IDPP) shown in Tables 6.5 and 6.6 demonstrate that the proposed solution achieves a high degree of both direct and indirect discrimination prevention for different values of the discriminatory threshold. The important point is that, by applying the proposed method, we get good results for both direct and indirect discrimination prevention at the same time. In addition, the values of MC and GC demonstrate that the proposed solution incurs low information loss.

Tables 6.5 and 6.6 show that we obtained lower information loss in terms of the GC measure in the Adult dataset than in the German Credit dataset. Another remark on these tables is that, although no redlining rules are detected in the Adult dataset for $\alpha \geq 1.3$ and in the German Credit dataset for $\alpha \geq 1.1$, the IDPP measure is computed and reported to show that in the cases where only direct discrimination exists, the elimination of direct discrimination by Algorithm 10 does not have a negative impact on indirect discrimination (i.e. non-redlining rules do not become redlining rules).

Figure 6.5 illustrates the effect of the impact minimization procedure, described in Section 6.7.1, on execution times and information loss of Method 1 for DRP, respec-
Figure 6.5: Execution times (left) and Information loss degree (right) of Method 1 for DRP for $\alpha \in [1.2, 1.7]$ with and without impact minimization.

Table 6.7: Adult dataset: number of frequent classification rules and $\alpha-$discriminatory rules found during the tests, for minimum confidence 10% and different values of minimum support (2%, 5%, and 10%)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>2%</th>
<th>5%</th>
<th>10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>274</td>
<td>46</td>
<td>27</td>
</tr>
</tbody>
</table>

As shown in this figure (right) impact minimization has a noticeable effect on information loss (decreasing MC and GC). However, as discussed in Section 6.8 and shown in Figure 6.5 (left), impact minimization substantially increases the execution time of the algorithm. For other methods, the same happens. Figure 6.5 (left) also shows that, by increasing $\alpha$, the number of $\alpha$-discriminatory rules and hence the execution time are decreased. Additional experiments are presented in the Appendix to show the effect of varying the minimum support and the minimum confidence on the proposed techniques.

As shown in Table 6.7, different values of the minimum support have an impact on the number of frequent rules and hence on the number of $\alpha$-discriminatory rules. As shown in Table 6.8, by increasing the value of the minimum support the information loss degrees (MC and GC) achieved by the different techniques decrease. Meanwhile, as shown in Table 6.8, the discrimination removal degrees (DDPD and DDPP) achieved by the different techniques remain the same (discrimination removal is maximum) for different values of the minimum support.

As shown in the left-hand side graph of Figure 6.6, different values of minimum confidence have a non-uniform impact on the information loss degree (average of MC and MC): sometimes increasing the minimum confidence can decrease the information loss degree and sometimes it can increase the information loss degree. On the other hand, the right-hand side graph of Figure 6.6 shows that the average of the discrimination removal degrees DDPD and DDPP achieved by different techniques remains the same (discrimination removal is maximum) for different values of the minimum support.
Table 6.8: Adult dataset: utility measures for minimum confidence 10%, $\alpha=1.2$ and $d = 0.9$; columns show the results for different values of minimum support (2%, 5% and 10%) and different methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MC 2%</th>
<th>MC 5%</th>
<th>MC 10%</th>
<th>GC 2%</th>
<th>GC 5%</th>
<th>GC 10%</th>
<th>DDPP 2%</th>
<th>DDPP 5%</th>
<th>DDPP 10%</th>
<th>DDPP 2%</th>
<th>DDPP 5%</th>
<th>DDPP 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRP (Method 1)</td>
<td>4.16</td>
<td>2.91</td>
<td>1.61</td>
<td>4.13</td>
<td>3.21</td>
<td>0.39</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 1) + RG</td>
<td>4.1</td>
<td>2.67</td>
<td>1.61</td>
<td>4.1</td>
<td>3.26</td>
<td>0.39</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DRP (Method 2) + RG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>91.58</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Figure 6.6: Adult dataset: Information loss (left) and discrimination removal degree (right) for discrimination prevention methods for minimum support=2%, $\alpha = 1.2$, $p = 0.9$ and minimum confidence in [10, 90]. DRP(Method i): Method i for DRP; RG: Rule Generalization.

6.10 Conclusions

Along with privacy, discrimination is a very important issue when considering the legal and ethical aspects of data mining. It is more than obvious that most people do not want to be discriminated because of their gender, religion, nationality, age and so on, especially when those attributes are used for making decisions about them like giving them a job, loan, insurance, etc. The perception of discrimination, just like the perception of privacy, strongly depends on the legal and cultural conventions of a society.

The purpose of this chapter was to develop a new pre-processing discrimination prevention methodology including different data transformation methods that can prevent direct discrimination, indirect discrimination or both of them at the same time. To attain this objective, the first step is to measure discrimination and identify categories and groups of individuals that have been directly and/or indirectly discriminated in the decision-making processes; the second step is to transform data in the proper way to remove all those discriminatory biases. Finally, discrimination-free data models can be produced from the transformed dataset without seriously damaging data quality. The
experimental results reported demonstrate that the proposed techniques are quite successful in both goals of removing discrimination and preserving data quality. We show that indirect discrimination removal can help direct discrimination removal.
Chapter 7

A study on the impact of data anonymization on anti-discrimination

In this chapter, we study the relation between data anonymization and anti-discrimination (pre-processing approach). We analyze how different data anonymization techniques (e.g., generalization) have an impact on anti-discrimination (e.g., discrimination prevention). When we anonymize the original data to achieve the requirement of the privacy model (e.g., k-anonymity), what will happen to the discriminatory bias contained in the original data? Our main motivation to do this study is finding an answer for three important questions. First, can providing protection against privacy attacks also achieve anti-discrimination in data publishing? Second, can we adapt and use some of the data anonymization techniques (e.g., generalization) for discrimination prevention? Third, can we design methods based on data anonymization to make the original data protected against both privacy and discrimination risks?

7.1 Non-discrimination Model

Civil rights laws [9, 35, 101] explicitly identify the attributes to be protected against discrimination. For instance, U.S. federal laws [101] prohibit discrimination on the basis of race, color, religion, nationality, sex, marital status, age and pregnancy. In our context, we consider these attributes as potentially discriminatory (PD) attributes. Let $DA$ be a set of PD attributes in data table $D(A_1, \cdots, A_n)$ specified by law. Comparing privacy legislation [34] and anti-discrimination legislation [35, 101], PD attributes can overlap with QI attributes (e.g. Sex, Age, Marital_status) and/or sensitive attributes (e.g. Religion in some applications). A domain $DA_i$ is associated with each attribute $A_i$ to indicate the set of values that the attribute can assume. In previous works on anti-discrimination [77, 78, 86, 56, 43, 44, 55, 108], as we consider in previous chapters, the authors propose discrimination discovery and prevention techniques w.r.t. specific
protected groups, e.g. black and/or female people. However, this assumption fails to capture the various nuances of discrimination since minority or disadvantaged groups can be different in different contexts. For instance, in a neighborhood with almost all black people, whites are a minority and may be discriminated. Then we consider $A_i = q$ to be a PD item, for every $q \in D_{A_i}$, where $A_i \in DA$, e.g. $Race = q$ is a PD item for any race $q$, where $DA = \{Race\}$. This definition is also compatible with the law. For instance, the U.S. Equal Pay Act [101] states that: “a selection rate for any race, sex, or ethnic group which is less than four-fifths of the rate for the group with the highest rate will generally be regarded as evidence of adverse impact”. An item $A_i = q$ with $q \in D_{A_i}$ is a potentially non-discriminatory (PND) item if $A_i \notin DA$, e.g. $Hours = 35$ where $DA = \{Race\}$.

Building on Definition 30 and considering the above fact, we introduce the notion of $\alpha$-protection for a data table.

**Definition 34** ($\alpha$-protective data table). Let $D(A_1, \ldots, A_n)$ be a data table, $DA$ a set of PD attributes associated with it, and $f$ be one of the measures from Fig. 6.1. $D$ is said to satisfy $\alpha$-protection or to be $\alpha$-protective w.r.t. $DA$ and $f$ if each PD frequent classification rule $c : A, B \rightarrow C$ extracted from $D$ is $\alpha$-protective, where $A$ is a PD itemset and $B$ is a PND itemset.

Note that $\alpha$-protection in $D$ not only can prevent discrimination against the main protected groups w.r.t. $DA$ (e.g., women) but also against any subsets of protected groups w.r.t. $A \setminus DA$ (e.g., women who have a medium salary and/or work 36 hours per week). Releasing an $\alpha$-protective (unbiased) version of an original data table is desirable to prevent discrimination with respect to $DA$. If the original data table is biased w.r.t. $DA$, it must be modified before being published (i.e. pre-processed). The existing pre-processing discrimination prevention methods are based on data perturbation, either by modifying class attribute values [56, 43, 44, 62] or by modifying PD attribute values [43, 44] of the training data. One of the drawbacks of these techniques is that they cannot be applied (are not preferred) in countries where data perturbation is not legally accepted (preferred), while generalization is allowed; e.g. this is the case in Sweden and other Nordic countries (see p. 24 of [93]). Hence, we focus here on generalization and suppression.

### 7.2 Data Anonymization Techniques and Anti-discrimination

In this section, we study how different generalization and suppression schemes have impact on anti-discrimination. In other words, when we anonymize $D$ to achieve the requirement of the privacy model, e.g. $k$-anonymity, w.r.t. QI, what will happen to $\alpha$-protection of $D$ w.r.t. $DA$? The problem could be investigated with respect to different possible relations between PD attributes and other attributes (i.e. QI and sensitive attributes) in $D$. In this context, we consider the general case where all attributes are QI expect for the class/decision attribute. Then, each QI attribute can be PD or not. In summary, the following relations are assumed: (1) $QI \cap C = \emptyset$, (2) $DA \subseteq QI$. As mentioned in Section 7.1, PD attributes can overlap with QI and/or sensitive and/or
Table 7.1: Private data table with biased decision records

<table>
<thead>
<tr>
<th>ID</th>
<th>Sex</th>
<th>Job</th>
<th>Age</th>
<th>Credit_approved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>Engineer</td>
<td>35</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Male</td>
<td>Engineer</td>
<td>38</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>Male</td>
<td>Lawyer</td>
<td>38</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Female</td>
<td>Writer</td>
<td>30</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Male</td>
<td>Writer</td>
<td>30</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Female</td>
<td>Dancer</td>
<td>31</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Female</td>
<td>Dancer</td>
<td>32</td>
<td>Yes</td>
</tr>
</tbody>
</table>

non-sensitive attributes. Considering all attributes as QI such that $DA \subseteq QI$ can cover all the above cases.

**Example 10.** Table 7.1 presents raw customer credit data, where each record represents a customer’s specific information. $Sex$, $Job$, and $Age$ can be taken as QI attributes. The class attribute has two values, $Yes$ and $No$, to indicate whether the customer has received credit or not. Suppose the privacy model is $k$-anonymity and $k = 2$. Table 7.1 does not satisfy 2-anonymity w.r.t. $QI = \{Sex, Job, Age\}$.

**Example 11.** Continuing Example 10, suppose $DA = \{Sex\}$, $\alpha = 1.2$ and $f = \text{slift}$. Table 7.1 does not satisfy 1.2-protection w.r.t. $f$ and $DA$ since for frequent PD rule $c$ equal to $\{Sex = \text{female}\} \rightarrow Credit\_approved = \text{no}$ we have $\text{slift}(c) = \frac{2/3}{1/4} = 2.66$. Then Table 7.1 is biased w.r.t. women.

### 7.2.1 Global Recoding Generalizations and Anti-discrimination

In this section, by presenting different scenarios, we will show that using global recoding generalizations (i.e. full-domain generalization, subtree generalization and sibling generalization) to achieve $k$-anonymity w.r.t. QI in $DB$ can lead to different situations regarding the $\alpha$-protection of $D$ w.r.t. $DA$.

**Global recoding generalizations not offering $\alpha$-protection.** It can happen in different scenarios. First, consider a data table $D$ with the same attributes as the one in Table 7.1, but many more records, and let $DA = \{Job\}$ and $QI = \{Sex, Job, Age\}$. Suppose $D$ is biased with respect to dancers or a subgroup of dancers, e.g. dancers who are women (i.e. $D$ does not satisfy $\alpha$-protection w.r.t. $DA = \{Job\}$). Generalizing all instances of 30, 31 and 32 values to the the same generalized value $[30, 35]$ to achieve $k$-anonymity w.r.t. QI in $D$ using full-domain generalization, subtree or sibling generalization cannot achieve $\alpha$-protection w.r.t. $DA = \{Job\}$, based on Definition 34. Second, consider a data table $D$ with the same attributes as the one in Table 7.1, but many more records, and let $DA = \{Job\}$ and $QI = \{Sex, Job, Age\}$. Suppose $DB$ is biased with respect to dancers. Generalizing all instances of $Dancer$ and $Writer$ values to the same generalized value $Artist$ to achieve $k$-anonymity in $D$ w.r.t. QI using full-domain generalization or subtree generalization, might cause the $Artist$ node to inherit the biased nature of $Dancer$. Then, this generalization cannot
achieve \( \alpha \)-protection w.r.t. \( DA = \{ Job \} \). Third, consider a data table \( D \) with the same attributes as the one in Table 7.1, but many more records, and let \( DA = \{ Age \} \) and \( QI = \{ Sex, Job, Age \} \). Suppose \( D \) is not biased (i.e. \( D \) is \( \alpha \)-protective) with respect to \( DA \). It means that all PD frequent rules w.r.t. \( DA \) extracted from it are not \( \alpha \)-discriminatory. However, \( D \) might contain PD rules which are \( \alpha \)-discriminatory and not frequent, e.g. \( \{ Age = 30, Sex = Male \} \rightarrow Credit\_approved = no, \{ Age = 31, Sex = Male \} \rightarrow Credit\_approved = no, \{ Age = 32, Sex = Male \} \rightarrow Credit\_approved = no \). Generalizing all instances of 30, 31 and 32 values to the same generalized value \([30, 35]\) to achieve \( k \)-anonymity w.r.t. \( QI \) in \( D \) using full-domain generalization, subtree or sibling generalization, can cause new frequent PD rules to appear, which might be \( \alpha \)-discriminatory and discrimination will show up after generalization, e.g. \( \{ Age = [30 - 35], Sex = Male \} \rightarrow Credit\_approved = no \).

Global recoding generalizations offering \( \alpha \)-protection. Consider Table 7.1 and let \( DA = \{ Sex \} \) and \( QI = \{ Sex, Job, Age \} \). Suppose that Table 7.1 is biased with respect to women or any subgroup of women, e.g. women who are 30 years old and/or who are dancers (i.e. Table 7.1 does not satisfy \( \alpha \)-protection w.r.t. \( DA = \{ Sex \} \). Generalizing all instances of \( Female \) values to the same generalized value \( Any\_sex \) to achieve \( k \)-anonymity w.r.t. \( QI \) in Table 7.1 can also achieve \( \alpha \)-protection w.r.t. \( DA = \{ Sex \} \), based on Definition 34.

To summarize, using global recoding generalizations to achieve the requirement of the privacy model (i.e. \( k \)-anonymity), depending on the generalization, can make original data less more or less protected against discrimination.

### 7.2.2 Local Recoding Generalizations and Anti-discrimination

In this section, by analyzing different scenarios, we will show how using local recoding generalization, i.e. cell generalization, to achieve \( k \)-anonymity w.r.t. \( QI \) in \( D \), has an impact on \( \alpha \)-protection of \( D \) w.r.t. \( DA \). In contrast to global recoding generalizations, in cell generalization some instances of a value may remain ungeneralized while other instances are generalized.

Consider Table 7.1 and let \( DA = \{ Sex \} \) and \( \alpha = 1.2 \). Table 7.1 does not satisfy 1.2-protection w.r.t. \( f = slift \) and \( DA \), since for frequent PD rule \( c \) equal to \( \{ Sex = female \} \rightarrow credit\_approved = no \), by using the definitions of confidence and \( slift \) (Expressions (6.1) and (6.2), resp.), we have \( slift(c) = \frac{supp(A,C)}{supp(A)} = \frac{2}{10} = 0.2 \). Table 7.1 neither satisfies 1.2-protection w.r.t. \( f = elift \) and \( DA \), since for PD rule \( c \), by using the definitions of confidence and \( elift \) (Expressions (6.1) and (6.4), resp.), we have \( elift(c) = \frac{supp(A,C)}{supp(C)} = \frac{2/3}{37} = 1.25 \).

Generalizing some instances of \( Male \) and/or \( Female \) values to the same generalized value \( Any\_sex \) to achieve \( k \)-anonymity w.r.t. \( QI = \{ Job, Sex, Age \} \) in Table 7.1 using cell generalization can lead to different impacts on 1.2-protection of Table 7.1 w.r.t. \( DA = \{ Sex \} \). The impact depends on the value of class attribute (e.g. \( Yes \) or \( No \)) of each record in which the value of PD attribute (e.g. \( Female \) or \( Male \)) is generalized. Table 7.2 shows four types of cell generalization that can happen to achieve \( k \)-anonymity.
Table 7.2: Different types of cell generalization

<table>
<thead>
<tr>
<th>ID</th>
<th>Sex</th>
<th>Job</th>
<th>Age</th>
<th>Credit_approved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1) Male</td>
<td>Engineer</td>
<td>35</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Male</td>
<td>Engineer</td>
<td>38</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>(2) Male</td>
<td>Lawyer</td>
<td>38</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Female</td>
<td>Writer</td>
<td>30</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Male</td>
<td>Writer</td>
<td>30</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>(3) Female</td>
<td>Dancer</td>
<td>31</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>(4) Female</td>
<td>Dancer</td>
<td>32</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In Table 7.1 with numbers (1), (2), (3) and (4). Below, we analyze the impact of each type on 1.2-protection of Table 7.1 w.r.t. DA.

- Type (1). Generalizing an instance of Male value to the generalized value Any-sex while the value of Credit_approved attribute in the record is Yes cannot make Table 7.1 more or less 1.2-protective w.r.t. \( f = elift \) but it can make Table 7.1 more 1.2-protective w.r.t. \( f = slift \). It is because this type of cell generalization cannot change the value of \( elift(c) \) but it can decrease the value of \( slift(c) \), which is in this example \( slift(c) = \frac{2}{3/7} = 2 \). This type of cell generalization increases the denominator of equation \( \frac{\text{supp}(A,C)}{\text{supp}(A)} \) while keeping the numerator unaltered.

- Type (2). Generalizing an instance of Male value to the generalized value Any-sex while the value of Credit_approved attribute in the record is No cannot make Table 7.1 more or less 1.2-protective w.r.t. \( f = elift \) but it can make Table 7.1 less 1.2-protective w.r.t. \( f = slift \). It is because this type of cell generalization cannot change the value of \( elift(c) \) but it can increase the value of \( slift(c) \). This type of cell generalization decreases the denominator of equation \( \frac{\text{supp}(A,C)}{\text{supp}(A)} \) while keeping the numerator unaltered.

- Type (3). Generalizing an instance of Female value to the generalized value Any-sex while the value of Credit_approved attribute for the record is No can make Table 7.1 more 1.2-protective w.r.t. \( f = elift \) since it can decrease the value of \( elift(c) \), which is in this example \( elift(c) = \frac{1/2}{3/7} = 1.16 \). This type of cell generalization decreases the numerator of equation \( \frac{\text{supp}(A,C)}{\text{supp}(A)} \) while keeping the denominator unaltered. In addition, this generalization can also make Table 7.1 more 1.2-protective w.r.t. \( f = slift \) since it can decrease the value of \( slift(c) \), which is in this example \( slift(c) = \frac{1/2}{1/4} = 2 \). This type of cell generalization decreases the numerator of equation \( \frac{\text{supp}(A,C)}{\text{supp}(A)} \) while keeping the denominator unaltered.

- Type (4). Generalizing an instance of Female value to the generalized value Any-sex while the value of Credit_approved attribute for the record is Yes can make Table 7.1 less 1.2-protective w.r.t. both \( f = elift \) and \( f = slift \) since
it can increase the values of $elift(c)$ and $slift(c)$, which are in this example $elift(c) = \frac{2}{1/3} = 2.33$ and $slift(c) = \frac{2}{1/4} = 4$, respectively. This type of cell generalization increases the numerator of equations $\frac{supp(A,C)/supp(A)}{supp(C)/|DB|}$ and $\frac{supp(A,C)/supp(A)}{supp(\neg A,C)/supp(\neg A)}$, respectively, while keeping the denominators unaltered.

Summarizing, using cell generalization to achieve the requirement of privacy model (e.g. $k$-anonymity), depend on how many records in each above types modified, can make original data table less or more protected against discrimination. In addition, only the generalization of type (3) can make the original data table $\alpha$-protective w.r.t. both $f = elift$ and $f = slift$ if enough number of records are modified. We can conclude that although cell generalization leads to less data distortion than global recoding generalizations, it can have less positive impact on discrimination removal than global recoding generalizations.

### 7.2.3 Multidimensional Generalizations and Anti-discrimination

By presenting different scenarios, we also study the impact of using multidimensional generalizations to achieve $k$-anonymity w.r.t. QI in $D$ on $\alpha$-protection of $D$ w.r.t. $DA$ and we observe the similar trend as cell generalization. For the sake of brevity and due to similarity with Section 7.2.2, we do not recall the details here.

### 7.2.4 Suppression and Anti-discrimination

In this section, by presenting different scenarios, we will show that using suppression techniques (i.e. record suppression, value suppression and cell suppression) to achieve $k$-anonymity w.r.t. QI in $D$ can lead to different situations regarding the $\alpha$-protection of $D$ w.r.t. $DA$. As shown in Section 7.2.2, Table 7.1 does not satisfy 1.2-protection w.r.t. $DA = \{Sex\}$ and both $f = elift$ and $f = slift$, since for PD rule $c$ equal to $\{Sex = female\} \rightarrow credit\_ approved = no$ we have $slift(c) = 2.66$ and $elift(c) = 1.55$.

Suppressing an entire record to achieve $k$-anonymity in Table 7.1 w.r.t. $QI = \{Job, Sex, Age\}$ using record suppression can lead to different impacts on the 1.2-protection of Table 7.1 w.r.t. $DA = \{Sex\}$. The impact depends on the value of PD attribute (e.g. Female or Male) and the value of class attribute (e.g. Yes or No) in the suppressed record. Table 7.3 shows four types of record suppression which can happen to achieve $k$-anonymity w.r.t. $QI$ in Table 7.1 with numbers (1), (2), (3) and (4). Below, we analyze the impact of each type on $\alpha$-protection of Table 7.1 w.r.t. $DA$.

- **Type (1).** Suppressing an entire record with the value of Male in Sex attribute and the value of Yes in Credit\_approved attribute can make Table 7.1 more 1.2-protective w.r.t. $f = elift$ since it can decrease the value of $elift(c)$, which is in this example $elift(c) = \frac{2/3}{2/7} = 1.33$. This type of record suppression increases the denominator of equation $\frac{supp(A,C)/supp(A)}{supp(C)/|DB|}$ while keeping the numerator unaltered. In addition, this suppression can also make Table 7.1 more 1.2-protective w.r.t. $f = slift$ since it can decrease the value of $slift(c)$, which is in this example $slift(c) = \frac{2/3}{2/7} = 2$. This type of record suppression increases
Table 7.3: Different types of record suppression

<table>
<thead>
<tr>
<th>ID</th>
<th>Sex</th>
<th>Job</th>
<th>Age</th>
<th>Credit_approved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1) Male</td>
<td>Engineer</td>
<td>35</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Male</td>
<td>Engineer</td>
<td>38</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>(2) Male</td>
<td>Lawyer</td>
<td>38</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Female</td>
<td>Writer</td>
<td>30</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Male</td>
<td>Writer</td>
<td>30</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>(3) Female</td>
<td>Dancer</td>
<td>31</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>(4) Female</td>
<td>Dancer</td>
<td>32</td>
<td>Yes</td>
</tr>
</tbody>
</table>

the denominator of equation $\frac{\text{supp}(A,C)}{\text{supp}(A)}$ while keeping the numerator unaltered.

- Type (2). Suppressing an entire record with the value of Male in Sex attribute and the value of No in Credit_approved attribute can make Table 7.1 less 1.2-protective w.r.t. both $f = elift$ and $f = slift$ since it can increase the values of $elift(c)$ and $slift(c)$. This type of record suppression decreases the denominator of equations $\frac{\text{supp}(A,C)}{\text{supp}(A) \cdot |D|}$ and $\frac{\text{supp}(A,C)}{\text{supp}(A)}$, respectively, while keeping the numerators unaltered.

- Type (3). Suppressing an entire record with the value of Female in Sex attribute and the value of No in Credit_approved attribute cannot make Table 7.1 more or less 1.2-protective w.r.t. $f = elift$ since it cannot change the value of $elift(c)$ substantially, which is in this example $elift(c) = \frac{1/2}{2/6} = 1.5$. This happen because this type of record suppression decreases the numerator of equation $\frac{\text{supp}(A,C)}{\text{supp}(A) \cdot |D|}$ while also decreasing its denominator. However, this type of record suppression can make Table 7.1 more 1.2-protective w.r.t. $f = slift$ since it can decrease the value of $slift(c)$, which is in this example $slift(c) = \frac{1/2}{1/4} = 2$. This suppression decreases the numerator of $\frac{\text{supp}(A,C)}{\text{supp}(A)}$ while keeping the denominator unaltered.

- Type (4). Suppressing an entire record with the value of Female in Sex attribute and the value of Yes in Credit_approved attribute can make Table 7.1 less 1.2-protective w.r.t. both $f = elift$ and $f = slift$ since it can increase the value of $elift(c)$ and $slift(c)$, which are in this example $elift(c) = \frac{2/2}{2/6} = 2$ and $slift(c) = \frac{2/2}{1/4} = 4$, respectively.

To summarize, using record suppression depending on how many records in each of the above types suppressed can make original data table more or less protected against discrimination after achieving privacy protection. In addition, only record suppression of type (1) can make original data table $\alpha$-protective w.r.t. both $f = elift$ and $f = slift$ if a sufficient number of records are suppressed.

Value suppression refers to suppressing every instance of a given value in a data table. Then, depending on which attribute values are suppressed after achieving privacy
Table 7.4: Summary of results

<table>
<thead>
<tr>
<th>Data anonymization techniques</th>
<th>Achieve $\alpha$-protection</th>
<th>Against $\alpha$-protection</th>
<th>No impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global recoding generalizations</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cell generalization/Cell suppression Type (1)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cell generalization/Cell suppression Type (2)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cell generalization/Cell suppression Type (3)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cell generalization/Cell suppression Type (4)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Multidimensional generalization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Record suppression Type (1)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Record suppression Type (2)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Record suppression Type (3)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Record suppression Type (4)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Value suppression</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Value suppression can offer $\alpha$-protection or not. Cell suppression refers to suppressing some instances of a given value in a data table. Then, similarly to cell generalization, depending on the values of suppressed cells and the class values of respective records, cell suppression can make original data more or less protected against discrimination. Thus, similarly to cell generalization, suppression techniques have less positive impact on discrimination removal than global recoding generalizations. Finally, Table 7.4 summarizes the results we obtained in this chapter.

7.3 Conclusion

In this chapter, we have investigated the relation between data anonymization techniques and anti-discrimination to answer an important question: how privacy protection via data anonymization impacts the discriminatory bias contained in the original data. By presenting and analyzing different scenarios, we learn that we cannot protect original data against privacy attacks without taking into account anti-discrimination requirements (i.e. $\alpha$-protection). This happens because data anonymization techniques can work against anti-discrimination. In addition, we exploit the fact that some data anonymization techniques (e.g. specific full-domain generalization) can also protect data against discrimination. Thus, we can adapt and use some of these techniques for discrimination prevention. Moreover, by considering anti-discrimination requirements during anonymization, we can present solutions to generate privacy- and discrimination-protected datasets. We also find that global recoding generalizations have a more positive impact on discrimination removal than other data anonymization techniques.
Bibliography


PART 2

RECORD LINKAGE FOR RISK ASSESSMENT
Probabilistic Record Linkage for Disclosure Risk Assessment
Natalie Shlomo

1 Social Statistics, School of Social Sciences, University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom
Natalie.Shlomo@manchester.ac.uk

Abstract. Disclosure limitation methods for protecting the confidentiality of respondents in survey microdata often use perturbative techniques which introduce measurement error into the categorical identifying variables. In addition, the data itself will often have measurement errors commonly arising from survey processes. There is a need for valid and practical ways to assess the protection against the risk of identification for survey microdata with measurement errors. A common disclosure risk scenario is when an intruder seeks to match the microdata with an external file. We will examine probabilistic record linkage as a means of assessing disclosure risk and relate it to disclosure risk measures under the probabilistic framework of the Poisson log-linear models.

Keywords: Poisson-log linear model, Misclassification error, Sample unique, Disclosure risk-data utility map

1 Introduction

Statistical Agencies are obligated to protect the confidentiality of individuals when releasing sample microdata arising from social surveys. The risk assessment is typically based on a disclosure risk scenario where an ‘intruder’ attempts to link the sample microdata to available public data sources through a set of identifying key variables that are common to both sources. The identification of an individual could then be used to obtain sensitive information and the disclosure of attributes. In order to limit the risk of identification, the statistical agency will implement disclosure limitation methods on the sample microdata, the extent of which depend on the mode of release, such as on-site data labs, data archives or public-use files. Disclosure limitation methods can be non-perturbative where the information content is reduced without altering the data. These include deleting variables, sub-sampling or recoding and collapsing categories of variables. Perturbative disclosure limitation methods alter the data by introducing forms of misclassification. These include data swapping (Dalenius and Reiss, 1982, Gomatam, Karr and Sanil, 2005), noise addition (Kim, 1984, Fuller, 1993, Brand, 2002) and fully synthetic data where the data released is based on a statistical model (Raghunathan, Reiter, and Rubin, 2003). For more information on these methods see also: Willenborg and De Waal, 2001, Domingo-Ferrer and Torra, 2001.

Before releasing sample microdata, statistical agencies need to quantify the disclosure risk of identification. One method for assessing this risk is to simulate an ‘intruder’ attack by using probabilistic record linkage techniques. One of the first
examples was carried out in Spruill (1982) who linked perturbed sample microdata back to the original sample using distance based matching. In many studies of this type, a conservative assessment of the risk of identification is obtained since it assumes that the ‘intruder’ has access to the original dataset and does not take into account the protection afforded by the sampling. More recent examples use the probabilistic record linkage framework of Fellegi and Sunter (F&S) (1969) (see: Yancey, Winkler and Creecy, 2002, Hawala, Stinson and Abowd, 2005 and Torra, Abowd and Domingo-Ferrer, 2006). The identifying key variables used for matching are typically categorical, such as sex, date of birth, marital status and locality. In the F&S framework, each potential pair is assigned a matching weight as described in Section 2. The matching weights are sorted and appropriate cut-offs determined according to pre-specified type I and type II error bounds. Pairs with high matching weights are considered to be correct matches and pairs with low matching weights are considered to be correct non-matches. Pairs with matching weights between the cut-off thresholds undergo clerical review. The matching weights are proxies for the probability of a correct match given an agreement or disagreement. These probabilities can be used as individual record-level measures of disclosure risk. Global measures of disclosure risk include the proportion of correct matches, the proportion of correct matches to false matches, and one minus the estimated false match rate.

In Skinner, 2008, the probabilistic record linkage framework of F&S is linked to the probabilistic modelling framework for quantifying identification risk based on the notion of population uniqueness (see: Skinner and Holmes, 1998, Elamir and Skinner, 2006, Skinner and Shlomo, 2008). The probabilistic modelling framework relies on distributional assumptions to draw inference from the sample and estimate population parameters. The individual disclosure risk measure is the expectation of a correct match given a sample unique on the set of key variables. The global measure of disclosure risk is obtained by summing over the sample uniques to derive the expected number of correct matches. Shlomo and Skinner (2010) expanded the original probabilistic modelling framework to include misclassification errors in the key variables, either arising naturally through stages of data processing or purposely introduced into the data as a perturbative disclosure limitation method. In this paper we provide empirical evidence of the relationship between the probabilistic record linkage framework of F&S and the probabilistic modelling framework based on the notion of population uniqueness taking into account misclassification errors. We also show how a risk-utility assessment might be carried out by a statistical agency for choosing optimal parameters of record swapping.

In section 2 we introduce the notation and theory of the two frameworks for disclosure risk assessment: the F&S probabilistic record linkage framework and the probabilistic modelling framework. We also provide examples that link the two frameworks as set out in Skinner (2008). Section 3 presents an empirical study based on an extract from the UK 2001 Census. We first assume the perspective of the statistical agency where a perturbative method of disclosure limitation has been applied to the data and therefore misclassification probabilities and population parameters are known. We apply both the record linkage and probabilistic modelling framework for assessing the risk of identification and compare results. We also demonstrate how we can estimate population parameters through log-linear modelling in the probabilistic modelling framework when population counts are unknown, or use...
the EM algorithm to estimate matching parameters in the F&S record linkage framework when the match status is unknown. Section 4 demonstrates a risk-utility assessment using the example of random and targeted data swapping at different perturbation rates. We conclude in Section 5 with a discussion.

2 Notation and Theory

In this section we describe the F&S probabilistic record linkage framework and the probabilistic modelling framework based on the notion of population uniqueness and taking into account misclassification. We demonstrate the relationship between the two frameworks.

2.1 Fellegi and Sunter Probabilistic Record Linkage

Using the notation of Skinner, 2008, let \( \tilde{X}_a \) denote the value of the vector of cross-classified identifying key variables for unit \( a \) in the microdata \((a \in s_1)\) with values labelled 1, 2,..., \( J \). Let \( X_b \) the corresponding value for unit \( b \) in the external database \((b \in s_2)\). The different notation of \( X \) allows for different values of the two vectors due to natural misclassification in the data or an application of a perturbative disclosure limitation method to the sample microdata file. We assume that \( X_b \) of the external database is fixed and \( \tilde{X}_a \) of the microdata are determined by a \( J \times J \) probability misclassification matrix \( \theta \), where:

\[
P( \tilde{X}_a = k \mid X_a = j ) = \theta_{kj}
\]

\( \theta_{jj} \) on the diagonal of the matrix is the probability of not misclassifying (perturbing) category \( j \). Some examples of probability misclassification matrices when used for the purpose of perturbing microdata are presented in Willenborg and De Waal (2001).

Based on the F&S theory of record linkage, a comparison vector \( \gamma(\tilde{X}_a, X_b) \) is calculated for pairs of units \((a,b)\in s_1 \times s_2\), where the function \( \gamma(\cdot,\cdot) \) takes values in a finite comparison space \( \Gamma \). In the simplest case, \( \gamma(\tilde{X}_a, X_b) = 1 \) if there is an agreement on value \( j \) of \( \tilde{X}_a \) and \( X_b \), and 0 otherwise. For the disclosure risk scenario we assume that the intruder uses the comparison vector to identify pairs of units which contain the same unit \((a,a)\in s_1 \times s_2\). Typically the intruder will use a combination of exact matching and probabilistic matching by considering only pairs that are blocked through an exact match on some subset \( \tilde{s} \subseteq s_1 \times s_2 \). The intruder seeks to partition the set of pairs in \( \tilde{s} \) into a set of matches: \( M = \{ (a,b) \in \tilde{s} \mid a \in s_1, b \in s_2, a = b \} \) and non-matches: \( U = \{ (a,b) \in \tilde{s} \mid a \in s_1, b \in s_2, a \neq b \} \). The approach by F&S is to define the likelihood ratio \( m/u \) as the matching weight where \( m \) is the probability of an agreement given a match \( m = P( \gamma(\tilde{X}_a, X_b) \mid (a,b) \in M ) \) and \( u \) is the probability of
an agreement given not a match $u = P(\gamma(X_a, X_b) | (a, b) \in NM)$. The higher values of the likelihood ratio are more likely to belong to $M$ and the lower values of the likelihood ratio are more likely to belong to $U$. In addition, under the assumption of independence the $m$-probability and the $u$-probability can be split into individual components for each separate key variable. Let $p = P((a, b) \in M)$ the probability that the pair is in $M$. The probability of a correct match $p_{M^Y} = P((a, b) \in M | \gamma(X_a, X_b))$ can be calculated using Bayes Theorem:

$$p_{M^Y} = mp/[mp + u(1 - p)]. \quad (2)$$

If the match status is unknown, the matching parameters $m, u$ and $p$ can be estimated using the EM algorithm which is an iterative maximum likelihood estimation procedure for incomplete data (not shown here). Based on the estimation of the parameters, the probability of a correct match given an agreement $p_{M^Y}$ can be estimated by (2).

### 2.2 Probabilistic Modelling for Measuring Identification Risk

The probabilistic modelling framework for estimating the risk of identification is based on theory which uses models for categorical key variables. Let $f = \{f_j\}$ denote a multiway frequency table, which is a sample from a population table $F = \{F_j\}$, where $j = 1, 2, \ldots, J$ indicates a cell defined by cross-classified categorical key variables and $f_j$ and $F_j$ denote the frequency in the sample and in the population cell $j$, respectively. Denote by $n$ and $N$ the sample and population size, respectively and the number of cells by $J$. Disclosure risk arises from small cells, and in particular when $f_j = F_j = 1$ (sample and population uniques). We focus on a global disclosure risk measure based on sample uniques: $\tau = \sum_j I(f_j = 1)/F_j$. This measure is the expected number of correct matches if each sample unique is matched to a randomly chosen individual from the same population cell. We consider the case that $f$ is known, and $F$ is an unknown parameter and the quantity $\tau$ needs to be estimated. An estimate of $\tau$ is:

$$\hat{\tau} = \sum_j I(f_j = 1)\hat{E}[1/F_j | f_j = 1] \quad (3)$$

where $\hat{E}$ denotes an estimate of the expectation. The formula in (3) is naive in the sense that it ignores the possibility of misclassification. A common assumption in the frequency table literature is $F_j \sim \text{Poisson}(\lambda_j)$, independently, where $\sum_j F_j = N$ is a random parameter. Binomial (or Poisson) sampling from $F_j$ means that $f_j | F_j \sim \text{Bin}(F_j, \pi_j)$ independently, where $\pi_j$ is the sampling fraction in cell $j$. By standard calculations we then have:

$$f_j \sim \text{Poisson}(\lambda_j\pi_j) \text{ and, } F_j | f_j \sim f_j + \text{Poisson}(\lambda_j(1 - \pi_j)), \quad (4)$$

where $F_j | f_j$ are conditionally independent.
We use the approach as developed in Skinner and Holmes, 1998, Elamir and Skinner, 2006 and Skinner and Shlomo, 2008 by introducing log linear models to estimate population parameters and estimating the risk of identification. The sample counts \{f_j\} are used to fit a log-linear model: \[ \log \mu_j = x_j' \beta \] in order to obtain estimates for the parameters: \[ \hat{\lambda}_j = \hat{\mu}_j / \pi_j \]. Using the second part of (4), the expected individual disclosure risk measure for cell \( j \) is defined by:

\[
E_{\hat{\lambda}_j} (1/F_j | f_j = 1) = \left[ 1 - e^{-\hat{\lambda}_j (1-\pi)} \right] / \hat{\lambda}_j (1-\pi).
\] (5)

Plugging \( \hat{\lambda}_j \) for \( \lambda_j \) in (5) leads to the desired estimates \[ \hat{E}_{\hat{\lambda}_j} [1/F_j | f_j = 1] \] and then to \( \hat{\tau} \) of (3).

The original probabilistic modelling approach did not consider the case of misclassification naturally arising in surveys or purposely introduced into the data as a disclosure limitation method. Shlomo and Skinner (2010) define disclosure risk measures that take into account misclassification. The individual disclosure risk measures and the aggregated global disclosure risk measure on misclassified sample uniques in this case is defined as:

\[
\tau = \sum_{j} \left( \frac{1}{F_j} \log \left( \frac{\sum_{k} F_{jk} \theta_j / (1-\pi \theta_j) / (1-\pi \theta_j)}{\sum_{k} \sum_{j} F_{jk} \theta_j / (1-\pi \theta_j)} \right) \right)
\] (6)

and it follows that \[ \tau = \sum_{j} \log \left( \frac{1}{F_j} \right) \] with equality holding if there is no misclassification. The extent to which the left hand side of this inequality is less than the right hand side measures the impact of misclassification on disclosure risk.

If the sampling fraction is small we can approximate (6) by:

\[ \tau \approx \sum_{j} \log \left( \frac{1}{F_j} \right) \] Moreover, if the population size is large, we have approximately \[ \sum_{j} F_j \theta_j = \tilde{F}_j \], where \( \tilde{F}_j \) is the number of units in the population which would have \( \tilde{X}_j = j \) if they were included in the microdata (with misclassification). Hence a simple approximate expression for the global risk of identification, natural for many social surveys, is:

\[
\tau = \sum_{j} \log \left( \frac{1}{F_j} \right)
\] (7)

The approximations in (7) does not depend upon \( \theta_j \) for \( j \neq k \) and so knowledge of these probabilities is not required in the estimation of risk if ‘acceptable’ estimates of \( \theta_j \) (the diagonal probabilities of not-misclassification) and \( \tilde{F}_j \) are available. Similar to the case with no misclassification, the measure in (7) may be interpreted as the expected number of correct matches among sample uniques.

Since the values of \( F_j \) or \( \tilde{F}_j \) appearing in (6) and (7) are typically unknown, we need to estimate them. We do suppose that the values of \( \theta_j \) are known, especially in the case that a statistical agency purposely perturbs the data as a disclosure limitation method. Expression (7) provides a simple way to extend the log-linear modelling approach described above. Since the \( \tilde{F}_j \), \( j = 1, \ldots, J \) represent the available data, all that is required is to ignore the misclassification and estimate \( 1/F_j \) from the
\( \tilde{f}_j, \quad j = 1, \ldots, J \) by fitting a log-linear model to the \( \tilde{f}_j, \quad j = 1, \ldots, J \) following the same criteria as before. This results in an estimate for the individual disclosure risk measure \( \hat{E}(1 / \tilde{F}_j | \tilde{f}_j = 1) \) based on the assumptions of the Poisson distribution for the population and sample counts. These estimates should be multiplied by \( \theta_j \) values and summed if aggregate measures of the form in (7) are needed.

2.3 The Relationship Between the Two Frameworks

Skinner (2008) relates the F&S record linkage framework to the probabilistic modelling framework by providing the following examples:

Example 1: Assume no misclassification has occurred, i.e. \( \tilde{X}_a = X_a \) in both the population (P) and the sample (s) and that the true match status is known by the agency. Assume that sample (s) was drawn by simple random sampling from the population P. We calculate the contingency table in Table 1 for each \( X_a = j \) in the realized sample where the rows are a binary agreement/disagreement on the comparison vector: \( \gamma(X_a, X_a) \) for pairs \((a, b) \in s \times P\) and the columns the true match status.

From Table 1, we can calculate directly \( p_{\text{match}} = 1 / F_j \). We also obtain that the \( m \)-probability defined as the probability of an agreement given a match is \( f_j / n \), and the \( u \)-probability defined as the probability of an agreement given not a match is \( f_j (F_j - 1) / n(N - 1) \). The probability of a correct match is \( p = 1 / N \). Using Bayes formula:

\[
p_{\text{correct}} = \frac{1 / N \times f_j / n}{1 / N \times f_j / n + (1 - 1 / N) f_j (F_j - 1) / n(N - 1)} = \frac{1}{F_j}
\]

Small \( F_j \) therefore results in a high probability of a correct match given an agreement in the comparison vector.

Table 1: Contingency table of binary agreement status and match status for \( X_a = j \) with no misclassification

<table>
<thead>
<tr>
<th></th>
<th>Non-match</th>
<th>Match</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disagree</td>
<td>( n(N - 1) - f_j (F_j - 1) )</td>
<td>( n - f_j )</td>
<td>( Nn - f_j F_j )</td>
</tr>
<tr>
<td>Agree</td>
<td>( f_j (F_j - 1) )</td>
<td>( f_j )</td>
<td>( f_j F_j )</td>
</tr>
<tr>
<td>Total</td>
<td>( n(N - 1) )</td>
<td>( n )</td>
<td>( Nn )</td>
</tr>
</tbody>
</table>

Example 2: In continuation of Example 1, assume now that the microdata has undergone misclassification (either as a result of errors or purposely perturbed for disclosure limitation). Denote \( \tilde{f}_j \) the observed misclassified sample counts with \( \tilde{X}_a = j \) derived by \( \tilde{f}_j = \theta_j f_j + \sum \theta_k f_k \). We calculate the contingency table on the
realized misclassified sample in Table 2 for \( \tilde{X}_a = j \) where the rows are a binary agreement/disagreement on the comparison vector: \( \gamma(\tilde{X}_a, X_a) \) for pairs \((a,b) \in s \times P\) and the columns the true match status.

Table 2: Contingency table of binary agreement status and match status for \( \tilde{X}_a = j \) with misclassification

<table>
<thead>
<tr>
<th></th>
<th>Non-match</th>
<th>Match</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disagree</td>
<td>( Nn - n - \tilde{j} F_j + \theta_0 f_j )</td>
<td>( n - \theta_0 f_j )</td>
<td>( Nn - \tilde{j} F_j )</td>
</tr>
<tr>
<td>Agree</td>
<td>( \tilde{j} F_j - \theta_0 f_j )</td>
<td>( \theta_0 f_j )</td>
<td>( \tilde{j} F_j )</td>
</tr>
<tr>
<td>Total</td>
<td>( Nn - n )</td>
<td>( n )</td>
<td>( Nn )</td>
</tr>
</tbody>
</table>

From Table 2, we can calculate directly

\[
p_{uv} = \frac{\theta_0 f_j}{\tilde{j} F_j} = \frac{\theta_0 / \bar{i} F_j}{\theta_0 / \tilde{i} F_j} = \frac{\theta_0}{\bar{i} F_j} = \frac{\theta_0}{\tilde{i} F_j}
\]

where \( \tilde{F}_j \) is the number of units in the population (P) with \( \tilde{X}_a = j \) (imagining that the misclassification takes place before the sampling). We also obtain the \( m \)-probability (the probability of an agreement given a match) of \( \theta_0 / f_j \) and the \( u \)-probability (the probability of an agreement given not a match) of \( (\tilde{j} F_j - \theta_0 f_j) / n(N-1) \). The probability of a correct match is \( p = 1/N \). Using Bayes formula:

\[
p_{uv} = \frac{1/N \times \theta_0 f_j / n}{1/N \times \theta_0 f_j / n + (1 - 1/N)(\tilde{j} F_j - \theta_0 f_j) / n(N-1)} = \frac{\theta_0 / \tilde{i} F_j}{\theta_0 / \tilde{j} F_j} = \frac{\theta_0}{\tilde{i} F_j} = \frac{\theta_0}{\tilde{j} F_j}
\]

Expression (9) is similar to the per-record individual risk measures used to define (7). Skinner (2008) also shows that the derivation of the probability of a correct match given an agreement holds for any subset of the population which may be selected arbitrarily.

3 Empirical Study

In this section, we provide empirical evidence based on real data of the connection between probabilistic record linkage according to F&S and the probabilistic modelling framework for calculating the risk of identification. We start from the perspective of the statistical agency where it is assumed that the misclassification matrix is known either because the data was purposely perturbed by the agency for disclosure limitation or a study was carried out to assess error rates in various stages of the data processing. We begin with assuming that population counts are known and hence the agency can calculate the necessary parameters to measure identification risk in both frameworks for this comparison. We also consider the case where population counts are unknown and examine the proximity of estimated individual per-record disclosure risk measures to true disclosure risk measures in both frameworks.
3.1 Preparation of the Data

We use the method of data swapping on an extract of individuals from the 2001 UK Census to compare the F&S framework and the probabilistic modelling framework. The population includes N=1,468,255 individuals and we draw a 1% simple random sample without replacement (n=14,683). There are six key variables for the risk assessment: Local Authority (LAD) (11), sex (2), age group (24), marital status (6), ethnicity (17) and economic activity (10) where the numbers of categories of each variable are in parenthesis (J=538,560). There are 2,873 sample uniques under the cross-classified key variables. We implement a random data swap by drawing a 20% sub-sample in each of the LADs. In each of the sub-samples, half of the individuals are flagged. For each flagged individual, an unflagged individual is randomly chosen within the sub-sample and their LAD variables swapped, on condition that the individual chosen was not previously selected for swapping and that the two individuals do not have the same LAD, i.e. no individual is selected twice for producing a swapping pair.

The misclassification matrix $\theta$ for the data swapping design of LAD can be expressed in terms of the 11 by 11 misclassification matrix defined by:

1. On the diagonal: $\theta_{jj} = 0.8$

2. Off the diagonal: $\theta_{jl} = 0.2\left[\frac{1}{\sum_{i=1}^{n_k} n_k} \right]$ where $n_k$ is the number of records in the sample in LAD k.

The number of sample uniques on the misclassified sample is 2,997.

3.2 Identification Risk Based on Probabilistic Frameworks and Modelling

Since we know the misclassification matrix $\theta$ and the true population counts $F_j$ in this study, we can compare the naïve risk measure in (3) and under misclassification in (6) based on the probabilistic modelling framework. Table 3 presents global disclosure risk measures for our sample, which are obtained by summing individual per-record risk measures across sample uniques. The first row of Table 3 shows the true disclosure risk $\tau$ in terms of the expected number of correct matches in the data before the misclassification. The second row in Table 3 contains the true disclosure risk $\hat{\tau}$ in (6) taking into account the misclassification and the third row the estimated disclosure risk measure under misclassification $\hat{\tau}$ defined by summing $\hat{\theta}_y\hat{E}(1/\hat{F}_j \mid \hat{F}_j = 1)$ across sample uniques. As can be seen, the estimation of the global disclosure risk measure follows closely the true disclosure risk measure (see Skinner and Shlomo, 2008 for a discussion on model selection and goodness of fit criteria for estimating the risk of identification using log-linear modelling).

The individual per-record risk measures for sample uniques as shown in (6) are more difficult to estimate accurately by estimates: $\theta_y\hat{E}(1/\hat{F}_j \mid \hat{F}_j = 1)$. Figure 1 compares the individual per-record estimated risk measures $\theta_y\hat{E}(1/\hat{F}_j \mid \hat{F}_j = 1)$ on
the X-axis with the individual risk measure from (6) assuming known population counts on the Y-axis. The figure is presented on the logarithmic scale.

Table 3: Global risk measures on sample uniques for the 20% random data swap in the probabilistic modelling framework

<table>
<thead>
<tr>
<th>Global Risk Measure</th>
<th>Expected correct matches out of sample uniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>True risk measure τ in original sample</td>
<td>363.0</td>
</tr>
<tr>
<td>Risk measure (6) under misclassification ( \tilde{\tau} )</td>
<td>298.9</td>
</tr>
<tr>
<td>Estimated risk measure under misclassification ( \hat{\tau} )</td>
<td>307.7</td>
</tr>
</tbody>
</table>

Figure 1 confirms that on average, the global disclosure risk measure in (6) is estimated accurately with the graph being symmetrical about the equality diagonal. The individual per-record risk measures however vary and their estimation is less accurate. From the perspective of an intruder who might use log-linear modelling to identify high risk individuals, it would be difficult to ascertain exactly which of the individuals are population uniques.

We turn now to the F&S probabilistic record linkage framework. For our record linkage experiment we block on all key variables that match exactly and calculate the probability of a correct match given an agreement on the perturbed LAD in each block. We focus only on sample uniques in order to compare to the probabilistic modelling framework. All possible pairs between the population dataset and the 2,997 perturbed sample uniques after blocking on all key variables except for LAD results in 1,600,685 possible pairs. Table 4 presents the counts of these pairs under this blocking strategy according to the true match status and the agreement/disagreement indicator on LAD.

From Table 4, the \( m \)-probability is 0.78, the \( u \)-probability is 0.09 and the probability of a correct match is \( p = 0.002 \). Note that the \( m \)-probability is the same as the overall non-misclassification rate (the diagonal of the misclassification matrix \( \Theta \)).
The u-probability represents the proportion of random agreements on LAD (1 out of 11). On average, the probability of a correct match given an agreement on LAD is: $p_{M|Y} = 0.016$ or 1.6%.

Table 4: Frequency counts of pairs blocked on agreeing key variables according to agreement/disagreement on LAD and the true match status

<table>
<thead>
<tr>
<th></th>
<th>Non-matches</th>
<th>Match</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disagree LAD</td>
<td>1,450,677</td>
<td>659</td>
<td>1,451,336</td>
</tr>
<tr>
<td>Agree LAD</td>
<td>147,011</td>
<td>2,338</td>
<td>149,349</td>
</tr>
<tr>
<td>Total</td>
<td>1,597,688</td>
<td>2,997</td>
<td>1,600,685</td>
</tr>
</tbody>
</table>

To assess the probability of a correct match given an agreement $p_{M|Y}$ for each individual sample unique, we implement the probabilistic record linkage separately in each block defined by cross-classifying non-perturbed key variables. Summing the probabilities of a correct match given an agreement $p_{M|Y}$ over the sample uniques, we obtain the global disclosure risk measure of 289.5 which is similar to the disclosure risk measures in Table 3.

From the analysis in Table 2, we expect that the probabilities $p_{M|Y}$ from the record linkage should be similar to those obtained as the individual per-record risk measure shown in (6) under the probabilistic modelling for those agreeing on LAD. This comparison is presented in Figure 2 for both the individual disclosure risk measure shown in (6) assuming known population counts (on the left side) and the estimated disclosure risk measures $\theta_j\hat{E}(1/\tilde{F}_j|\tilde{f}_j=1)$ (on the right side). The individual disclosure risk measures shown in (6) follow closely the probabilities of a correct match given an agreement $p_{M|Y}$ from the F&S framework. In addition, the estimated disclosure risk measures also follow $p_{M|Y}$ but with more variance.

Figure 2: Plot of $p_{M|Y}$ against individual disclosure risk measures shown in (6) (left) and the estimated disclosure risk measure $\theta_j\hat{E}(1/\tilde{F}_j|\tilde{f}_j=1)$ (right) for sample uniques agreeing on LAD (logarithmic scale)
Turning to the estimation of \( p_{\gamma M} \) in the F&S framework, we demonstrate results from the EM algorithm using data from one particular block as shown in Table 5.

**Table 5: Frequency counts of pairs within a single block of agreeing key variables according to agreement/disagreement on LAD and true match status**

<table>
<thead>
<tr>
<th></th>
<th>Non-match</th>
<th>Match</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disagree LAD</td>
<td>916</td>
<td>1</td>
<td>917</td>
</tr>
<tr>
<td>Agree LAD</td>
<td>100</td>
<td>3</td>
<td>103</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1,016</strong></td>
<td><strong>4</strong></td>
<td><strong>1,020</strong></td>
</tr>
</tbody>
</table>

From Table 5, the true parameters for this particular block are: \( m \)-probability is 0.750, \( u \)-probability is 0.098, the probability of a correct match is 0.0039 and 
\[
\gamma = \frac{2}{3p_{\gamma M}} = 0.029.
\]

We initiate the EM algorithm with an \( m \)-probability of 0.78, \( u \)-probability of 0.78 and the probability of a correct match 0.002. Convergence in the EM algorithm means that the sum of the squared change of estimates of the \( m \) and \( u \)-probabilities between iterations is less than 0.0000001. The estimation of the EM algorithm resulted in: \( \hat{m} = 0.798 \), \( \hat{u} = 0.099 \), and \( \hat{p} = 0.0022 \). From here, we obtain:

\[
\gamma = \frac{2}{3\hat{p}_{\gamma M}} = \frac{0.0022(0.798)}{0.0022(0.798) + (1 - 0.0022)(0.099)} = 0.017.
\]

As can be seen, it is difficult to estimate the parameters exactly using the EM algorithm. Generally, the EM algorithm will estimate parameters more accurately when there is a large number of pairs and a relatively large number of correct matches (approximately over 5%).

### 4 Risk-Utility Framework for Assessing Disclosure Limitation Methods

We demonstrate how statistical agencies might assess a disclosure limitation method and its parameters. We focus on random and targeted data swapping with varying rates of perturbation: 2%, 5%, 10% and 20%. Given a measure of information loss, we plot a disclosure risk-data utility map (Duncan, et al. 2001). For disclosure risk, we use the global disclosure risk measure defined in (6). Information loss will be measured by the relative absolute average distance per cell defined as follows:

Let \( D \) represent a frequency distribution for a two-way table defined by LAD and Ethnicity and let \( D(r,c) \) be the frequency in the cell in row \( r \), \( r=1,..,R \) and column \( c \), \( c=1,...,C \). The distance metric is:

\[
RAAD(D_{\text{pert}}, D_{\text{orig}}) = 1 - \frac{\sum_{r,c} D_{\text{pert}}(r,c) - D_{\text{orig}}(r,c)}{\sum_{r,c} D_{\text{orig}}(r,c)} (10)
\]

with \( \text{pert} \) and \( \text{orig} \) referring to the perturbed and original table respectively. The RAAD provides a measure of the average absolute perturbation per cell compared to the average cell size of the original table.

Figure 3 contains the disclosure risk-data utility map. The points on the map represent different candidate releases of record swapping. The points are denoted by \( T \) for targeted or \( R \) for random; and 20 for 20%, 10 for 10%, 5 for 5% or 2 for 2%. The points are plotted against the risk measure \( \tau_\theta \) in (6) on the Y-Axis and the
information loss measure $RAAD$ in (10) for LAD*Ethnicity on the X-Axis. We see that for a given level of data utility, we need approximately half of the level of perturbation under the targeted record swapping compared to the random record swapping. Depending on the tolerable risk threshold determined by the agency, the optimal method of record swapping and the swap rate is the one found on the frontier of the disclosure risk-data utility map represented by the connecting line (Gomatam, et al., 2005). This finding could vary in other settings and an agency could use a similar disclosure risk-data utility approach, based on its own data, to determine the preferred disclosure limitation approach.

**Figure 3: Disclosure risk-data utility map for record swapping (T – targeted, R – random and 2, 5, 10 and 20 swap rates) for table LAD*Ethnicity**

![Disclosure risk-data utility map for record swapping](image)

**5 Discussion**

In this paper, we have provided empirical evidence of the connection between the F&S record linkage framework to the probabilistic modelling framework for estimating the risk of identification based on the notion of population uniqueness as discussed in Skinner (2008). We have seen that statistical agencies are able to estimate accurate global disclosure risk measures that can be used to assess optimal disclosure limitation methods through a disclosure risk-data utility framework assuming that the probability of not being misclassified or perturbed is known, even if there is no population data available.

Individual per-record disclosure risk measures are more difficult to estimate without knowing true population counts in both frameworks. The estimation is carried out through the log linear modelling for the probabilistic modelling framework or the EM algorithm for the F&S record linkage framework. The results show that from the perspective of the ‘intruder’, it is difficult to identify high risk sample uniques, due to the variability of the estimation of the risk measures.

**Acknowledgements.** The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement nº 262608 (DwB - Data without Boundaries).
REFERENCES


The research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 262608 (DwB - Data without Boundaries).