An Integer Optimization Approach to k-Anonymity on Nominal Data

Vom Fachbereich IV der Universität Trier
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation

von

Vera Charlotte Cost

Trier, 2023

Betreuer:	Prof. Dr. Sven de Vries
	Prof. Dr. Ralf Münnich
1. Berichterstatter:	Prof. Dr. Sven de Vries
2. Berichterstatter:	Prof. Dr. Ralf Münnich
Datum der Disputation:	31.03.2023

Academic Career of the Author

08/2019 – 10/2022	**Mitglied des Graduiertenkollegs „Algorithmic Optimization“**
07/2019	**Master of Science Angewandte Mathematik**
10/2017 – 07/2019	Studium der Angewandten Mathematik an der Universität Trier
09/2017	**Bachelor of Science Mathematik**
10/2014 – 09/2017	Studium der Mathematik an der Georg-August-Universität Göttingen
04/2014 – 09/2014	Studium der Wirtschaftsmathematik an der Philipps-Universität Marburg
06/2013	**Abitur**
08/2005 – 06/2013	Kaiserin-Friedrich-Gymnasium in Bad Homburg vor der Höhe

Summary

The publication of statistical databases is regulated by law, e.g. national statistical offices are only allowed to publish data if the data cannot be attributed to individuals. However, the information loss due to anonymization should be kept minimal. In this thesis, we analyze the anonymization method used in the German census in 2011 and we propose a new anonymization method for nominal data including integer programming. The proposed method replaces data rows in a microdata set with representative values, such that k-anonymity is satisfied, i.e. each data row is identical to at least $k-1$ other rows. This new method is particularly suited for nominal data because it includes next to the overall dissimilarities of the data rows also errors in resulting frequency tables, which are of high interest for nominal data in practice.

First, we study the anonymization method SAFE, which was used in the last German census in 2011. The underlying SAFE-basic problem aims to minimize the maximum error in a subset of frequency tables. We prove that a fundamental variant of the SAFE-basic problem is $\mathcal{NP}$-hard already, which justifies the use of heuristic approaches for larger data sets. Moreover, we find that SAFE takes additional criteria into account like the mean error in frequency tables or the deviation from the original microdata set, which are not addressed in the SAFE-basic problem. Therefore, we propose a modification of the SAFE-basic problem. Our findings and the strength of this proposed modification are underlined by our experimental results, in which SAFE and the modified variant outperform the SAFE-basic problem with respect to maximum absolute and relative errors and χ^2-errors.

In the second part of this thesis, we propose a new method for microaggregation, which is tailored to nominal data. This method follows a typical two-step structure of first partitioning the data set in clusters with at least k elements, which are as similar as possible to each other, and then replacing all cluster elements with representative values to obtain k-anonymity. For the partitioning step, we point out that in contrast to numerical data, for nominal data, it is not sufficient to minimize distances between cluster elements and potential representatives but the distances between each pair of cluster elements have to be taken into account to find clusters with minimum dissimilarities. The proposed method respects these findings and is based on an in-

teger program with variables for all clusters with k to $2k-1$ elements, which can be formulated as

$$\begin{aligned} \min\nolimits_{x\in\{0,1\}^{\mathcal{C}}} \quad & \sum_{C\in\mathcal{C}} w_C x_C \\ \text{s.t.} \quad & \sum_{C\in\mathcal{C}} a_{iC} x_C = 1 \quad \forall i \in \mathcal{I}, \end{aligned}$$

where $\mathcal{I}$ is the index set of the data rows, $\mathcal{C} \subset 2^{\mathcal{I}}$ the set of possible clusters, w_C the sum of Hamming-distances between each pair of data points in cluster C and $a_{iC} \in \{0,1\}$ indicates whether data row i is contained in cluster C. The linear relaxation of this problem is suitable for a column generation scheme and leads to a fractional solution. Based on the dual information provided by a fractional solution, we construct an integer solution to the partitioning problem above. The basic idea is to add clusters iteratively until each data row is assigned to exactly one cluster. The added cluster is chosen, such that it minimizes an auxiliary problem and contains only not yet assigned data rows. This auxiliary problem is motivated by the minimization of the duality gap and includes two parameters. The first parameter weights the dual information and the second parameter rewards clusters of larger size. Our computational experiments show that the inclusion of dual information for finding an integer solution is beneficial but should not be overemphasized. Similarly, the size of the clusters should not dominate the similarity of cluster elements. Furthermore, our experiments demonstrate that the better the quality of a solution to the partitioning problem the lower the information loss after anonymization.

For the aggregation step, we present a mixed-integer problem formulation to find cluster representatives. To this end, we take errors in a subset of frequency tables into account because frequency tables are of particular interest for nominal data in practice. Since all clusters contribute jointly to the entries of the frequency tables, the selection of a cluster representative depends on all clusters for nominal data, which differs from microaggregation of numerical data. The objective of the proposed formulation is the minimization of the sum of χ^2-errors in frequency tables while staying as close as possible to the original microdata set. We only allow cluster elements as representatives instead of artificial combinations to prevent unreasonable combinations. Furthermore, we reformulate the problem to a minimum edge-weighted maximal clique problem in a multipartite graph. With this reformulation, we allow for a different view angle, which is useful to design heuristic approaches. We propose a greedy algorithm and a clique heuristic based on the reformulation to approach the aggregation problem. While the greedy algorithm requires shorter runtimes, the performance of the clique heuristic shows its strength if the partitioning has a high quality. Moreover, we formulate a mixed-integer program, which combines the partitioning and the aggregation step and aims to minimize the sum of χ^2-errors in frequency tables.

Our experimental study presents particularly strong results of the proposed method with respect to relative criteria, the maximum relative error and the sum of χ^2-errors, in most cases. With respect to the maximum absolute error, the SAFE method shows its strength. In regression analysis, SAFE and the proposed method are competitive with the exact solving of the formulated mixed-integer program, which combines both steps.

We conclude that the inclusion of integer programming in the context of data anonymization is a promising direction to reduce the inevitable information loss through anonymization particularly for nominal data.

Contents

1. Introduction **1**

1.1. Mathematical Background and Notation 1

1.2. Literature Review 5

1.2.1. Different Anonymization Approaches 5

1.2.2. Clustering and Microaggregation 8

1.2.3. Column Generation 12

1.3. Scope of This Thesis 13

2. SAFE Method **17**

2.1. Problem Description and Formulation 17

2.2. Outline of the Algorithm 19

3. Analysis of the SAFE Method **25**

3.1. Complexity Analysis 25

3.2. Discussion of the SAFE Method 29

3.3. Summary 32

4. Partitioning Step **33**

4.1. Motivation for Clustering-based Partitioning 35

4.2. Problem Formulation 36

4.3. Finding a Fractional Solution 38

4.4. Finding an Integer Solution 42

4.4.1. Preliminaries 43

4.4.2. Application to the Partitioning Problem 46

4.5. Summary of the Proposed Partitioning Step 49

5. Aggregation Step **51**

5.1. Introduction 51

5.2. Formulation of the Aggregation Problem 55

5.3. Reformulation to a Maximal Clique Problem in a Multipartite Graph 56

5.4. Heuristic Approaches 59
5.4.1. Greedy Algorithm 60
5.4.2. Graph-based Heuristic 60
5.5. Summary of the Proposed Aggregation Step 62
5.6. Exact Method Combining Both Steps 62

6. Computational Experiments 65
6.1. Analysis Criteria 65
6.2. Computational Experiments 66
6.2.1. Data Collection 67
6.2.2. Hard- and Software 69
6.2.3. Parameter Settings in the Partitioning Step 69
6.2.4. Quality of the Proposed Partitioning 73
6.2.5. Quality of the Proposed Aggregation Methods 76
6.2.6. Effect of Control Tables 82
6.2.7. Comparison of Different Methods 83
6.2.8. Logistic Regression 93
6.2.9. Increasing the Size of the Instances 95
6.3. Summary 97

7. Conclusion 101

A. Parameter Selection for the SAFE Method 105

B. Further Computational Results 107

Bibliography 115

Chapter 1

Introduction

National statistical offices are required to protect privacy when publishing data. Typically, these confidentiality rules are regulated by law. For example, in Germany, §16 ii BStatG states that data for federal statistics can only be released if they cannot be attributed to individuals. The subject of dealing with data anonymization is called *Statistical Disclosure Control* (SDC). Anonymization methods face two challenges. On the one hand, they must ensure the confidentiality for individual data. On the other hand, the information loss should be kept minimal to preserve data utility.

In general, anonymization is required for different types of data, e.g. dynamic data sets and static data sets. Here, we focus on census data, which are collected once every ten years and, thus, are static data. For a census, categorical data are of high interest and, typically, frequency tables are published that count the number of individuals, which meet certain properties.

This thesis addresses the question, how privacy-preserving data can be constructed using Integer Optimization methods to minimize the information loss. The focus is on nominal data and the resulting errors in frequency tables, which are of high interest for the practical use of the data.

1.1. Mathematical Background and Notation

A *microdata set* is a collection of data records, which correspond to individual statistical objects, e.g. people or households. These records are collected, for example, in the census and contain various attributes. An attribute is a characteristic or feature, e.g. ZIP code, which can take different values, e.g. 54296. Survey attributes can be divided into *identifiers*, *quasi-identifiers*, *confidential attributes* and *non-confidential attributes*. Identifiers uniquely identify an individual statistical object, e.g. personal ID number. In a first step of anonymization, these identifiers are deleted from a microdata set. In consequence, we assume that the given microdata set does not contain any identifiers. Quasi-identifiers are attributes, which can be used together to disclose identity, e.g. ZIP code, date of birth, sex. Confidential attributes contain sensitive information, e.g. medical diagnosis. Non-confidential attributes contain non-sensitive information and are therefore considered to not need protection. This thesis focuses on the handling of quasi-identifiers. To ensure privacy, it is necessary to ensure that quasi-identifiers cannot be used for re-identification.

Alternatively, survey attributes can be divided into *numerical* (quantitative) and *categorical* (qualitative) attributes. Categorical attributes are variables with a finite set of values. They can be further divided into *nominal* and *ordinal* attributes. Nominal data consist of unordered categories, while ordinal data include categories with an intrinsic order or ranking. In our research, we focus on nominal attributes, which do not have any natural order in contrast to ordinal attributes.

We denote the set of nominal attributes by $\mathcal{J} = \{1, \dots, m\}$. Since categorical attributes have a finite set of values, we can represent the values by natural numbers. We denote by $V(j) \subset \mathbb{N}$ the *domain* of attribute $j \in \mathcal{J}$, i.e. a finite set of unique attribute values. We define $V := \bigtimes_{j \in \mathcal{J}} V(j) = V(1) \times \cdots \times V(m)$ to be the set of all value combinations. Then, a data vector corresponding to a statistical object can be represented as an m-tuple drawn from V. Later, we will often refer to a statistical object by its value combinations and use the set notation to group objects. While technical correctness would require the addition of unique identifiers to the tuples for element distinctiveness, we choose to simplify the notation for better readability by omitting these identifiers. We denote the set of statistical objects, e.g. people or households, by $\mathcal{I} = \{1, \dots, n\}$.

We will use two different data representations. First, we use the common form of a *microdata table.* A microdata table $Y^0 \in V^n$ is a table, where each row corresponds to a statistical object and each column corresponds to an attribute. An entry $Y^0_{ij} \in V(j)$ equals the value of statistical object $i \in \mathcal{I}$ for attribute $j \in \mathcal{J}$. We denote the number of unique combinations of attribute values by $\tilde{n} = \prod_{j \in \mathcal{J}} |V(j)|$. Secondly, we also use a *frequency vector* $y^0 \in \mathbb{N}_0^{\tilde{n}}$. A frequency vector shows the frequency of occurrence of a combination of attribute values in the data. An entry y^0_l equals the number of occurrences of combination $l = (v_1, \dots, v_m) \in V$. The microdata set can also be represented by a frequency vector only containing the non-zero entries of y^0, if support is specified.

In the following, we illustrate the notations with an example.

Example 1.1.1. We consider a survey, which asks $n = 5$ people about the attributes $\mathcal{J} = \{\textit{sex}, \textit{citizenship}, \textit{working status}\}$. Let the domains of the attributes be $V(\textit{sex}) = \{\text{female}, \text{male}\}$, $V(\textit{citizenship}) = \{\text{Germany}, \text{Others}\}$, and $V(\textit{working status}) = \{$employed, unemployed, inactive$\}$. In Figure 1.1 we show an exemplary data set in the described representations, a microdata table (1.1a), a frequency vector (1.1b) and a frequency vector in abbreviated form (1.1c). The non-abbreviated frequency vector consists of $\tilde{n} = 12 = 2 \cdot 2 \cdot 3$ entries, which equals the number of unique value combinations.

Sex	*Citizenship*	*Working status*
female	Germany	employed
female	Germany	inactive
female	Germany	employed
male	Others	inactive
male	Germany	unemployed

(a) Microdata table Y^0.

$$\begin{array}{l} \left(\begin{array}{c} 2 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{array}\right) \end{array} \begin{array}{l} \text{female,Germany,employed} \\ \text{female,Germany,unemployed} \\ \text{female,Germany,inactive} \\ \text{female,Others,employed} \\ \text{female,Others,unemployed} \\ \text{female,Others,inactive} \\ \text{male,Germany,employed} \\ \text{male,Germany,unemployed} \\ \text{male,Germany,inactive} \\ \text{male,Others,employed} \\ \text{male,Others,unemployed} \\ \text{male,Others,inactive} \end{array}$$

(b) Frequency vector y^0.

$$\left(\begin{array}{c} 2 \\ 1 \\ 1 \\ 1 \end{array}\right) \begin{array}{l} \text{f,G,e} \\ \text{f,G,i} \\ \text{m,G,u} \\ \text{m,O,i} \end{array}$$

(c) Abbreviated frequency vector.

Figure 1.1.: Forms of microdata presentation.

A commonly used concept to define anonymous data is given by *k-anonymity*, which was proposed by Sweeney (2002). The basic idea is to prohibit rare or even unique data rows and edit the microdata set such that each existing combination of attributes occurs at least k times in the microdata set. Unique or very rare attribute combinations are considered most at risk because they might be linked to individuals and therefore violate privacy. The main advantage of k-anonymity is that these combinations are prevented. The property of k-anonymity is defined as follows.

Definition 1.1.2 (k-Anonymity, Sweeney, 2002)**.** A microdata table Y^0 is said to be *k-anonymous* if each row Y^0_{i*} is identical to at least $k-1$ other rows. A frequency vector y^0 is called *k-anonymous* if for each entry y^0_l is either $y^0_l = 0$ or $y^0_l \geq k$.

Note that a microdata set is not k-anonymous if there is at least one data record, which does not fulfill the property described above. We call a data record, for which the k-anonymity property does not hold, a *confidentiality problem.* With respect to Example 1.1.1, for $k = 2$, the data records (female, Germany, inactive), (male, Germany, unemployed) and (male, Others, inactive) are confidentiality problems. Figure 1.2 shows an example of a 2-anonymous microdata set represented as a microdata table (1.2a) and a frequency vector (1.2b).

A very important tool for categorical data are *frequency tables.* A frequency table displays the absolute or relative frequencies of combinations of specific attributes and is of high interest especially for census data. In this thesis, we use frequency tables with absolute frequencies.

Definition 1.1.3 (Frequency table)**.** Let $\{j_1, \ldots, j_d\} \subseteq \mathcal{J}$ be a set of categorical attributes. A *frequency table* t, also called a *contingency table*, is a table consisting of $\prod_{h=1}^{d} |V(j_h)|$ table cells, i.e. the number of unique value combinations with respect to a subset of attributes. A table cell is defined by a combination of attribute values

Sex	*Citizenship*	*Working status*
female	Germany	employed
male	Germany	inactive
female	Germany	employed
male	Germany	inactive
male	Germany	inactive

(a) A 2-anonymous microdata table.

$$\left(\begin{array}{c}2\\0\\0\\0\\0\\0\\0\\3\\0\\0\\0\\0\end{array}\right)\begin{array}{l}\text{female,Germany,employed}\\\text{female,Germany,unemployed}\\\text{female,Germany,inactive}\\\text{female,others,employed}\\\text{female,others,unemployed}\\\text{female,others,inactive}\\\text{male,Germany,employed}\\\text{male,Germany,unemployed}\\\text{male,Germany,inactive}\\\text{male,others,employed}\\\text{male,others,unemployed}\\\text{male,others,inactive}\end{array}$$

(b) A 2-anonymous frequency vector.

Figure 1.2.: An illustrative 2-anonymous microdata set.

$(\bar{v}_{j_1}, \ldots, \bar{v}_{j_d}) \in V(j_1) \times \cdots \times V(j_d)$ and shows the frequency of its occurrence in the microdata set, which is as follows.

a) Given a microdata table Y^0, the entry for $(\bar{v}_{j_1}, \ldots, \bar{v}_{j_d})$ is given by

$$|\{i \in \mathcal{I} \colon Y^0_{ij_h} = \bar{v}_{j_h} \text{ for all } h = 1, \ldots, d\}|.$$

b) In the notation of a frequency vector y^0, the entry equals the sum over all entries y^0_l with $l = (v_1, \ldots, v_m)$, such that $v_{j_h} = \bar{v}_{j_h}$ for all $h = 1, \ldots, d$.

As described in the definition above, a frequency table is defined by a set of attributes. The number of attributes defines the dimension of the frequency table.

Definition 1.1.4 (Dimension of a frequency table)**.** Let $\{j_1, \ldots, j_d\} \subseteq \mathcal{J}$ be the set of attributes that describe a frequency table. Then, the frequency table is called *d-dimensional.*

For illustration purposes, the following example shows frequency tables corresponding to the microdata set from Example 1.1.1.

Example 1.1.5. The microdata set in Example 1.1.1 contains three attributes. Therefore, three 1-dimensional frequency tables, three 2-dimensional frequency tables, and one 3-dimensional frequency table exist. Figure 1.3 shows examples for frequency tables.

In general, frequency tables are only useful for categorical data. In contrast to nominal data, numerical attributes have many different values and, therefore, frequency tables are in general not useful unless the data is grouped into ranges of values, i.e. transformed into categorical data. Next, we describe properties of frequency tables. A frequency table is limited to a subset of attributes but includes all possible combinations within this subset. Therefore, each data row is counted in exactly one

Sex	Counts
female	3
male	2

(a) A 1-dimensional frequency table.

	Working status		
Sex	employed	unemployed	inactive
female	2	0	1
male	0	1	1

(b) A 2-dimensional frequency table.

Figure 1.3.: Exemplary frequency tables corresponding to Example 1.1.1.

table cell. Thus, the sum over all cell entries equals the total number of statistical objects n. Furthermore, an m-dimensional table includes *all* attributes and is therefore equivalent to the frequency vector y^0, which was defined previously. Frequency tables can also be defined to include marginal sums, which sum rows and columns, respectively. However, this is equivalent to a corresponding lower dimensional table. Therefore, we neglect marginal sums. For each subset of attributes, a corresponding frequency table can be created. Therefore, $\binom{m}{d}$ frequency tables of dimension d can be generated from a data set including m attributes. In total, $\sum_{d=1}^{m}\binom{m}{d} = 2^m - 1$ frequency tables correspond to a microdata set.

Furthermore, we are working with the dissimilarity between data records. For nominal data, there is no natural distance like the Euclidean distance for numerical data. Instead of the Euclidean distance, we use the *Hamming distance*, which counts inequalities of two data rows. Let

$$\delta(Y^0_{ij}, Y^0_{lj}) := \begin{cases} 1, & \text{if } Y^0_{ij} \neq Y^0_{lj}, \\ 0, & \text{if } Y^0_{ij} = Y^0_{lj}. \end{cases} \tag{1.1.1}$$

Then, the Hamming-distance is defined as follows.

Definition 1.1.6 (Hamming-distance)**.** The *Hamming distance* between two vectors Y^0_{i*} and Y^0_{l*} is defined by

$$w_{il} := \sum_{j\in\mathcal{J}} \delta(Y^0_{ij}, Y^0_{lj}).$$

We will work with *clustering*

1.2. Literature Review

1.2.1. Different Anonymization Approaches

The definition of anonymity is a central aspect when discussing anonymization approaches. In Section 1.1, we gave a definition of the common concept of k-anonymity, which was introduced by Sweeney (2002). This approach ensures that there is no unique or very rare data record in the published microdata set. A microdata set is

k-anonymous if each data record is identical to at least $k-1$ other records. In recent years, another approach to anonymity has also gained in interest, which is called *differential privacy*. The basic idea of differential privacy is that the conclusions, which can be drawn from the analysis of a data set, should be independent of whether the data of an individual are included in the data set or are not included. To this end, noise is added to the data so that a response when an individual record is included in the data set is almost as likely as the same response when the individual record is not included. The first main contributions to this field come from Dinur and Nissim (2003), Dwork and Nissim (2004) and Dwork et al. (2006). Dwork (2008) describe the development of differential privacy from semantic security in cryptosystems (Goldwasser and Micali, 1984) to the usage in statistical disclosure control. However, Clifton and Tassa (2013) point out that differential privacy cannot replace k-anonymity and vice versa. They argue that k-anonymity is used to publish anonymized data regardless of the type of analysis and queries, which are performed on the data by any users. In contrast to that, differential privacy is originally devised for *queries*, which have to be known in advance, and is therefore classically embedded in the field of data *mining* instead of data *publishing*. Later, differential privacy was extended to the field of data publishing, as summarized by Zhu et al. (2017). However, Zigomitros et al. (2020) expose that there are still some challenges with respect to the efficiency of using differential privacy as privacy-preserving method to publish microdata sets. In this thesis, we focus on k-anonymity because it is frequently used, e.g. in the German census in 2011, and allows the publication of microdata sets for general purposes with reasonable benefit.

Next, we provide an overview of different approaches to anonymize data as described by Hundepool et al. (2012), Willenborg and De Waal (2012) and Domingo-Ferrer et al. (2016). Mainly, anonymization approaches can be divided into *perturbative* and *non-perturbative* methods and into methods on the level of *frequency tables* and on the level of *microdata*.

As discussed by di Vimercati et al. (2011), in the past, data were mostly published in aggregated form, i.e. as macrodata. For categorical data, frequency tables are a typical aggregated form. Later, however, anonymization on the level of microdata became research focus, since it allows for more flexibility. Users can perform any analysis on published microdata and are not restricted to the aggregated information, which is provided by published macrodata. In this thesis, we focus on the anonymization on microdata level.

Perturbative methods modify the original microdata set to ensure anonymity. Instead of the original microdata set the perturbed data are released. Typically, these methods aim to ensure that statistics computed on the modified data are close to the statistics computed on the original data. Contrarily, non-perturbative methods do not alter data and, thus, whenever a value is published, it is consistent with the truth. Here, anonymity is ensured by suppressing some of the information or by reducing the detail of information.

For k-anonymity, the non-perturbative methods suppression and generalization were first used as illustrated by Samarati and Sweeney (1998). Suppression methods hide certain values. Intuitively, the fewer values are suppressed, the less information is lost. Note that it is not sufficient to suppress only sensitive values to prevent recalculation of the original values from the published entries. To find an optimal suppression on frequency tables, several strategies exist, e.g. by Kelly et al. (1992), Cox (1995), and Fischetti and Salazar (2001). In contrast to suppression methods, which lead to a total information loss for certain values, generalization methods reduce the detail of the published information. To this end, in generalization methods, values are replaced by truthful but more general values. Note that generalizing (or recoding) to the most general value is equivalent to suppression. The less generalized and the more specific the published values, the less information is lost. There are several works on global recoding, e.g. by Bayardo and Agrawal (2005) and Fung et al. (2005), and on local recoding, e.g. by Sweeney (2002), LeFevre et al. (2005) and Xu et al. (2006). Moreover, suppression and generalization can also be combined to reduce the information loss as proposed by Samarati and Sweeney (1998) and Hurkens and Tiourine (1998).

Machanavajjhala et al. (2007) point out privacy issues, which can arise from homogeneity attacks (attribute disclosure) and background knowledge attacks, when non-perturbative methods like generalization and suppression are used to obtain k-anonymous data. In the first scenario, problems arise when all entities in the same group are originally identical and, thus, even without re-identification of a person, the true value can be revealed if a person can be clearly assigned to a group. Background knowledge might be exploited to exclude some values and thus infer the true value. To overcome these issues, Machanavajjhala et al. (2007) extend the concept of k-anonymity to l-diversity, which requires at least l distinct "well-represented" sensitive values in each equivalence class, and Li et al. (2007) propose t-closeness as a further extension, which requires the distance between the distribution of sensitive values in each equivalence class and in the whole data set to remain below threshold t. However, these problems can also be circumvented by using perturbative methods. When perturbative methods are used, a user does not know which of the values are truthful and therefore cannot draw any conclusions with certainty from k-anonymous data. Therefore, in this work we focus on a perturbative method to obtain k-anonymity.

Another aspect for anonymization methods is the kind of data. Some methods, which are suitable for numerical data, cannot be applied to categorical data and vice versa. Domingo-Ferrer (2008) provides an overview over anonymization methods for numerical and for categorical variables. The most common perturbative methods for numerical variables are noise addition (Brand, 2002), microaggregation (Domingo-Ferrer and Torra, 2005), rank swapping (Dalenius and Reiss, 1982), rounding (Cox & Ernst, 1982), and resampling (Domingo-Ferrer and Mateo-Sanz, 1999 and Domingo-Ferrer, 2008). For categorical variables, the most common perturbative methods are rank swapping (Moore, 1996), PRAM (Post Randomization Method; Kooiman, 1997), MASSC (Micro Agglomeration, Substitution, Subsampling and Calibration; Singh et al., 2004), and microaggregation (Torra, 2004). In this thesis, we focus on a microaggregation procedure.

1.2.2. Clustering and Microaggregation

Microaggregation is a popular perturbative method to obtain k-anonymous microdata. The main idea is to partition the microdata set into clusters of size greater than or equal to k. Then, in each cluster, the individual data records are substituted by the same values. By this substitution, all records in the cluster coincide with each other and, therefore, k-anonymity holds. In this section, we first describe clustering problems, which are related to the partitioning problem since they also aim to partition the microdata set into clusters of similar data rows. Clustering problems were first introduced for numerical data and only later, extensions on categorical data were studied. Then, we highlight the differences between clustering and microaggregation. Similar to clustering, microaggregation procedures were developed for numerical data first and then extended to categorical data.

It is common to divide the microaggregation process into two steps, the partitioning and the aggregation step. In the partitioning step, the aim is to assign the data vectors, such that the clusters are homogeneous, i.e. contain similar data vectors. The partitioning step is reminiscent of clustering problems such as the well-known c-means clustering problem[1] for numerical data. Both tasks consist of dividing a data set into groups of similar elements. While the microaggregation procedure requires clusters with a minimum size k, the cluster sizes do not matter in classical clustering problems. Instead, the number of clusters is required to be c. In this section, we recall the c-means clustering problem, show the differences between clustering and microaggregation problems and discuss the problems, which arise when extending approaches for numerical data to categorical data.

The c-means clustering problem deals with numerical data and goes back to MacQueen et al. (1967). The goal is to partition a given data set $\mathcal{I}$ into c clusters of similar data rows and to find cluster representatives for each cluster. The objective is to minimize the sum of squared errors between the data vectors and the cluster centroid of the cluster to which they are assigned. It can be formulated by using binary variables a_{is} to indicate whether a data vector Y^0_{i*} is assigned to cluster s, and variables $\mu_s \in \mathbb{R}^m$ for the cluster representatives. The problem can be formulated by

$$\begin{aligned}
&\min_{a,\mu} \sum_{s=1}^{c} \sum_{i \in \mathcal{I}} a_{is} \cdot \|Y^0_{i*} - \mu_s\|^2 && && (1.2.1)\\
&\text{s.t.} \ \sum_{s=1}^{c} a_{is} = 1 && \forall i \in \mathcal{I}, && (1.2.2)\\
&\qquad a_{is} \in \{0,1\} && \forall i \in \mathcal{I}, s = 1, \dots, c, && (1.2.3)\\
&\qquad \mu_s \in \mathbb{R}^m && \forall s = 1, \dots, c && (1.2.4)
\end{aligned}$$

[1] Usually, this problem is called the k-means clustering problem. Here, it is renamed to avoid confusion with k-anonymity.

for given data vectors $Y^0_{i*} \in \mathbb{R}^m$. The objective function in (1.2.1) takes the distance between data vector Y^0_{i*} and the representative of cluster s into account, if Y^0_{i*} is assigned to cluster s. Constraints (1.2.2) ensure, that each data vector is assigned to exactly one cluster.

The most popular approach is the algorithm by Lloyd (1982). The basic idea is to alternately choose cluster representatives and assign the data rows to the best matching representatives. As an initialization, c vectors $\mu_1, \ldots, \mu_c$ are chosen randomly as cluster representatives. Afterwards, each data vector is assigned to the cluster with the closest cluster representative. So, a data row Y^0_{i*} is assigned to cluster $\tilde{s} := \arg\min_{s=1,\ldots,c} \|Y^0_{i*} - \mu_s\|^2$. After the assignment, each cluster representative μ_s is updated to be the mean of the assigned data vectors. This procedure of alternately assigning data vectors to clusters and updating cluster representatives is repeated until there are no more changes. Since the method works with the Euclidean distance and the mean of data vectors, it is only defined for numerical data, where these are properly defined.

Huang (1998) proposes an adaption of Lloyd's algorithm for *categorical* data, the so-called c-modes algorithm[2]. Since neither the Euclidean distance nor the mean are defined for categorical data, Huang (1998) proposes the following adaptions for categorical data. The objective function is adapted to replace the Euclidean distance with the Hamming-distance, which counts the number of mismatches between two data vectors and the mean is replaced with the *plurality mode*, which is the vector of most frequent values. Then, the proposed method follows the same procedure as the algorithm by Lloyd (1982). Note that the resulting cluster representative is not necessarily contained in the original microdata set. This is also the case for numerical data, where the mean of vectors does not necessarily occur as a data vector in the microdata set. Moreover, the plurality mode is ambiguous and San et al. (2004) propose a c-representative algorithm to overcome the non-uniqueness of the mode as cluster representative. They propose to choose data vectors with corresponding frequencies in the cluster as representatives. The works by Chen and Wang (2013) and Nguyen et al. (2019) go in the same direction and they propose to include probability distributions in the cluster representations.

Clustering problems aim to analyze the data and underlying structures. These problems differ from the task of microaggregation procedures, which is to preserve privacy by replacing individual data records with common representative values. To this end, in the partitioning step of a microaggregation procedure, the size of the clusters is relevant in contrast to the number of clusters. Each non-empty cluster must contain at least k elements to obtain k-anonymous microdata via microaggregation. Based on the formulation (1.2.1)–(1.2.4) of the clustering problem, the microaggregation problem can be formulated as follows. Since the problem enforces at least k elements in each cluster, the maximum number of non-empty clusters is $c := \lfloor \frac{|\mathcal{I}|}{k} \rfloor$, which is known in advance. Let $d(\cdot, \cdot)$ denote a distance function, which is defined according to the

[2] Usually, this problem is called k-modes problem.

given data. Then, the partitioning problem in the microaggregation procedure is given by

$$
\begin{aligned}
\min_{a,\mu} \; & \sum_{s=1}^{c} \sum_{i \in \mathcal{I}} a_{is} \cdot d(Y^0_{i*}, \mu_s) && && (1.2.5)\\
\text{s.t.} \; & \sum_{s=1}^{c} a_{is} = 1 && \forall i \in \mathcal{I}, && (1.2.6)\\
& \sum_{l \in \mathcal{I} \setminus \{i\}} a_{ls} \geq (k-1) \cdot a_{is} && \forall i \in \mathcal{I}, && (1.2.7)\\
& a_{is} \in \{0,1\} && \forall i \in \mathcal{I}, s = 1, \dots, c, && (1.2.8)\\
& \mu_s \in \mathbb{R}^m && \forall s = 1, \dots, c. && (1.2.9)
\end{aligned}
$$

The objective function in (1.2.5) is analogous to the objective function in (1.2.1) and takes into account the distance between a data record and the cluster representative of the cluster, to which the record is assigned. Similar to the clustering problem, Constraints (1.2.6) ensure that each data vector is assigned to exactly one cluster. The lower bound on the cluster sizes, which is crucial for a microaggregation procedure, is ensured by Constraints (1.2.7). Note that in a feasible solution, empty clusters are allowed but each non-empty cluster is enforced to contain at least k elements. Constraints (1.2.8) enforce the indicator variables a_{is} to be binary. Constraints (1.2.9) are substituted by $\mu_s \in \mathbb{Z}^m$ for categorical data.

The most-common heuristic approach for microaggregation on numerical data is the MDAV algorithm, which was proposed by Domingo-Ferrer and Mateo-Sanz (2002). The method is designed for fixed-size microaggregation, i.e. the resulting clusters have a fixed size of k (possibly except for one cluster). The outline of the algorithm is as follows. At first, all data records are marked as unassigned. The centroid $\hat{Y^0}$ over all unassigned data vectors is computed. Vector Y^0_{i*} the most distant to $\hat{Y^0}$ is computed. Then, vector Y^0_{l*} the most distant to Y^0_{i*} is computed. The $k-1$ closest data vectors to Y^0_{i*} and Y^0_{l*} are clustered with Y^0_{i*} and Y^0_{l*}, respectively, and marked as assigned. This procedure is repeated with all unassigned data vectors until there remain less than $2k$ unassigned vectors or all data vectors are assigned to a cluster. If there remain at least k data vectors, they build a new cluster. Otherwise, the remaining data vectors are assigned to their closest, already existing cluster. However, specifying fixed cluster sizes is a restriction, which Solanas et al. (2006) aim to overcome by their proposed V-MDAV algorithm. The V-MDAV algorithm is based on the MDAV algorithm but it allows for variable cluster sizes. A parameter is introduced in V-MDAV, which decides about adding another element to a group, which already contains at least k elements. Following the same goal, Soria-Comas et al. (2019) also propose a method, which allows variable cluster sizes. However, their method is more closely aligned with Lloyd's algorithm and does not need a parameter for the cluster sizes. We depict the idea of the algorithm. First, all elements are randomly assigned into clusters of at least k elements. Then, equivalently to Lloyd's algorithm, the centroids of the current clusters are computed and all elements are (re-)assigned to the cluster with the nearest centroid, iteratively. To ensure k-anonymity, elements are considered for reassignment only if the current cluster has more than k elements. In this way, the minimum cluster size of k is respected in each iteration. A cluster of k elements can also be dissolved and divided among the other clusters if this yields a lower group heterogeneity overall. The mentioned approaches were all introduced for numerical data. Later, microaggregation methods were extended for categorical data.

The first work on microaggregation for categorical data was done by Torra (2004). The basic idea is to replace the Euclidean distance with the Hamming-distance for nominal data and the number of categories separating two values for ordinal data. For the computation of the centroid, he proposes to use the plurality mode for nominal data and the (convex) median for ordinal data. Then, an algorithm for the c-modes problem is performed. Afterwards, some elements are relocated such that the clusters fulfill the minimum-size constraint, which is necessary for k-anonymity. In his article, Torra (2004) does not specify the relocation of data records further. Also, Marés and Torra (2012) describe a microaggregation procedure for categorical data. They also first find a partitioning, which fits the c-modes problem. Then, the smallest clusters are merged iteratively until all clusters have at least k elements. In both methods, the data records in a cluster are replaced by the values of the cluster centroid. However, there are multiple possibilities for the aggregation step, in which the microdata set is changed based on the partitioning. For example, in the context of local recoding each individual value is replaced with an interval for numerical data or a set of distinct values for categorical data, which contains all values in the respective cluster. Aghdam and Sonehara (2016) contribute to this field and come up with a greedy bottom-up algorithm, which can be performed on mixed data, i.e. including numerical and categorical variables.

The reported approaches all have in common that they focus on the distances between data rows and (potential) cluster representatives. For numerical data, this is reasonable since the cluster representatives are classically the mean values and, hence, the distance between a data row and a cluster representative also allows to draw conclusions about the distances between this data row and the other data records, which are assigned to the same cluster. However, for categorical data, this approach is not sufficient. In general, data records, which are close to a representative, do not necessarily have to be close to each other. We address this problem in Chapter 4.

However, there are also alternative approaches considering the overall dissimilarity in each cluster instead. For example, Byun et al. (2007) define the so-called k-member clustering problem as follows. The task is to find a set of clusters $\mathcal{C}^* = \{C_1, \ldots, C_c\}$ such that

1. $C_s \cap C_t = \emptyset \quad \forall s, t \in \{1, \ldots, c\}, s \neq t$ (disjoint clusters)

2. $\bigcup_{s=1}^{c} C_s = \mathcal{I}$ (partitioning)

3. $|C_s| \geq k \quad \forall s = 1, \dots, c$ (k-anonymity)

4. $\sum_{s=1}^{c} |C_s| \cdot \max_{Y_{i*}^0, Y_{l*}^0 \in C_s} d(Y_{i*}^0, Y_{l*}^0)$ is minimal (minimum information loss),

where d is a distance function. By using the maximum function, they determine the quality of a cluster by its diameter. Their specific calculation of the information loss (4) matches the generalization method, which they use for anonymization. In contrast to the approaches mentioned earlier, their method does not only take the distances between data records and a centroid into account but considers the overall dissimilarity by using the cluster diameter. We contribute with this thesis in a similar direction and propose a method, which takes an overall dissimilarity for each cluster into account. However, our approach with overall dissimilarity differs from the reported approach by Byun et al. (2007) since we propose a data-perturbative approach. Along the same lines, Castro et al. (2022) interpret the partitioning problem in a microaggregation procedure as a clustering problem with restriction on the cluster sizes. Their proposed method aims to minimize cluster weights, which are defined as the sum of distances between *each* pair of data records. By taking the cluster weights based on distances between each two data records into account, they include an overall dissimilarity instead of only taking the distances to a cluster centroid into account. They focus on numerical data and present a heuristic based on microaggregation approaches for numerical data and only mention that an extension to categorical data is possible. We build on their idea and extend the work to nominal data. The main aspect in their work is a formulation of the partitioning problem in a column-oriented way, which can be used in a column generation procedure. Since column generation is crucial for their work and our approach, we outline the column generation scheme in the next section.

1.2.3. Column Generation

In this section, we outline the so-called *column generation* procedure, which is part of the proposed method. A column generation scheme is well suited for linear programs with a very high number of variables and a comparatively small number of constraints. The initial idea can be traced back to Ford Jr and Fulkerson (1958). We refer to Lübbecke (2010) and Desrosiers and Lübbecke (2005) in this section and give a brief introduction to the method.

Let a general linear program be given by

$$\min_x \sum_{C \in \mathcal{C}} w_C x_C \tag{1.2.10}$$

$$\text{s.t.} \sum_{C \in \mathcal{C}} a_{iC} x_C = 1 \quad \forall i \in \mathcal{I}, \tag{1.2.11}$$

$$x_C \geq 0 \qquad \forall C \in \mathcal{C} \tag{1.2.12}$$

with given weights w_C, parameters a_{iC} and an index set $\mathcal{C}$. In the context of column generation, problem (1.2.10)–(1.2.12) is called the *master problem*. Column generation is usually used in cases, in which the number of variables $|\mathcal{C}|$ is exponential in the number of constraints $|\mathcal{I}|$. Due to the large number of variables, it is not practical to consider all variables explicitly. However, in usual applications, in an optimal solution only about $|\mathcal{I}|$ variables are non-zero. Column generation exploits the fact that the linear program is easy to solve for a small variable set and adds variables iteratively. The procedure starts with replacing the index set $\mathcal{C}$ with a smaller subset $\mathcal{C}' \subset \mathcal{C}$. The program obtained by this replacement is called the *restricted master problem* (RMP). Iteratively, variables are added to the restricted master problem, which might be included in an optimal solution. Whether a variable is added to the RMP is decided based on the reduced costs. Let u_i be dual variables corresponding to Constraints (1.2.11) for all $i \in \mathcal{I}$. The reduced costs for x_C are defined as

$$w_C - \sum_{i \in \mathcal{I}} u_i a_{iC}.$$

They state the amount by which the objective function coefficient according to x_C would have to decrease before it would be possible to assume a non-zero value for x_C in an optimal solution. Therefore, when the reduced costs are negative, the respective variable is potentially included in an optimal solution. It is common to search for variables with negative reduced costs implicitly by solving the *pricing problem*, which is given by

$$\min_{C \in \mathcal{C}} w_C - \sum_{i \in \mathcal{I}} u_i a_{iC}.$$

A column generation scheme first solves the restricted master problem and derives dual variables. With the current dual variables, variables with negative reduced costs are computed. This procedure is repeated until the optimal objective value of the pricing problem is non-negative, i.e. there are no more variables with negative reduced costs. An optimal solution to the restricted master problem at this point also solves the master problem optimally.

Usually, the pricing problem is rewritten to an implicit form, which typically is a well-structured optimization problem. However, in general the pricing problem again is not easy to solve optimally. It is therefore common to first apply heuristics and only solve the pricing problem to optimality if no heuristic provides variables with negative reduced costs. In the best case, the pricing problem only needs to be solved to optimality once to prove that an optimal solution to the master problem is found.

1.3. Scope of This Thesis

This thesis contributes to the subject of microaggregation for nominal data. First, we study the anonymization approach, which was used in the last German census in 2011. To this end, we analyze the optimization problem, which is described by Höhne (2015) and prove $\mathcal{NP}$-hardness for a slightly modified version. This result justifies the search for heuristic approaches. Moreover, we discuss the underlying optimization problem and find that the applied heuristic SAFE covers aspects, which are not included in the optimization problem. Therefore, we propose a modification of the optimization problem to also include the important aspect of remaining close to the original frequency vector after anonymization.

Typically, a microaggregation procedure is divided into two steps, the partitioning and the aggregation step. We follow this idea and propose methods for both, the partitioning and the aggregation step, which respect the specific properties of nominal data. Based on works by Castro et al. (2022) and Zhao et al. (2018), we propose a column generation-based heuristic for the partitioning problem for nominal data. We interpret the partitioning problem as a problem of finding weight-minimum cliques in a complete graph with a lower bound on the clique size. Each data record corresponds to a vertex and the Hamming-distance between two vertices is used as edge weights. We use column generation to obtain a fractional solution, which provides dual information. Building on the idea by Zhao et al. (2018), we use this dual information and an auxiliary problem to construct an integer solution and present the formulation of the auxiliary problem for the partitioning problem. A main advantage of this way of data partitioning is that in each cluster not only the distance to potential representatives is taken into account but also the dissimilarity between *each* data pair in the same clique. This is beneficial because in contrast to numerical data, for categorical data it is not clear if two data records, which are similar to the same representative are also similar to each other. Next, we analyze the aggregation phase, in which representative values are selected. Especially for nominal data, frequency tables are of high interest in practice. Similar to the SAFE heuristic used in the German census 2011, we therefore take resulting errors in a pre-selected set of frequency tables, the so-called control tables, into account. Our approach differs from the idea of the SAFE heuristic because the representatives are selected for previously constructed cliques. Moreover, we discuss to include the χ^2-error in table cells as an objective and we present a mathematical formulation of the problem. Furthermore, we show that this optimization problem can also be reformulated to a clique problem in a multipartite graph. This representation can be useful to develop further heuristics and we present one possible heuristic based on this graph problem.

Based on the optimization problem formulated by Höhne (2015) and our proposed modification, we present a mathematical optimization problem, which combines the two phases of a microaggregation procedure and aims to minimize the sum of χ^2-errors in the control tables. We use the exact solving of this optimization problem for comparison on small instances in our computational experiments.

Finally, we study the performance of the individual steps of the proposed microaggregation procedure and compare the proposed methods with the SAFE heuristic and optimal solutions to the presented optimization problems by computational experiments. We analyze several aspects, including the parameter settings, different analysis criteria for the comparison of different methods, the effects on a logistic regression based on anonymized data, and runtime performances for increased instance sizes.

An overview of the method, which we present in this thesis, is outlined in Figure 1.4.

1. Partitioning Step
   - Find a fractional clustering of the original microdata set with clusters of size between k and $2k-1$.
   - Use dual information provided by the fractional solution to find an integer clustering.

2. Aggregation Step
   - Find cluster representatives taking χ^2-errors in resulting frequency tables into account.
   - In each cluster, replace individual data records by representative values to obtain k-anonymity.

Figure 1.4.: Scheme of our proposed method.

Chapter 2

SAFE Method

The SAFE method was used to anonymize the microdata from the German census in 2011. We refer to the work by Höhne (2015), Höhne (2010) and Höhne (2003) and outline the algorithm in this chapter. The SAFE method is a data-perturbative method to generate a k-anonymous microdata set. To this end, a predefined set of frequency tables is taken into account such that the maximum absolute error between cell values before and after anonymization is minimized. We call the tables in the predefined set *control tables* and denote the set of control table cells by $\mathcal{T}$.

In Section 2.1, we describe the underlying optimization problem before outlining the SAFE method in Section 2.2.

2.1. Problem Description and Formulation

The SAFE method is based on an optimization problem, which aims to minimize the maximum absolute error in the control tables. To formulate this optimization problem, the representation of microdata by a frequency vector is used, which we denote by y^0.

We take a closer look at the calculation of frequency tables. As described in Section 1.1, a frequency table cell equals the sum of the corresponding entries in a frequency vector y^0. Therefore, all table cells can be obtained by multiplying y^0 with a binary matrix A, which is a block matrix. Each block of matrix A corresponds to one table and each row corresponds to a cell in the table. We therefore can write

$$A = \left(\begin{array}{c} A_1 \\ \hline \vdots \\ \hline A_T \end{array} \right),$$

where T is the number of frequency tables. We examine the properties of each matrix A_s corresponding to a d-dimensional table s. Let $\{j_1, \ldots, j_d\} \subset \mathcal{J}$ be the attributes defining table s. Assuming that the frequency vector y^0 represents all possible unique combinations of attribute values, the matrix A_s fulfills the following properties, such that $A_s y^0$ computes the table entries:

1. In each column there exists exactly one 1 as each element of the frequency vector is counted in exactly one table cell for each table.

2. The rows of A_s must sum to the n-dimensional vector of ones. This is due to the fact that each entry of the frequency vector y^0 is counted in one of the cells of table s.

3. In each row of the matrix the same number of ones must occur. Let $(v_{j_1}, \dots, v_{j_d})$ be the combination of attribute values that defines a cell in table s. This combination occurs in $\prod_{j \in \mathcal{J} \setminus \{j_1,\dots,j_d\}} |V(j)|$ flavors. As this number of combinations depends only on the attributes and not on the specific values of the cell, it is the same for all table cells.

The SAFE method works with an abbreviated frequency vector, which only contains all unique combinations that actually occur in the microdata set. We denote the number of distinct occurring combinations, i.e. the length of the abbreviated frequency vector, by n_a. In this case, property 1 and 2 remain the same. The last property does not hold anymore because not all possible combinations occur in the frequency vector. Thus, the number of combinations, which are counted in a table cell, can vary.

The following optimization problem for finding 3-anonymous microdata was the basis for the SAFE-method.

$$\min_{y,b} \max_{t \in \mathcal{T}} |b_t| - g_t \tag{2.1.1}$$

$$\text{s.t.}\quad Ay + b = Ay^0, \tag{2.1.2}$$

$$\sum_{i=1}^{n_a} y_i = \sum_{i=1}^{n_a} y_i^0, \tag{2.1.3}$$

$$y_i \in \mathbb{N}_0 \setminus \{1,2\} \qquad \forall i \in \mathcal{I}, \tag{2.1.4}$$

$$b_t \in \mathbb{R} \qquad \forall t \in \mathcal{T}, \tag{2.1.5}$$

where $g_t \geq 0$ is fixed. The purpose of value g_t is to allow for some tolerance to errors within the cell's data. We call this problem the *SAFE-basic problem.* The objective function in (2.1.1) is the maximum absolute error over all control table cells subtracted by parameter g_t, which is predefined for each cell $t \in \mathcal{T}$. As the impact of a given absolute error depends on the value of the affected table cells, Höhne (2015) proposes to adjust the parameters g_t on the corresponding value of table cell t with larger g_t for a larger value in t. The vector b is enforced to be the vector of all errors in the control table cells by Constraint (2.1.2). As discussed above, the term Ay^0 equals a vector of all original cell values of the control tables. Analogously, the vector Ay contains the cell values resulting from frequency vector y, which is a variable of the optimization problem. Therefore, by Constraint (2.1.2), an entry b_t is defined as the error between the cell values for t before and after anonymization. Furthermore, Constraint (2.1.3) ensures that the number of data rows in the original microdata set equals the number of data rows in the microdata set after anonymization. Moreover, Constraints (2.1.4) ensure that y is a 3-anonymous frequency vector by excluding counts of 1 and 2. Constraints (2.1.2) together with (2.1.4) enforce all entries of b to be integer, therefore Constraints (2.1.5) suffice. Note that the absolute value in the objective function can be linearized by introducing auxiliary variables.

2.2. Outline of the Algorithm

In this section, we report the SAFE method presented by Höhne (2015), which is designed for 3-anonymity.

The SAFE method can be assigned to the microaggregation methods. It does not divide the procedure into partitioning and aggregation but combines both steps in one. The basic idea is to look at three entries of the frequency vector at a time and group them, such that the maximum error in the control tables is minimized and the constructed frequency vector is k-anonymous. The groups are found by testing possible changes in the frequency vector for each three entries and selecting the best one. First, the method tries to group only confidentiality problems together. If this does not produce an acceptable solution with respect to the number of eliminated confidentiality problems, also non-confidentiality problems are taken into account.

The algorithm SAFE runs through the original frequency vector y^0 piecewise and adopts its entries iteratively such that finally it returns a k-anonymous frequency vector. We depict the single steps of the algorithm in the following. Before the frequency vector y^0 is given to the algorithm, the entries are sorted, such that the represented attributes are sorted in ascending order based on the number of possible values. For instance, the exemplary frequency vector illustrated in Figure 1.1b is sorted in this way.

Description of Algorithm 1 The procedure of the SAFE method is depicted in Algorithm 1. The SAFE method runs through the given frequency vector y^0 step by step. It divides the vector into smaller parts and runs through the parts iteratively. A size of the parts is predefined and the next entries of the sorted frequency vector corresponding to this number are examined together. In each part, the method identifies the confidentiality problems in the frequency vector. Sequentially, the algorithm runs through each three indices of confidentiality problems and goes through Algorithm 2. This algorithm returns a frequency vector, which is modified in the given indices by testing possible changes, such that the number of confidentiality problems is decreased and a predefined bound $b+g$ on the absolute error between original and new cell entries is respected. When the algorithm has iterated over all confidentiality problems in the part, the proportion of eliminated confidentiality problems is computed. If this proportion is accepted, the next part is added and again, all confidentiality problems are considered, including the ones remaining from the previous part. In each iteration, the proportion of eliminated confidentiality problems is computed. If an adequate number of confidentiality problems are eliminated with respect to the size of the analyzed part, the procedure is repeated until all confidentiality problems are eliminated. If too few confidentiality problems are eliminated in an iteration, the bounds b are increased and the procedure is repeated. Higher bounds b allow more variability in Algorithm 2. If the algorithm has iterated over all parts and the number of eliminated confidentiality problems is accepted, there can still be confidentiality problems. If the proportion of eliminated confidentiality problems in total is too low, then the procedure is repeated by including the two nearest non-zero values next to the confidentiality problems. If the proportion remains too low, then the confidentiality problems and their two nearest values are taken into account. Finally, if confidentiality problems still occur and the number of eliminated ones is too low, the procedure is repeated with all values in the frequency vector - independent from their classification as confidentiality problems or zero. In each variant, the algorithm stops, if in the whole frequency vector all confidentiality problems are eliminated as the incumbent frequency vector then fulfills 3-anonymity. Next, the resulting frequency vector y^*, which fulfills the property of k-anonymity, is improved by applying Algorithm 3.

Description of Algorithm 2 Given the original and a current working frequency vector as well as three indices to consider, the selection rule described in Algorithm 2 decides how to change the current working data in these entries such that the *best* modified working data is returned. The set AC ("all candidates") is initialized with an empty set. Iteratively, all possible changes between 0 and ±3 are tried out. A frequency vector c is added to the set AC if it contains only non-negative entries, does not differ too much from the original frequency vector y^0, and the absolute error between original and resulting control cell values lies in the given bound $b_t + g_t$. The allowed absolute error between original and modified frequency vector is determined by parameter *"allowed_deviation"*. The absolute error between changed and original cell entries is computed by $|(Ay^0)_t - (Ac)_t|$ for candidate c and table cell t. If any absolute error exceeds the given bound $b_t + g_t$, the candidate c is not included in the set AC.

The set AC contains modified frequency vectors, which are potentially returned by Algorithm 2. From this set, the *best* candidate is returned in regard of several criteria. These criteria are reviewed by generating the following subsets. The set C_min_problems contains all candidates in AC, which have the minimum number of confidentiality problems. Next, the elements in C_min_problems minimizing a penalty function s are included in the more restrictive subset C_min_penalty. Here, the penalty function s penalizes frequency vectors, which generate tables cells with absolute errors near to the bound $b_t + g_t$. These candidates allow less changes in further iterations in the SAFE method. Let $f_t := |(Ay^0)_t - (Ac)_t|$ denote the absolute error in table cell t corresponding to candidate c. The penalty function s used in SAFE is defined by

```
   Input: sorted original frequency vector $y^0$, $k$ for $k$-anonymity, initial bound $b$.
   Output: $k$-anonymous frequency vector $y^*$.
 1 Initialization: $y^* \leftarrow y^0$;
 2 while there are confidentiality problems in $y^*$ do
 3     Consider single parts of $y^*$;
 4     for each part do
 5         Find all indices of confidentiality problems $[i_1, \dots, i_{\tilde{n}}]$ in the current
             and previously analyzed parts of $y^*$;
 6         for each three consecutive indices $[i_l, i_{l+1}, i_{l+2}]$ do
 7             Run Algorithm 2 with the input $y^0$, $y^*$, $b$ and $[i_l, i_{l+1}, i_{l+2}]$ and get
                 the possibly changed frequency vector $y^*$;
 8         end
 9         if there are no more confidentiality problems in $y^*$ then
10             break;
11         end
12         if #eliminated confidentiality problems in this part is proportionally
             too low then
13             Increase bound $b$.
14             Go back to line 5;
15         end
16     end
17     if #eliminated confidentiality problems in $y^*$ is proportionally too low
         then
18         Redo lines 3–16 replacing $[i_l, i_{l+1}, i_{l+2}]$ with $[i_l - s, i_l, i_l + \tilde{s}]$, for the
             smallest $s$ and $\tilde{s}$, such that the elements are non-zero (and not
             necessarily confidentiality problems);
19     end
20     if #eliminated confidentiality problems in $y^*$ is still proportionally too low
         then
21         Redo lines 3–16 with $[i_l - 1, i_l, i_l + 1]$ instead of $[i_l, i_{l+1}, i_{l+2}]$;
22     end
23     if #eliminated confidentiality problems in $y^*$ is still proportionally too low
         then
24         Redo lines 3–16 with $[i, i+1, i+2]$ for all $i = 1, \dots, n-2$ instead of
             $[i_l, i_{l+1}, i_{l+2}]$;
25     end
26 end
27 Run algorithm 3 to improve the current solution and get an improved
     $k$-anonymous frequency vector $y^*$;
28 return $y^*$
```

Algorithm 1: SAFE.

$s := \sum_{t \in \mathcal{T}} s_t$, where

$$s_t := \begin{cases} 9, & \text{if } f_t = b_t + g_t, \\ 4, & \text{if } f_t = b_t + g_t - 1, \\ 1, & \text{if } f_t = b_t + g_t - 2, \\ 0, & \text{else.} \end{cases}$$

In the next step, the candidates in C_min_penalty are chosen, that minimize the sum of absolute errors in the table cells. The sum of absolute errors is computed by $\sum_{t \in \mathcal{T}} |(Ay^0)_t - (Ac)_t|$. All candidates in C_min_penalty with minimum sum of absolute errors are included in the set C_min_sum_error. If this set is non-empty, the algorithm returns one of its elements. Otherwise, it returns the frequency vector that was given as input without any change.

```
Input: original frequency vector $y^0$, working frequency vector $y$, bound $b$,
       extra error $g$, indices $[i_1, i_2, i_3]$, parameter allowed_deviation, penalty
       function $s$.
Output: new working frequency vector $y^z$.
Initialization: AC ← $\emptyset$, c ← $y$, $y^z$ ← $y$;
Consider $y_{i_1}, y_{i_2}$ and $y_{i_3}$;
for $(h_1, h_2, h_3) \in \{-3, -2, \dots, 2, 3\}^3$ do
    $c_{i_j} \leftarrow y_{i_j} + h_j$ for $j = 1, 2, 3$;
    if $c_{i_j} \geq 0$ and $|c_{i_j} - y^0_{i_j}| \leq$ allowed_deviation for $j = 1, 2, 3$ then
        if $|(Ay^0)_t - (Ac)_t| \leq b_t + g_t$ for all $t \in \mathcal{T}$ then
            AC ← AC ∪ {c};
        end
    end
end
C_min_problems ← {c in AC: #confidentiality problems in c is minimal};
C_min_penalty ← {c in C_min_problems: penalty function $s(c)$ is minimal};
C_min_sum_error ← {c in C_min_penalty: sum of absolute errors in control
  tables is minimal};
if C_min_sum_error ≠ $\emptyset$ then
    $y^z$ ← first(C_min_sum_error);
return $y^z$
```

Algorithm 2: Selection rule.

Description of Algorithm 3 When the SAFE method has found a 3-anonymous frequency vector y, it calls Algorithm 3 to improve the solution, such that the maximum error between original and new table cell entries is decreased to remain below given target bounds b^*. The basic idea is to modify the 3-anonymous frequency vector, such that the maximum error in the resulting table cells does not exceed some decreased bounds $\tilde{b}$. Initially, bound $\tilde{b}$ is set to $b + g$. They are decreased until $|(Ay^0)_t - (Ay^*)_t| > \tilde{b}_t$ for a certain number of table cells, which is defined by parameter *max_number_exceedings*. The reduction of the bounds depends on the current allowed bounds but is simplified in this representation. Then, the algorithm runs through the whole frequency vector and each three values are considered and modified by Algorithm 4. Analogously to Algorithm 2, again all possible changes between 0 and ±3 are analyzed. In contrast to Algorithm 2, the property of 3-anonymity is preserved during all modifications. The procedure is repeated until the target bounds b^* are respected by y^*, too many table cells exceed the decreased bounds or a maximum number of iterations is reached, which is defined by parameter *max_number_iterations*.

```
Input: original frequency vector $y^0$, 3-anonymous frequency vector $y$, target
       bounds $b^*$, parameters $max\_number\_exceedings$,
       $max\_number\_iterations$, $max\_total\_number\_exceedings$.
Output: 3-anonymous and improved frequency vector $y^*$.
Initialization: $y^* \leftarrow y$, $num\_iter = 0$, current bounds $\tilde{b} \leftarrow b + g$;
while there is an index $t$ with $\tilde{b}_t > b^*_t$ and there are less than
  $max\_total\_number\_exceedings$ indices with $|(Ay^0)_t - (Ay^*)_t| > \tilde{b}_t$ do
    Decrease bounds $\tilde{b}_t$ by 1 until further decreasing would lead to
      $|(Ay^0)_t - (Ay^*)_t| > \tilde{b}_t$ for at least $max\_number\_exceedings$ indices or
      $\tilde{b}_t = b^*_t$;
    for $i = 1, \ldots, len(y^*)-2$ do
        Run Algorithm 4 with input $y^0$, $y^*$, $\tilde{b}$ and $[i, i+1, i+2]$ and get the
          possibly changed frequency vector $y^*$;
    end
    $num\_iter = num\_iter + 1$;
    if $num\_iter = max\_number\_iterations$ then
        break;
    end
end
return $y^*$
```

Algorithm 3: Improvement of the solution.

Description of Algorithm 4 The selection procedure for the improvement of the solution is similar to the procedure described in Algorithm 2. However, in the improvement selection procedure, candidates are only added to the set AC (“all candidates”), if they fulfill the property of 3-anonymity. Thus, a new improved frequency vector y^z never contains values of 1 or 2, i.e. there are no confidentiality problems in y^z. Instead, to find the improved y^z, the minimum number of table cells that exceed the new (decreased) bound b is taken into account. These are table cells where

$f_t = |(Ay^0)_t - (Ay)_t| > b_t$. The set C_min_exceedings contains all elements in AC with minimum number of table cells exceeding the given bound. Analogously to the previous selection procedure, the set C_min_penalty is a subset of C_min_exceedings containing the frequency vectors with minimal value for a penalty function $\tilde{s}$. In the improvement step, another penalty function is used in the SAFE method, namely $\tilde{s} := \sum_{t \in \mathcal{T}} \tilde{s}_t$, where

$$\tilde{s}_t = \begin{cases} 500, & \text{if } f_t = b_t, \\ 9, & \text{if } f_t = b_t - 1, \\ 4, & \text{if } f_t = b_t - 2, \\ 1, & \text{if } f_t = b_t - 3, \\ 0, & \text{else.} \end{cases}$$

With this penalty function, candidates with a maximum error equal to the allowed bounds will be changed preferably. Again, from the elements with minimum penalty, in the set C_min_sum_error the frequency vectors with minimum sum of absolute errors in the table cells are collected. If this set is non-empty, an arbitrary frequency vector in this set is returned, otherwise the unmodified working frequency vector y is returned.

```
Input: original frequency vector $y^0$, 3-anonymous frequency vector $y$, bound
       $b$, indices $[i_1, i_2, i_3]$, penalty function $\tilde{s}$.
Output: (improved) 3-anonymous frequency vector $y^z$.
Initialization: AC ← $\emptyset$, c ← $y$, $y^z$ ← $y$;
for $(h_1, h_2, h_3) \in \{-3, -2, \dots, 2, 3\}^3$ do
    $c_{i_j} \leftarrow y_{i_j} + h_j$ for $j = 1, 2, 3$;
    if $c_{i_j} \geq 0$ and $|c_{i_j} - y^0_{i_j}| \leq$ allowed_deviation for $j = 1, 2, 3$ then
        if $c_{i_j} \notin \{1, 2\}$ for $j = 1, 2, 3$ then
            AC ← AC $\cup$ {c};
        end
    end
end
C_min_exceedings ← {c in AC: #exceedings of bound $b$ in the table cells
    corresponding to $c$ is minimal};
C_min_penalty ← {c in C_min_exceedings: penalty function $\tilde{s}$ is minimal};
C_min_sum_error ← {c in C_min_penalty: sum of absolute errors in control
    tables is minimal};
if C_min_sum_error $\neq \emptyset$ then
    $y^z$ ← first(C_min_sum_error);
return $y^z$
```

Algorithm 4: Selection rule for improvement step.

Chapter 3

Analysis of the SAFE Method

In the previous chapter, we have described the SAFE method, which was used as anonymization algorithm in the German census in 2011. The underlying SAFE-basic problem aims to minimize the maximum error in a set of frequency tables. In this chapter, we investigate the complexity of this problem and analyze the SAFE method. In Section 3.1, we prove that a slightly modified version of the SAFE-basic problem is $\mathcal{NP}$-hard. This result justifies heuristic approaches. In Section 3.2, we discuss that the SAFE method includes several aspects, which are not part of the underlying optimization program. Hence, we propose a modification of the SAFE-basic problem.

3.1. Complexity Analysis

In the following, we show that a slightly modified version of the SAFE-basic problem (2.1.1)–(2.1.5) is $\mathcal{NP}$-hard. Therefore, it is reasonable to investigate and develop heuristic approaches to find a k-anonymous microdata set.

Known complexity results for k-anonymity We first provide an overview of known complexity results in this research area. Note that complexity results found in literature correspond to different underlying optimization problems as different approaches to find k-anonymous microdata are used. As discussed in Chapter 1.2, the first approaches used data suppression. Meyerson and Williams (2004) show that if the domain size of the attribute values is not restricted, i.e. $|V(j)| > n$ is allowed for all $j \in \mathcal{J}$, it is $\mathcal{NP}$-hard to find a k-anonymous microdata set by suppression. The corresponding decision problem is as follows. Let Y^0 be a microdata set of n data vectors. Let $\star$ be a fresh symbol not contained in any value set $V(j)$. A function $s\colon V^n \to \bigtimes_{j\in\mathcal{J}}(V(j)\cup\{\star\})$ is called a *suppressor*, if $s(Y^0)_{ij} \in \{Y^0_{ij}, \star\}$ holds for all data rows $i \in \mathcal{I}$ and all attributes $j \in \mathcal{J}$. A value Y^0_{ij} is called *suppressed* if $s(Y^0)_{ij} = \star$. Given the microdata set Y^0 and a scalar $K > 0$, is there a suppressor s, such that $s(Y^0)$ is k-anonymous and the total number of vector coordinates suppressed by s is at most K? Aggarwal et al. (2005) even show $\mathcal{NP}$-hardness for the special case that $|V(j)| = 3$ for all $j \in \mathcal{J}$, which strengthens the previous complexity result.

In the literature, also achieving k-anonymity by generalization is a well-studied field. For instance, Byun et al. (2007) prove that the decision problem to the *k-member clustering problem* is $\mathcal{NP}$-complete. In a generalization scheme, the data records are replaced by more general values, i.e. intervals or sets of values that contain the true value, such that at least k data rows are identical to each other. For data suppression, the number of suppressed values is decisive for the information loss. For data generalization, the degree of generalization determines the information loss. They regard generalization as a clustering problem with minimum-size constraint for the clusters. The degree of generalization depends on the decision which data records are assigned to the same cluster. For each cluster, the generalized values are selected as the smallest interval or set, such that all true values of the cluster are included. The information loss is defined as the degree of generalization and depends on the similarity between the cluster elements. We denote the information loss by $IL(C)$ for cluster C. The decision problem is as follows. Given n data records and a scalar $K > 0$, is there a clustering $\mathcal{C}^* = \{C_1, \dots, C_s\}$, such that $|C_i| \geq k$ for all $i = 1, \dots, s$, and $\sum_{i=1}^{s} IL(C_i) < K$? It is noteworthy, that generalizing an attribute to the most general form is equivalent to data suppression. In their proof, Byun et al. (2007) build on the complexity results given by Aggarwal et al. (2005), and show that the decision problem for data suppression is a special case of their decision problem for data generalization.

In contrast to suppression- and generalization-based approaches, the optimization problem (2.1.1)–(2.1.5) uses *data perturbation* to achieve k-anonymity. Moreover, the information loss also takes the resulting errors in the frequency tables into account. In the next section, we show that a slight modification of the SAFE-basic problem is $\mathcal{NP}$-hard.

Complexity of modified SAFE-basic problem We analyze the complexity of a slight modification of optimization problem (2.1.1)–(2.1.5). We use the same notation as before and prove that the following problem is $\mathcal{NP}$-hard.

$$\min_{y,b} \max_{t \in \mathcal{T}} |b_t| - g_t \tag{3.1.1}$$

$$\text{s.t. } Ay + b = Ay^0, \tag{3.1.2}$$

$$\sum_{i=1}^{n_a} y_i = S, \tag{3.1.3}$$

$$y_i \in \mathbb{N}_0 \setminus \{1, 2\} \; \forall i \in \mathcal{I}, \tag{3.1.4}$$

$$b_t \in \mathbb{Z} \qquad \forall t \in \mathcal{T}, \tag{3.1.5}$$

where $S \in \mathbb{N}_0$. The modification lies in Constraint (3.1.3), where the right-hand side equals a constant S instead of the sum over all entries of the original frequency vector y^0. By using $S = \sum_{i=1}^{n_a} y_i^0$, we see that optimization problem (2.1.1)–(2.1.5) is included in the more general variant. Note that in general, a subproblem of an $\mathcal{NP}$-hard problem does not have to be $\mathcal{NP}$-hard itself.

For the proof of $\mathcal{NP}$-hardness, we first need to define the underlying decision problem. We call the decision problem referring to problem (3.1.1)–(3.1.5) *3-ANONYM*. It is given by the following description.

3-ANONYM

Given: Matrix $A \in \{0,1\}^{|\mathcal{T}| \times n_a}$, vectors $y^0 \in \mathbb{N}_0^{n_a}$, $g \in \mathbb{Z}^{|\mathcal{T}|}$, and scalars $S, K \in \mathbb{N}_0$.

Question: Are there vectors $y^* \in (\mathbb{N}_0 \setminus \{1,2\})^{n_a}$ and $b^* \in \mathbb{Z}^{|\mathcal{T}|}$, such that Constraints (3.1.2)–(3.1.5) and $\max_{t \in \mathcal{T}} |b_t^*| - g_t \leq K$ hold?

We reduce 3-ANONYM to the 3-bounded 3-dimensional matching problem. We recall the underlying decision problem, which we call 3DM-3.

3DM-3

Given: Disjoint sets W, X, Y with $|W| = |X| = |Y| =: q$ and a set $M \subset W \times X \times Y$, where each element of W, X and Y occurs in at most three triples of M, respectively.

Question: Is there a subset $M' \subset M$ such that $|M'| = q$ and no two elements agree in any coordinate?

It is shown by Garey and Johnson (1979) that the 3-dimensional matching is $\mathcal{NP}$-complete even in the 3-bounded case. We use this complexity result and show the following theorem by reduction.

Theorem 3.1.1. *3-ANONYM is $\mathcal{NP}$-complete.*

Proof. Since a non-deterministic algorithm need only guess vectors y^* and b^* and check in polynomial time whether all constraints and the inequality $\max_{t \in \mathcal{T}} |b_t^*| - g_t \leq K$ are fulfilled, 3-ANONYM is in $\mathcal{NP}$.

Let W, X, Y and $M \subset W \times X \times Y$ be the sets of an arbitrary instance of 3DM-3 and $q := |W| = |X| = |Y|$. We construct a matrix A, vectors y^0, g and scalars S, K, such that Constraints (3.1.2)–(3.1.5) are fulfilled and $\max_{t \in \mathcal{T}} |b_t^*| - g_t \leq K$ holds if and only if M contains a feasible matching.

We construct a vector $y^0 \in \mathbb{N}_0^{q^3}$ such that each element corresponds to a triple in $W \times X \times Y$. We use the representation of data as a non-abbreviated frequency vector, where also entries of zero are allowed. Define $\mathcal{I} := \{1, \ldots, q^3\}$ and let $\phi\colon W \times X \times Y \to \mathcal{I}$ be a bijection. We denote the index set corresponding to set M by $\mathcal{I}_M := \{i \in \mathcal{I}\colon \phi^{-1}(i) \in M\}$ and its complement by $\mathcal{I}_{\bar{M}} := \mathcal{I} \setminus \mathcal{I}_M$. In the end, a solution y^* corresponds to a matching, such that $y_i^* = 3$ whenever the triple $\phi^{-1}(i)$ is in the matching and $y_i^* = 0$ otherwise.

Set

$$y_i^0 = \begin{cases} 1, & \text{if } i \in \mathcal{I}_M, \\ 0, & \text{otherwise.} \end{cases}$$

Next, we construct the control matrix $A \in \{0,1\}^{(3q+q^3)\times q^3}$, such that the matrix corresponds to all one- and the three-dimensional table. So, the control matrix is a block matrix

$$A = \left(\begin{array}{c} A_1 \\ \hline A_2 \\ \hline A_3 \\ \hline A_4 = I \end{array} \right),$$

where $A_1, A_2, A_3 \in \{0,1\}^{q\times q^3}$ generate the one-dimensional tables and I is the identity matrix in $\mathbb{R}^{q^3\times q^3}$, which generates the three-dimensional table. One can interpret the sets W, X and Y as attributes, where the elements correspond to the attribute values. Hence, control tables A_1, A_2, and A_3 map the number of occurrences of elements $w_i \in W$, $x_i \in X$, $y_i \in Y$, respectively. Since we regard the 3-dimensional matching problem in the 3-bounded variant, $(Ay^0)_i \in \{1,2,3\}$ holds for all $i = 1,\ldots,3q$. For cells in the 3-dimensional table $(Ay^0)_i \in \{0,1\}$ holds for all $i = 3q+1,\ldots,3q+q^3$ because they coincide with the vector y^0 itself.

Now, for $t = 1,\ldots,3q$ set

$$g_t = \begin{cases} 2, & \text{if } (Ay^0)_t = 1, \\ 1, & \text{if } (Ay^0)_t = 2, \\ 0, & \text{otherwise} \end{cases}$$

and for $i = 1,\ldots,q^3$ set

$$g_{3q+i} = \begin{cases} 2, & \text{if } i \in \mathcal{I}_M, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, set $K = 0$ and $S = 3q$.

Next, we show that the instance of 3-ANONYM constructed in this way is solvable if and only if M contains a matching.

First, we show that if there is a feasible solution to the constructed instance, then there exists a matching in M. For a solution (y^*, b^*), Equations (3.1.2) must hold. In particular, it must hold

$$\begin{aligned} & & (A_4 y^*)_i + b^*_{t_i} &= (A_4 y^0)_i \\ \Leftrightarrow & & (I y^*)_i + b^*_{t_i} &= (I y^0)_i \\ \Leftrightarrow & & y^*_i + b^*_{t_i} &= y^0_i \end{aligned} \tag{3.1.6}$$

for all $i \in \mathcal{I}$, where $t_i \coloneqq 3q + i$.

For all $i \in \mathcal{I}_{\bar{M}}$, which are indices not belonging to any triple in the set M, Equations (3.1.6) mean $y_i^* + b_{t_i}^* = 0$ by the definition of y^0.

Since $K = 0$, and $g_{t_i} = 0$ for all $i \in \mathcal{I}_{\bar{M}}$, a solution must also satisfy

$$\max_{i \in \mathcal{I}_{\bar{M}}} |b_{t_i}^*| = \max_{i \in \mathcal{I}_{\bar{M}}} |b_{t_i}^*| - g_{t_i} \leq \max_{t \in \mathcal{T}} |b_t^*| - g_t \leq 0.$$

Therefore, $b_{t_i}^* = 0$ for all $i \in \mathcal{I}_{\bar{M}}$ and a solution y^* must fulfill $y_i^* = 0$ for all $i \in \mathcal{I}_{\bar{M}}$.

Now, we consider all indices belonging to triples in the set M. For all $i \in \mathcal{I}_M$, Equations (3.1.6) are equivalent to $y_i^* + b_{t_i}^* = 1$ by the definition of y^0. So,

$$y_i^* = 1 - b_{t_i}^*$$

for all $i \in \mathcal{I}_M$ must hold. Here, we get the inequalities

$$\max_{i \in \mathcal{I}_M} |b_{t_i}^*| - 2 \leq 0 \quad \Leftrightarrow \quad -2 \leq b_{t_i}^* \leq 2 \quad \forall i \in \mathcal{I}_M.$$

Together with Constraint (3.1.4), it follows that $y_i^* \in \{0, 3\}$ for all $i \in \mathcal{I}_M$.

Consider Equations (3.1.2) corresponding to the one-dimensional tables. As mentioned before, it must hold $(Ay^*)_t + b_t^* = (Ay^0)_t \in \{1, 2, 3\}$ for all $t = 1, \ldots, 3q$. Since $y_i^* \in \{0, 3\}$ for all $i \in \mathcal{I}$, we obtain $(Ay^*)_t \in \{0, 3, 6, \ldots\}$ for all $t = 1, \ldots, 3q$. By construction, $0 \leq g_t \leq 2$, so $-2 \leq b_t^* \leq 2$ for all $t = 1, \ldots, 3q$ for a solution in order to obey $\max_t |b_t^*| - g_t \leq 0$. Therefore, we can strengthen the statement to $(Ay^*)_t \in \{0, 3\}$.

Given that $y_i^* = 0$ for all $i \in \mathcal{I}_{\bar{M}}$ and $y_i^* \in \{0, 3\}$ for all $i \in \mathcal{I}_M$, we can conclude due to Equation (3.1.3) with $S = 3q$ that it must hold $y_i^* = 3$ for exactly q indices in $\mathcal{I}_M$. Now assume, there are two indices $i_1, i_2 \in \mathcal{I}_M$ such that $y_{i_1}^* = y_{i_2}^* = 3$ and the triples $\phi^{-1}(i_1)$ and $\phi^{-1}(i_2)$ share at least one value z_i. Then, there is a one-dimensional table counting the occurrences of z_i and thus, the corresponding table cell $(Ay^*)_i \geq 6$, which is a contradiction to $(Ay^*)_t \in \{0, 3\}$ for all $t = 1, \ldots, 3q$. In conclusion, there cannot be two elements of the vector with value 3 corresponding to non-disjoint triples. As there must be q selected elements, they correspond to a matching in M.

Now, we consider the other direction. If a matching M' is found, then setting to 3 all elements of the vector y^* corresponding to included triples will give a solution of the 3-ANONYM-instance choosing b^* appropriately.

□

The result of $\mathcal{NP}$-completeness for 3-ANONYM justifies that we focus on heuristic approaches to the problem for larger instances.

3.2. Discussion of the SAFE Method

In this section, we discuss the SAFE method in comparison to the underlying SAFE-basic problem. We report several observations and propose a modification of the objective function. The benefit of the proposed modification is that the deviation from the original frequency vector is included in the objective.

First, in contrast to the optimization problem, the SAFE algorithm does not exactly follow the objective function. While the optimization problem minimizes the maximum absolute error in the resulting frequency tables, the SAFE method also implements selection rules that take other criteria into account. These criteria include the deviation from the original frequency vector, the absolute errors, and the mean errors in the tables. Since these criteria are relevant in practice, it is beneficial for the performance of the algorithm to include these additional measurements but this is not represented in the SAFE-basic problem.

Second, we inspect the choice of the objective function. The maximum error in the control tables is to be minimized. However, this criterion does not ensure that the resulting k-anonymous microdata is as similar as possible to the original microdata set. Whenever there are different solutions with the same objective value, the k-anonymous microdata set with the highest similarity to the original microdata set should be chosen to minimize the information loss due to anonymization. The following example illustrates problems caused by using exclusively the maximum error in the objective function to determine the k-anonymous microdata set.

Example 3.2.1. Let the original microdata set be given as depicted in Table 3.1a. Let all one-dimensional tables be the control tables that are taken into account in the anonymization process. They are depicted in Table 3.1b.

Sex	*Citizenship*
female	Germany
female	Germany
male	Others
male	Others

(a) Original microdata set.

Sex	Counts
female	2
male	2

Citizenship	Counts
Germany	2
Others	2

(b) Control tables from the original microdata set.

Figure 3.1.: An exemplary microdata set and corresponding frequency tables.

Assume, we want to publish a 2-anonymous microdata set from the given one. In this case, the original microdata set already satisfies 2-anonymity and, thus, is an optimal solution to problem (2.1.1)–(2.1.5). However, also the 2-anonymous microdata depicted in Figure 3.2a is optimal with respect to the maximum absolute error. The control tables remain unchanged as depicted in Table 3.2b. Therefore, this microdata set could equally be chosen. However, since the original microdata set already satisfies 2-anonymity, choosing an alternative optimal solution introduces unnecessary information loss.

To avoid the described problem and minimize the information loss, we propose a modification of the program to include a penalization for the differences between the

Sex	*Citizenship*
female	Germany
female	Germany
male	Others
male	Others

(a) 2-anonymized microdata set.

Sex	Counts
female	2
male	2

Citizenship	Counts
Germany	2
Others	2

(b) Control tables from the 2-anonymized microdata set.

Figure 3.2.: An optimal solution to the SAFE-basic problem.

original and the new microdata set as additional term to find an optimal k-anonymous microdata set. The objective function we propose is

$$\left(\max_{t \in \mathcal{T}} |b_t| - g_t\right) + w \sum_{i=1}^{n_a} |y_i - y_i^0|.$$

A parameter $w > 0$ is introduced to weight the terms, where a higher value for w results in a larger weight of the absolute differences between original and new frequency vector. However, the main focus should remain on the maximum error. To find a k-anonymous microdata set with minimal maximum error while taking the deviations between the frequency vectors into account, we choose $w = \frac{1}{\sum_{i=1}^{n_a} y_i^0}$. The results of this method are shown in Chapter 6.

Properties of a SAFE solution We observed that the SAFE heuristic does not necessarily yield a feasible solution to the SAFE-basic problem due to Constraint (2.1.3). This constraint ensures that the number of statistical objects does not change. The SAFE heuristic emphasizes that a solution remains close to the original frequency vector and also that the resulting frequency tables remain close to the original tables but it does not ensure that the number of statistical objects remains the same. In the computational experiments described in Section 6.2.7, we see instances, for which a solution of SAFE does not preserve the original number of data rows. In this thesis, we propose another heuristic approach to achieve k-anonymous microdata. Using this method, the number of data rows is preserved, i.e Constraint (2.1.3) is satisfied.

Another interesting aspect is the handling of zeros in the frequency vector and in the resulting frequency tables. Structural zeros occur in frequency tables, where the value combination corresponding to the cell is unreasonable. By preserving structural zeros, data plausibility is maintained. The SAFE method is based on a frequency vector that contains all *occurring* attribute combinations and modifies this frequency vector to obtain k-anonymous microdata. Therefore, a combination that does not exist in the original microdata set will not be created during the SAFE algorithm. However, after the anonymization process, some originally occurring attribute combinations are changed to a frequency of zero. Thus, in the resulting k-anonymous microdata set it is indistinguishable whether a zero is a structural zero or was introduced during the anonymization process. Note that even though the SAFE method uses the abbreviated frequency vector, which only contains occurring combinations, the frequency vector after anonymization, which is published, contains *all* possible combinations.

3.3. Summary

In summary, we have proven $\mathcal{NP}$-completeness for 3-ANONYM, which justifies heuristic approaches to find k-anonymous microdata sets by data perturbation. We identified limitations in the well-known heuristic SAFE approach with respect to deviations from the underlying SAFE-basic problem and preserving the number of statistical objects, and in the SAFE-basic problem with respect to minimizing the information loss. Therefore, we propose a different approach to the microaggregation problem for nominal data, which is outlined in the following.

Chapter 4

Partitioning Step

In this chapter, we describe a new approach to the first step of a microaggregation process to obtain k-anonymity – the partitioning step. In the partitioning step, data records are assigned to clusters, which are then used in the subsequent aggregation step. During the microaggregation procedure, the microdata set to be anonymized is changed cluster by cluster. In each cluster, the individual data records are replaced such that they are identical to each other. To ensure k-anonymity, the clusters must satisfy a minimum size of k. Since all individual records in the same cluster are changed in the same way, information loss is reduced if similar data records are assigned to the same cluster. To find such clusters with similar elements, published approaches directly extend the problem from numerical to categorical data like the work by Torra (2004) and Marés and Torra (2012). However, in contrast to that, Byun et al. (2007) formulate a clustering-based problem to approach the partitioning step. We go in this direction and present a novel method, which respects the similarity between all data records that are potentially assigned to the same cluster.

The first part of this chapter builds on the work of Castro et al. (2022). They focus on the microaggregation problem *for numerical data* and formulate an integer program for the partitioning step, for which except for the integrality constraints the constraints and the objective function are linear. This program has exponentially many variables. Relaxing the problem into a linear program, the structure is suitable for a column generation scheme. This method exploits the fact that only very few of the variables are actually included in an optimal solution. In many practical applications, the column generation scheme can handle linear programs with a large number of variables. To this end, the algorithm iteratively adds variables by using the dual information arising from a solution to the restricted master problem, which includes a subset of the variables. The general scheme of a column generation procedure is presented in Section 1.2.3. However, it is not clear how best to obtain an *integer* solution to the original master problem. Castro et al. (2022) use two heuristics to obtain an integer solution based on rounding. In each iteration of the column generation scheme, a current fractional solution is obtained. The two heuristics are used to construct an integer solution from the current fractional solution. The first heuristic works with simple rounding. Iteratively, an integer solution is constructed by rounding the values of the variables with the largest value in the current fractional solution. In this process, variables are neglected, which would lead to violations in the constraints when being added to the current integer solutions. The second heuristic is also based on rounding but computes the integer solution with respect to the edges. This heuristic starts with a clustering of all singletons as clusters and iteratively merges clusters according to the values in the fractional solution with respect to the edges. Their computational results show that in most cases the second heuristic finds a better integer solution. The computational results are done on numerical data. They compare their method to two popular heuristics for numerical data, MDAV and V-MDAV. While the MDAV algorithm only allows fixed size clusters, the V-MDAV heuristic extends MDAV and also allows variable size clusters. The solutions of both methods are included as initial solutions to the method proposed by Castro et al. (2022). Therefore, the returned solution is at least as good as the solutions provided by the MDAV heuristic and the V-MDAV heuristic. However, their computational results even show strong improvements in the relative gap between the heuristically found objective value and optimal value of the relaxed integer program. In conclusion, their work shows that it can be beneficial to include a column generation scheme into the partitioning step in a microaggregation procedure. Thus, further research in this direction and, especially, the extension to categorical data seem to be promising.

In general, variables that are included in fractional solutions do not have to be useful in integer solutions. Therefore, our work not only extends the method from numerical data to nominal data but also uses a different approach to obtain an integer solution from the fractional solution. This approach replaces the rounding heuristics used by Castro et al. (2022). The second part of this chapter describes our approach, which builds on the work by Zhao et al. (2018). Instead of rounding the fractional solution, the provided dual information is used as guidance to construct an integer solution. For fractional programs it is well-known that the so-called duality gap between an optimal solution to the linear program and an optimal solution to its dual problem is zero. For integer programs, this is in general not the case. However, Larsson and Patriksson (2006) show optimality results on integer programs, which are similar to optimality results on linear programs but do not assume the duality gap to be zero. In the following, we construct an integer solution by solving an auxiliary problem, which is based on these optimality conditions. This problem includes the dual information. A solution to the partitioning problem is constructed by iteratively adding clusters that minimize this auxiliary problem.

This chapter is structured as follows. The fact that the partitioning problem was initially considered for numerical data influenced later work on categorical data. We motivate in Section 4.1 that for categorical data it is preferable to consider the distances between *all* data records in a potential cluster rather than only considering distances to a representative of a cluster. Note that in this aspect, categorical data behave differently from numerical data. In Section 4.2 we show a problem formulation for finding clusters for the partitioning step, which is suitable for a column generation scheme. We see the application of a column generation scheme, which leads to a fractional solution in Section 4.3. For numerical data, Castro et al. (2022) further considered this problem. We build on the idea by Zhao et al. (2018) to construct an integer solution. We present a procedure that uses the derived dual information in Section 4.4. In Section 4.5, we conclude this chapter with a summary.

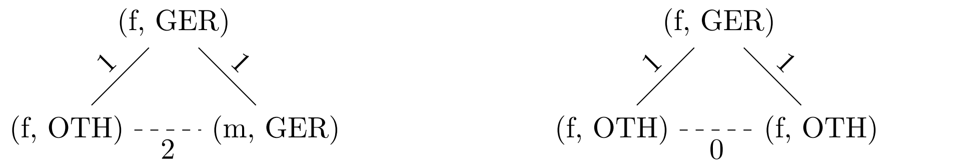
4.1. Motivation for Clustering-based Partitioning

In this section, we examine a fundamental difference between numerical and categorical data with respect to the similarity between data records, which is taken into account when finding a suitable data set partitioning. For numerical data, it is common to use the mean values of the cluster elements as representatives as discussed in Section 1.2.2. Each data record is assigned to the cluster with the closest representative. Partitioning all records into clusters such that distances between each individual data record and the mean of the corresponding cluster are minimal also yields a solution that minimizes the overall dissimilarity in each cluster. The first extensions to categorical data by Huang (1998) adopted the procedure directly by replacing the mean with the plurality mode for nominal data. The dissimilarity for nominal data is measured by using the Hamming-distance. However, nominal data records can differ from each other even though they are close to their representative (in terms of the Hamming-distance). The following example illustrates this situation of high dissimilarity within a cluster even though data records are close to the representative with respect to the Hamming-distance.

Example 4.1.1. Let $\mathcal{J} = \{sex, citizenship\}$ be the set of attributes with values $V(sex) = \{\mathrm{f}, \mathrm{m}\}$ and $V(citizenship) = \{\text{GER, OTH}\}$, respectively. Let d denote the Hamming-distance.

We consider a cluster $C_1 \coloneqq \{(\text{f, GER}), (\text{f, OTH}), (\text{m, GER})\}$. Assume data record (f, GER) is selected as representative. The sum over all Hamming-distances only to the representative is 2. This case is depicted in Figure 4.1a by the solid lines. Let $C_2 \coloneqq \{(\text{f, GER}), (\text{f, OTH}), (\text{f, OTH})\}$ and data record (f, GER) be the representative. The sum over all Hamming-distances only to the representative is again 2. This case is depicted in Figure 4.1b by the solid lines. However, when we consider the distances between *all* data records the sum over all Hamming-distances is 4 for cluster C_1 but only 2 for cluster C_2. This case is depicted in Figure 4.1 by solid and dashed lines. Thus, cluster C_2 would be preferable due to its homogeneity.

In consequence, it is necessary for nominal data to take the distances between all data records, which are assigned to the same cluster, into account instead of only considering the distances to a representative of the cluster, which is sufficient for numerical data.



(a) Hamming-distances in cluster C_1.

(b) Hamming-distances in cluster C_2.

Figure 4.1.: Hamming-distances in two exemplary clusters with the same representative (f, GER).

4.2. Problem Formulation

In this section, we present an integer program for the partitioning problem in a microaggregation procedure, which takes the distances of all data records to be assigned to the same cluster into account and which is the basis of the further work in this thesis. The program formulates the problem of partitioning the given microdata set into clusters of *similar* data records with minimum cluster size. Since we consider nominal data, we choose the Hamming-distance to measure similarity, which counts the number of mismatches between two data records and is defined in Section 1.1.

There are different names for the partitioning step. In the context of microaggregation, it is common to use the term *partitioning*. However, in the context of data analysis and graph theory one also encounters the term *clustering*. In this thesis, we will use both terms synonymously.

As discussed before, for categorical data it is not sufficient to consider only the distances between data rows and single cluster representatives because even though data rows can be close to their representative, they still can significantly differ from each other. Therefore, we propose an approach that takes the distances between all data rows in a cluster into account. For that purpose, we interpret the microdata set as a complete graph $G = (\mathcal{I}, E)$, where each vertex in the graph corresponds to a data record in the microdata set. Each edge $e = (i, l)$ is given a weight w_{il}, which equals the dissimilarity between the two incident data rows. Corresponding to the data rows Y^0_{i*} and Y^0_{l*}, the weight is given by $w_{il} = \sum_{j \in \mathcal{J}} \delta(Y^0_{ij}, Y^0_{lj})$. We refer to the definition of the Hamming-distance in Section 1.1 and denote the indicator showing the mismatches by $\delta(\cdot, \cdot)$. The aim is to partition the vertex set of the graph. Since the graph is complete, each subset of vertices is a *clique*. A clique $C \subset \mathcal{I}$ is a subset of vertices such that every two distinct vertices are adjacent. That is, the subgraph induced by a clique C is complete. Let $E(C)$ denote the edge set of the subgraph induced by $C \subset \mathcal{I}$. We define the weight of a clique C to be the sum of the weights of all edges in $E(C)$, i.e. $w_C := \sum_{i,l \in C} w_{il}$. In this view, the task of finding a suitable partitioning of the microdata set is equivalent to finding a weight-minimum clustering of a complete graph with a lower bound on the cluster sizes and the terms *clique* and *cluster* can be used synonymously in this context.

By the nature of the problem, the cluster sizes are restricted to a minimum of k to obtain k-anonymity. Moreover, as already discussed by Domingo-Ferrer and Mateo-Sanz (2002), the cluster sizes can also be bounded from above by $2k-1$ because clusters of a size greater than or equal to $2k$ can be split into smaller clusters with at least k elements without increasing the weight of the clustering. Therefore, it is sufficient to consider clusterings with clusters of size between k and $2k-1$. The set of all clusters with size between k and $2k-1$ can be written as $\mathcal{C} \coloneqq \{C \subset \mathcal{I} \colon k \leq |C| \leq 2k-1\}$. Without loss of generality, we assume an optimal clustering to be a subset $\mathcal{C}^* \subset \mathcal{C}$. The formulation we show in this section is written in a column-oriented form as done by Castro et al. (2022), who focused on numerical data. In the problem formulation, the difference between handling numerical and categorical data lies in the dissimilarity measurement used, i.e. in the edge weights.

To describe a cluster $C \in \mathcal{C}$ let

$$a_{iC} \coloneqq \begin{cases} 1, & \text{if } i \in C, \\ 0, & \text{else} \end{cases}$$

for each data row $i \in \mathcal{I}$ be binary indicators, which show whether data row i is contained in cluster C. We use binary variables x_C for all clusters $C \in \mathcal{C}$, which will indicate whether cluster C is contained in a solution $\mathcal{C}^*$ or not.

The integer program underlying our approach to the partitioning step of the microaggregation procedure is given by

$$\min_x \sum_{C \in \mathcal{C}} w_C x_C \tag{4.2.1}$$

$$\text{s.t.} \sum_{C \in \mathcal{C}} a_{iC} x_C = 1 \ \forall i \in \mathcal{I}, \tag{4.2.2}$$

$$x_C \in \{0,1\} \qquad \forall C \in \mathcal{C}. \tag{4.2.3}$$

The objective function in (4.2.1) is the sum of weights of all clusters that are contained in a solution. Thus, the goal of the problem is to minimize the weight of the resulting clustering. Constraints (4.2.2) ensure that each data row is contained in exactly one cluster.

Both the objective function and Constraints (4.2.2) are linear. Nevertheless, the optimization problem also has exponentially many variables because there are as many variables as elements in the set $\mathcal{C}$. The set $\mathcal{C}$ consists of all clusters with a size between k and $2k-1$ and, thus, has $\sum_{s=k}^{2k-1} \binom{n}{s}$ many elements, where $n \coloneqq |\mathcal{I}|$. However, the problem structure can be used in a column generation scheme when the linear relaxation of the problem is considered. A column generation scheme has proven useful for many very large linear problems in practice. Therefore, it seems to be a promising approach to find a fractional solution to the linear relaxation. In the next section, we outline the application of the column generation scheme.

4.3. Finding a Fractional Solution

The optimization program (4.2.1)–(4.2.3) describes the problem of finding a weight-minimum clustering such that each cluster has a size between k and $2k-1$. Since the problem is an integer program with exponentially many variables, it is challenging to solve. However, the problem structure allows the use of a column generation scheme if the integrality constraints are neglected. We have described how column generation addresses problems with such a structure in Section 1.1. In this section, we discuss the linear relaxation of problem (4.2.1)–(4.2.3) and show a column generation approach to obtain a fractional solution. In the linear relaxation, the integrality constraints (4.2.3) are replaced by $x_C \in [0,1]$ for all $C \in \mathcal{C}$. The resulting optimization problem is a linear program with an exponential number of variables.

In the context of column generation, the linear relaxation of problem (4.2.1)–(4.2.3) is called the *master problem*. The idea of the method is to replace the variable set of the master problem with a small subset such that the remaining linear program with less variables is easily solved. The resulting problem is called the *restricted master problem* (RMP). Then, variables are added to the restricted master problem until an optimal solution to the master problem can be obtained.

In the restricted master problem, the variable set $\mathcal{C}$ is replaced by a small subset $\mathcal{C}' \subset \mathcal{C}$. The restricted master problem is written as

$$\min_x \sum_{C \in \mathcal{C}'} w_C x_C \tag{4.3.1}$$

$$\text{s.t.} \sum_{C \in \mathcal{C}'} a_{iC} x_C = 1 \quad \forall i \in \mathcal{I}, \tag{4.3.2}$$

$$x_C \geq 0 \qquad \forall C \in \mathcal{C}'. \tag{4.3.3}$$

Instead of writing $x_C \in [0,1]$, we can write $x_C \geq 0$ because of Constraints (4.3.2) and a_{iC} being defined to be binary.

The restricted variable set $\mathcal{C}'$ can be initialized with any heuristic solution. However, we decided to even choose a simpler initialization by starting with a singleton $\mathcal{C}' = \{C\}$, where C is defined by $a_{iC} = 1$ for all $i \in \mathcal{I}$. In other words, C is a cluster containing *all* data records. Even though this initial restricted variable set is not a subset of the original variable set, it has the advantage of being very simple and is a feasible solution in terms of Constraints (4.3.2). Without loss of generality, this cluster is not included in an optimal solution to the master problem since it can be split into clusters of size between k and $2k-1$ without increasing the objective value.

In the column generation scheme, the variable set $\mathcal{C}'$ is extended iteratively. Variables, i.e. clusters $C \in \mathcal{C}$, that have a non-zero value in an optimal solution are added. Whether a variable obtains a non-zero value can be seen from the reduced costs, which are computed by using dual variables. The dual problem to the linear program above aims at finding a maximum lower bound to the objective function (4.3.1). It contains constraints for each variable corresponding to clusters $C \in \mathcal{C}'$, and variables for each constraint in (4.3.2), i.e. for each data row $i \in \mathcal{I}$. The dual program to problem (4.3.1)–(4.3.3) can be written as

$$\begin{aligned} \max_{u} \; & \sum_{i\in\mathcal{I}} u_i && && (4.3.4)\\ \text{s.t. } & \sum_{i\in\mathcal{I}} a_{iC} u_i \leq w_C && \forall C \in \mathcal{C}', && (4.3.5)\\ & u_i \geq 0 && \forall i \in \mathcal{I}. && (4.3.6) \end{aligned}$$

The reduced costs are defined by $w_C - \sum_{i\in\mathcal{I}} a_{iC} u_i$ and can be seen as the amount by which an objective coefficient w_C would have to be increased until the corresponding variable x_C is non-zero in an optimal solution. Therefore, a variable with negative reduced costs might be part of an optimal solution and is added to the variable set $\mathcal{C}'$.

Instead of computing the reduced costs explicitly for exponentially many columns, the *pricing problem* finds variables with negative reduced costs *implicitly.* We consider the formulation of the pricing problem next. It can be written as

$$\begin{aligned} \min_{v,z} \; & \sum_{e\in E} w_e z_e - \sum_{i\in\mathcal{I}} u_i v_i && && (4.3.7)\\ \text{s.t. } & z_{il} \geq v_i + v_l - 1 && \forall (i,l) \in E, && (4.3.8)\\ & k \leq \sum_{i\in\mathcal{I}} v_i \leq 2k-1, && && (4.3.9)\\ & z_e \geq 0 && \forall e \in E, && (4.3.10)\\ & v_i \in \{0,1\} && \forall i \in \mathcal{I}. && (4.3.11) \end{aligned}$$

In our application, a variable x_C in the master problem (4.2.1)–(4.2.3) corresponds to a cluster with a size between k and $2k-1$. The objective function in (4.3.7) is the reduced costs of the cluster indicated by the v_i-variables. They are computed with the help of current dual variables u_i, which are obtained by solving the current restricted master problem (4.3.1)–(4.3.3). We use binary variables v_i for all vertices $i \in \mathcal{I}$. They indicate whether a vertex $i \in \mathcal{I}$ is contained in an optimal cluster, i.e. a cluster with minimum reduced costs. Additionally, the problem contains variables z_e to indicate whether an edge $e \in E$ is contained in an optimal cluster or not. Even though it is known that each set of vertices automatically is a clique because the graph is complete, the edge variables are needed to compute the reduced costs. We denote the edge variables by $z_e = z_{il}$ for all $e = (i,l) \in E$. Constraints (4.3.8) ensure that an edge is included if both its incident vertices are. Note that due to the non-negativity of the edge weights, we do not have to explicitly consider the opposite direction, which is that an edge is not included if at least one of its incident vertices is not. Constraint (4.3.9) enforces the size of the resulting cluster to be between k and $2k-1$. It suffices to enforce non-negativity in Constraint (4.3.10) instead of $z_e \in \{0,1\}$ because of the non-negativity of the edge weights and Constraints (4.3.8) together with Constraints (4.3.11).

Also the work by Ji and Mitchell (2007) deals with a column-oriented formulation of the investigated partitioning problem. They focus on a branch-and-price scheme and how cutting planes can be included in their algorithm. However, they do not exploit that it is sufficient to limit clusters to those with a size less than or equal to $2k - 1$. Castro et al. (2022) showed a formulation of the problem using only edge variables.

As mentioned in Section 1.1, it is useful to apply heuristics for finding variables with negative reduced costs. In case of multiple heuristics, it is reasonable to apply a fast method first and only if it is not successful to apply further heuristics. Therefore, we first apply a greedy heuristic. Only if it does not return suitable variables, we apply a more costly enumeration heuristic.

The procedure of the greedy algorithm is depicted in Algorithm 5 and can be described as follows. Given the original edge weights w_e and the current dual variables u_i, the algorithm iteratively constructs clusters C. It returns a set S of clusters with negative reduced costs respecting the size-constraint, or the empty set. Returning multiple clusters, i.e. variables of the master problem, with negative reduced costs can speed up the algorithm since multiple variables are added to the restricted master problem and less iterations of the column generation scheme might be needed. However, it is also not required to add *all* variables with negative reduced costs. A parameter n_{max} determines the maximal number of clusters that are added to the set S. The set S is initialized to be the empty set. Iteratively, clusters with negative reduced costs are added, if found. To ensure that clusters distinct from each other are constructed, the added clusters are built only of vertices that are not assigned to any previous cluster. To keep track of the vertices, a set N is initialized to be the set of all data rows and will consist of all unassigned vertices. A cluster C to be added to S is created by starting with an unassigned vertex i with maximum value of the dual variable u_i. Iteratively, vertices, which are not assigned yet, are added to C. The criterion for the choice of the selected vertex is the increase of the cluster weight of C, i.e., the reduced costs, resulting from the inclusion of the vertex. In the next step, it is checked whether the reduced costs can be decreased by adding more elements to the cluster. For this purpose, extensions of the cluster to clusters of at most $2k - 1$ elements are considered. The cluster with minimum reduced costs is added to the set S, if the reduced costs are negative. The procedure is repeated on the remaining vertices until either the cluster has a non-negative weight, or fewer than k vertices remain, or the algorithm has found n_{max} clusters. In this context, the notation $\delta(C)$ denotes all edges that have one incident vertex in the set C.

If the greedy algorithm does not provide a solution, we run through an enumeration heuristic, which we depict in Algorithm 6. The algorithm starts with a vertex i with maximum dual value u_i and finds a cluster with negative reduced costs by enumeration, or returns the empty set. The $2k - 2$ closest vertices to i according to the edge weights are taken into account. All clusters of size between k and $2k - 1$ are enumerated and the corresponding reduced costs are computed. The first found cluster with negative reduced costs is returned.

```
Input: Complete graph $G = (\mathcal{I}, E)$ with edge weights $w_e$ for all $e \in E$ and
       dual variables $u_i$ for all $i \in \mathcal{I}$, parameter $n_{max}$.
Output: A set $S$ of at most $n_{max}$ clusters of size between $k$ and $2k-1$ with
        negative reduced costs, or $S = \emptyset$.
Initialization: $N = \mathcal{I}$, $S = \emptyset$;
while $|N| > k$ and $|S| < n_{max}$: do
    Find vertex $i \in N$ such that $u_i \geq u_l$ for all $l \in N$;
    Set $C = \{i\}$;
    while $|C| < k$ do
        Find $i \in \arg\min_{l \in N \setminus C} \{\sum_{e \in \delta(C): l \in e} w_e - u_l\}$;
        Set $C = C \cup \{i\}$;
    end
    Set $C' = copy(C)$;
    while $|C'| < 2k - 1$ do
        Find $i = \arg\min_{l \in N \setminus C'} \{\sum_{e \in \delta(C'): l \in e} w_e - u_l\}$;
        Set $C' = C' \cup \{i\}$;
        if $reduced\_costs(C') < reduced\_costs(C)$ then
            Set $C = C'$;
        end
    end
    if $reduced\_costs(C) \geq 0$ then
        return $S$;
    end
    Set $S = S \cup \{C\}$;
    Set $N = N \setminus C$;
end
return $S$
```

Algorithm 5: Greedy heuristic for the pricing problem.

If both heuristics do not return any cluster with negative reduced costs, the pricing problem can be solved exactly. If the optimal value is non-negative, there are no more variables with negative reduced costs. Therefore, the procedure can be aborted and an optimal solution to the restricted master problem also solves the master problem optimally. However, since we are interested in finding an integer solution, it might not be necessary to solve the fractional problem optimally. Therefore, we consider both options, terminating the column generation algorithm whenever both heuristics do not find any variables with negative reduced costs and solving the master problem to optimality. The first case is of interest because of the faster runtimes and it also provides a feasible fractional solution. We use the dual information from the obtained fractional solution to get an integer clustering.

```
Input: Complete graph $G = (\mathcal{I}, E)$ with edge weights $w_e$ for all $e \in E$ and
       dual variables $u_i$ for all $i \in \mathcal{I}$.
Output: A cluster $C$ of size between $k$ and $2k - 1$ with negative reduced
        costs, or $\emptyset$.
Find vertex $i \in \mathcal{I}$ such that $u_i \geq u_l$ for all $l \in \mathcal{I}$;
Find $2k - 2$ closest vertices $N \subset \mathcal{I}$ to $i$;
for each subset $N' \subset N$ with $k - 1 \leq |N'| \leq 2k - 2$ do
    Compute reduced costs $\sum_{e \in E(N' \cup \{i\})} w_e - u_v$;
    if reduced_costs$(N' \cup \{i\}) < 0$ then
        return $C = N' \cup \{i\}$;
    end
end
return $\emptyset$.
```

Algorithm 6: Enumeration heuristic for the pricing problem.

To give an overview of the column generation scheme that is used to find a fractional solution to the partitioning problem of interest, we summarize the procedure as follows.

1. Set $\mathcal{C}' = \{C\}$ with C being the cluster that contains all data rows $i \in \mathcal{I}$.
2. Iterate
   a) Solve RMP (4.3.1)–(4.3.3) with variable set $\mathcal{C}'$ and obtain dual solutions u_i.
   b) Find variables with negative reduced costs:
      (i) Run greedy algorithm.
      (ii) If greedy algorithm was not successful: Run enumeration heuristic.
      (iii) If enumeration heuristic was not successful: Solve pricing problem (4.3.7)–(4.3.11) to optimality.
   c) If variables S with negative reduced costs are found, set $\mathcal{C}' = \mathcal{C}' \cup S$.
      Else: break.

4.4. Finding an Integer Solution

The problem of interest in this chapter is to find a partitioning of a microdata set into clusters of size between k and $2k - 1$. In the last section, we have seen how to approach the linear relaxation of the program with a column generation scheme. In this section, we analyze how the information provided by a fractional solution can be used to heuristically find an integer solution to the original problem (4.2.1)–(4.2.3). Next, we describe an approach, which exploits the dual information obtained by a fractional solution and iteratively constructs an integer-feasible clustering of the microdata set. The method starts with an empty clustering and adds clusters iteratively until all vertices are covered. We choose the clusters to be added by solving an auxiliary problem following an idea by Zhao et al. (2018). To this end, we formulate the auxiliary problem for the application to the partitioning problem.

4.4.1. Preliminaries

We first report optimality results by Larsson and Patriksson (2006) for integer programs. These results are the basis for the auxiliary problem, which we use to find a cluster to be added. In general, for integer programs, the duality gap is expected to be non-zero. However, Larsson and Patriksson (2006) have shown conditions for the optimality of primal and dual solutions to integer programs regardless of the size of the duality gap. Based on these results, the auxiliary problem introduced by Zhao et al. (2018) aims to minimize the duality gap to find a near-optimal integer solution.

Consider the general discrete optimization problem

$$\min_x w^T x \tag{4.4.1}$$

$$\text{s.t. } Ax \geq b, \tag{4.4.2}$$

$$x \in X \tag{4.4.3}$$

with $X \neq \emptyset$ being a bounded discrete set.

Let

$$L(x, u) := b^T u + (w^T - u^T A)x,$$

then, the Lagrangian dual function for the optimization problem above can be written as

$$h(u) := \min_{x \in X} L(x, u) = \min_{x \in X} b^T u + (w^T - u^T A)x.$$

and the Lagrangian dual problem is

$$\max_{u \geq 0} h(u). \tag{4.4.4}$$

Let h^* be an optimal objective value of the Lagrangian dual problem and z^* be an optimal objective value of the primal problem (4.4.1)–(4.4.3). Then, for integer programs, the duality gap $\Gamma := z^* - h^*$ is non-zero in general. However, the following optimality result by Larsson and Patriksson (2006) holds regardless of the size of the duality gap.

Theorem 4.4.1 (Larsson and Patriksson, 2006, Theorem 3)**.** *Let $x, u \in X \times \mathbb{R}^m_{\geq 0}$. Then, the following statements are equivalent:*

1. *The primal problem* (4.4.1)–(4.4.3) *is solved by x and the dual problem* (4.4.4) *is solved by u.*

2. *There exist (ϵ, δ), so that the pair (x, u) satisfies the following conditions:*
    - a) *Lagrangian ϵ-optimality: $L(x, u) \leq h(u) + \epsilon$*
    - b) *Primal feasibility: $Ax \geq b$*
    - c) *δ-complementarity: $u^T(Ax - b) \leq \delta$.*
    - d) *$\epsilon + \delta \leq \Gamma$ with Γ being the duality gap and $\epsilon, \delta \geq 0$.*

3. *The pair (ϵ, δ) with $\epsilon, \delta \geq 0$ satisfies $\epsilon + \delta = \Gamma$, and the pair (x, u) satisfies the following saddle point like condition for the Lagrangian function*

$$L(x, v) - \delta \leq L(x, u) \leq L(y, u) + \epsilon \quad \forall (y, v) \in X \times \mathbb{R}^m_{\geq 0}.$$

The pair (ϵ, δ) depends on x and u, i.e. $\epsilon = \epsilon(x, u)$ and $\delta = \delta(x, u)$. However, for simplicity, we chose the simpler notation in this case. The theorem shows a strong relation between the duality gap Γ and the pair (ϵ, δ). Motivated by this, Zhao et al. (2018) propose an approach that finds a solution x by minimizing a convex combination of ϵ and δ when dual variables $\tilde{u}$ are given. Constraints (4.4.2) are taken into account with a penalization term in the auxiliary problem. The minimization problem is as follows.

$$\min_{x \in X} \alpha\epsilon(x, \tilde{u}) + (1 - \alpha)\delta(x, \tilde{u}) + \lambda \sum_{i \in \mathcal{I}} \max \{0, b_i - (Ax)_i\}, \tag{4.4.5}$$

where $\alpha \in (0, 1]$ is a weighting parameter for the convex combination and $\lambda > 0$ is a penalty parameter. We assume $\alpha > 0$ since otherwise it completely excludes the original objective function (4.2.1), which is not reasonable. Zhao et al. (2018) reformulate this problem. We show the rearrangement in more detail. We begin with the convex combination of ϵ and δ. By the inequalities

$$\epsilon \geq L(x, \tilde{u}) - h(\tilde{u}) = b^T\tilde{u} + (w^T - \tilde{u}^T A)x - h(\tilde{u})$$

and

$$\delta \geq \tilde{u}^T(Ax - b)$$

from Theorem 4.4.1, the convex combination of ϵ and δ is restricted by

$$\begin{aligned} \alpha\epsilon(x, \tilde{u}) + (1 - \alpha)\delta(x, \tilde{u}) &\geq \alpha(b^T\tilde{u} + w^T x - \tilde{u}^T Ax - h(\tilde{u})) + (1 - \alpha)(\tilde{u}^T Ax - \tilde{u}^T b) \\ &= \alpha w^T x - 2\alpha\tilde{u}^T Ax + \tilde{u}^T Ax + \alpha(b^T\tilde{u} - h(\tilde{u})) - (1 - \alpha)(\tilde{u}^T b) \\ &= \alpha w^T x - 2\alpha\tilde{u}^T Ax + \tilde{u}^T Ax + \text{const}(\tilde{u}). \end{aligned}$$

Since $\tilde{u}$ is given, the last part is a constant term. In the minimization problem, this term therefore can be neglected. Instead of solving problem (4.4.5) directly, the following equivalent minimization problem is solved:

$$\begin{aligned}
&\min_{x\in X} \alpha w^T x - 2\alpha\tilde{u}^T Ax + \tilde{u}^T Ax + \lambda \sum_{i\in\mathcal{I}} \max\{0, b_i - (Ax)_i\} \\
=&\min_{x\in X} w^T x - 2\tilde{u}^T Ax + \frac{1}{\alpha}\tilde{u}^T Ax + \lambda \sum_{i\in\mathcal{I}} \max\{0, b_i - (Ax)_i\} \\
=&\min_{x\in X} (w^T - \gamma\tilde{u}^T A)x + \lambda \sum_{i\in\mathcal{I}} \max\{0, b_i - (Ax)_i\} \\
=&\min_{x\in X} (w^T - \gamma\tilde{u}^T \sum_{C\in\mathcal{C}} a_{*C})x_C + \lambda \sum_{i\in\mathcal{I}} \max\left\{0, b_i - (\sum_{C\in\mathcal{C}} a_{*C}x_C)_i\right\}
\end{aligned}$$

where $\gamma := (2 - \frac{1}{\alpha}) \in (-\infty, 1]$. We replaced the term Ax by $\sum_{C\in\mathcal{C}} a_{*C}x_C$, where a_{*C} denotes the C-th column of A, to emphasize a column-oriented formulation.

In a typical program, which is suitable for column generation, the set X is a high-dimensional set. Also, the introduced minimization problem can not be solved explicitly in practice, for large instances. We therefore use an heuristic approach to the problem. The method increments single variables x_C iteratively. In each iteration, a variable with the least contribution to the introduced objective function of the auxiliary problem is incremented. For this purpose, for a current vector $\tilde{x}$ we consider the set of indices that can be potentially increased by one such that the resulting vector is again contained in the feasible set X. We define such set by

$$\mathcal{C}^{R(\tilde{x})} := \{C \in \mathcal{C} \colon (\tilde{x}_1, \dots, \tilde{x}_C + 1, \dots, \tilde{x}_{|\mathcal{C}|})^T \in X\}.$$

Let $x = (\tilde{x}_1, \dots, \tilde{x}_{\tilde{C}} + 1, \dots, \tilde{x}_{|\mathcal{C}|})^T$ for a given $\tilde{x}$. Since both vectors only differ by one in index $\tilde{C}$, we can write $\sum_{C\in\mathcal{C}}(a_{*C}x_C)_i = \sum_{C\in\mathcal{C}}(a_{*C}\tilde{x}_C)_i + a_{i\tilde{C}}$.

So, we are looking for an index $\tilde{C}$ minimizing

$$\min_{C\in\mathcal{C}^{R(\tilde{x})}} w_C - \gamma\tilde{u}^T a_{*C} + \lambda \sum_{i\in\mathcal{I}} \max\{0, \tilde{b}_i - a_{iC}\} \tag{4.4.6}$$

for given dual values $\tilde{u}$ derived from a fractional solution and with defining $\tilde{b} := b - \sum_{C\in\mathcal{C}} a_{*C}\tilde{x}_C$ for the current vector $\tilde{x}$. We can interpret the value $\tilde{b}$ as by how much is missing to satisfy Constraints (4.4.2) by the current solution $\tilde{x}$. Remember, that $\lambda > 0$ is a penalty parameter for Constraints (4.4.2) and parameter $\gamma = (2 - \frac{1}{\alpha}) \in (-\infty, 1]$ replaces parameter α in the convex combination of ϵ and δ. Zhao et al. (2018) suggest to exclude $\alpha \in (0, \frac{1}{2})$, which corresponds to $\gamma < 0$ as this is equivalent to changing the signs of the dual variables $\tilde{u}$. Setting $\gamma = 0$, which means $\alpha = \frac{1}{2}$, corresponds to minimizing over the original costs w_C and putting equal weights on ϵ and δ. The setting $\gamma = 1$, which means $\alpha = 1$, corresponds to minimizing over the reduced costs $w_C - \tilde{u}^T a_{*C}$. Therefore, parameter $\gamma \in [0, 1]$ determines the range of the objective function between the original and the reduced costs. Note that the value $\gamma = 0$ completely excludes the dual information from the fractional solution. Even though variables, which are useful in a fractional solution, are not necessarily useful for an integer solution, we expect a value $\gamma > 0$ to return better results since then the available information is included. In practice, similarly to the pricing problem in a column generation scheme, the auxiliary problem is not solved explicitly. In the next section we will see an implicit formulation of the minimization problem above applied to the considered partitioning problem.

4.4.2. Application to the Partitioning Problem

We have seen a general form of the auxiliary problem proposed by Zhao et al. (2018) in the previous section. The idea is to build an integer solution by exploiting dual information from a fractional solution. In this section, we apply problem (4.4.6) to find an integer solution to the partitioning problem formulated in Section 4.2. In our application, the feasible set is $X = \{0,1\}^{|\mathcal{C}|}$ since we use binary variables for all clusters with a size between k and $2k-1$. For a given vector $\tilde{x} \in X$, the set of incrementable indices is $\mathcal{C}^{R(\tilde{x})} = \{C \in \mathcal{C} \colon \tilde{x}_C = 0\}$, which are all indicator variables of clusters, which are not included in the current clustering. We denote by $\mathcal{I}'(\tilde{x}) \subset \mathcal{I}$ the set of unassigned data rows under vector $\tilde{x}$, which is

$$\mathcal{I}'(\tilde{x}) \coloneqq \{i \in \mathcal{I} \colon \tilde{x}_C = 0 \text{ for all } C \in \mathcal{C} \text{ with } i \in C\}.$$

In this application, the vector $\tilde{b} = b - \sum_{C \in \mathcal{C}} a_{*C}\tilde{x}_C$ indicates, which data rows $i \in \mathcal{I}$ are not in any cluster C with $\tilde{x}_C = 1$. The vector $\tilde{b}$ is defined by

$$\tilde{b}_i = \begin{cases} 1, & \text{if } i \in \mathcal{I}'(\tilde{x}), \\ 0, & \text{if } i \in \mathcal{I} \setminus \mathcal{I}'(\tilde{x}). \end{cases}$$

Therefore, we can rewrite the sum in the last term in Problem (4.4.6) by

$$\sum_{i \in \mathcal{I}} \max\{0, \tilde{b}_i - a_{iC}\} = \sum_{i \in \mathcal{I}'(\tilde{x})} (1 - a_{iC}), \tag{$\star$}$$

which is the number of unassigned vertices under $\tilde{x}$ that remain unassigned when including cluster C to the current solution. As mentioned before, we do not solve the auxiliary problem 4.4.6 explicitly. Instead, we find an index C to be incremented by adding constraints that enforce $C \in \mathcal{C}^{R(\tilde{x})}$ for a current vector $\tilde{x} \in X$. This results in an optimization problem very similar to the pricing problem in a column generation scheme. We introduce variables v_i for all data rows $i \in \mathcal{I}$ and edge variables z_e for all edges $e = (i,l) \in E$. These variables will indicate which vertices are contained in the cluster to be added to the current solution. As before in the pricing problem, we need constraints to ensure constructing a cluster in $\mathcal{C} = \{C \subset \mathcal{I} \colon k \le |C| \le 2k-1\}$. Using

($\star$), we can write the objective function in (4.4.6) as

$$\sum_{e\in E} w_e z_e - \gamma \sum_{i\in\mathcal{I}} \tilde{u}_i v_i + \lambda \sum_{i\in\mathcal{I}'(\tilde{x})} (1-v_i) = \sum_{e\in E} w_e z_e - \sum_{i\in\mathcal{I}} \gamma \tilde{u}_i v_i - \sum_{i\in\mathcal{I}'(\tilde{x})} \lambda v_i + \sum_{i\in\mathcal{I}'(\tilde{x})} \lambda.$$

Note that the constant term $\sum_{i\in\mathcal{I}'(\tilde{x})} \lambda$ can be neglected in the minimization problem. Since we want to ensure that each vertex in the data set $\mathcal{I}$ is not assigned to more than one cluster, we restrict the auxiliary problem to variables v_i for all yet unassigned vertices, i.e. vertices $i \in \mathcal{I}'(\tilde{x})$. The problem we use to find an integer clustering is

$$\begin{aligned}
\min_{v,z} \quad & \sum_{e\in E} w_e z_e - \sum_{i\in\mathcal{I}'(\tilde{x})} (\gamma u_i + \lambda)\cdot v_i && && (4.4.7)\\
\text{s.t.} \quad & z_{il} \geq v_i + v_l - 1 && \forall i,l \in \mathcal{I}'(\tilde{x}), && (4.4.8)\\
& k \leq \sum_{i\in\mathcal{I}'(\tilde{x})} v_i \leq 2k-1, && && (4.4.9)\\
& z_e \geq 0 && \forall e \in E, && (4.4.10)\\
& v_i \in \{0,1\} && \forall i \in \mathcal{I}'(\tilde{x}) && (4.4.11)
\end{aligned}$$

with $\gamma \in [0,1]$ and $\lambda > 0$ as parameters, and $\mathcal{I}'(\tilde{x}) \subset \mathcal{I}$ the set of unassigned vertices under $\tilde{x}$. Constraint (4.4.9) ensures that the cluster size is between k and $2k-1$. In order to obtain a clique, an edge must be included if and only if both incident vertices are included, which is ensured by Constraints (4.4.8). We do not need constraints for the opposite direction as all edge weights w_e are non-negative. Also, we do not need to enforce the variables z_e to be binary because the vertex variables v_i are binary and the weights w_e are non-negative.

We now describe the proposed method, which is also depicted in Algorithm 7. The method starts with an empty clustering, which corresponds to $\tilde{x}_C = 0$ for all $C \in \mathcal{C}$. We simplify the notation to $\mathcal{I}'(\tilde{x}) = \mathcal{I}'$. The set $\mathcal{I}'$ is initialized to be the set of all vertices $\mathcal{I}$ since there is no assigned vertex at first. Then, problem (4.4.7)–(4.4.11) returns a cluster and, thus, an index $C \in \mathcal{C}$ to be incremented. Therefore, the cluster C is added to the clustering $\mathcal{C}^*$, which will be returned. All vertices included in the found cluster are removed from set $\mathcal{I}'$. This procedure is repeated as long as there are at least k unassigned vertices left. This procedure ensures that a solution to the problem above always yields a new cluster. If there are less than k vertices unassigned, each remaining vertex is assigned to the best fitting existing cluster. Due to this last step, there could be clusters of size greater than $2k-1$. Therefore, all such clusters are divided into smaller clusters of size between k and $2k-1$. To this end, we simply divide the clusters by assigning the first k elements into one cluster and the remaining vertices into another cluster.

Next, we review the meaning of parameter λ in Problem (4.4.7)–(4.4.11) for the given application. The parameter was introduced as a penalty for violations of the constraints. In this application, the constraints ensure that each vertex is contained in exactly one cluster. However, the parameter can also be interpreted as follows. Due

```
Input: Graph $G = (\mathcal{I}, E)$ with edge weights $w_e$ for all $e \in E$ and dual
       variables $u_i$ for all $i \in \mathcal{I}$, parameters $\gamma \in [0, 1]$, $\lambda > 0$, and $k$ for
       $k$-anonymity.
Output: A partitioning $\mathcal{C}^*$ of $G$ into clusters of size between $k$ and $2k - 1$.
Initialization: Set $\mathcal{I}' = \mathcal{I}$, $\mathcal{C}^* = \emptyset$;
while $|\mathcal{I}'| \geq k$ do
    Solve problem (4.4.7)–(4.4.11) to optimality with optimal solution $(\tilde{v}, \tilde{z})$;
    Set $C = \{i \in \mathcal{I}' \colon \tilde{v}_i = 1\}$;
    Set $\mathcal{C}^* = \mathcal{C}^* \cup \{C\}$, $\mathcal{I}' = \mathcal{I}' \setminus C$;
end
for $i \in \mathcal{I}'$ do
    Find cluster $C' \in \mathcal{C}^*$ such that $w_{C' \cup \{i\}} \leq w_{C \cup \{i\}}$ for all $C \in \mathcal{C}^*$;
    Update $\mathcal{C}^*$: Set $\mathcal{C}^* = \mathcal{C}^* \setminus \{C'\} \cup \{C' \cup \{i\}\}$ ;
end
for $C \in \mathcal{C}^*$ do
    if $|C| \geq 2k$ then
        Separate the first $k$ elements of $C$ into a distinct cluster;
        Update $\mathcal{C}^*$ by replacing $C \in \mathcal{C}^*$ with the smaller clusters;
    end
end
return $\mathcal{C}^*$.
```

Algorithm 7: Greedy heuristic for finding an integer partitioning.

to its non-negativity, λ provides a reward for each variable that is set to 1, i.e. for each vertex that is included in the returned cluster. Clearly, by adding a vertex to a cluster, the dissimilarity in the cluster can not decrease. However, it often is useful to add a larger cluster to the solution with respect to the overall dissimilarity over all clusters, which can be seen from the following case. Assume, we have given a data set that includes $k+1$ data vectors that are highly similar to each other but not identical, and highly dissimilar to the remaining data records. Assume, these vectors would be assigned to the same cluster in an optimal clustering. The proposed method adds clusters one by one. With $\lambda = 0$, a cluster with k of the $k + 1$ similar data records would be preferred over a cluster of size $k + 1$ since the data vectors are not identical to each other and in the worst case the remaining data row is assigned to a cluster with more dissimilar records. If chosen correctly, the parameter $\lambda > 0$ can favor the case to assign all $k + 1$ data vectors to the same cluster. Note that choosing a value for parameter λ that is too high leads to more heterogeneous clusters as then, more emphasis is placed on the larger size than on the cluster weight.

4.5. Summary of the Proposed Partitioning Step

The basis of the method is the column-oriented formulation (4.2.1)–(4.2.3) and an approximate or optimal fractional solution to the linear relaxation. Next, given dual information provided by the fractional solution, an integer clustering is constructed iteratively. We describe the two steps further.

1. A fractional solution to the linear relaxation of the partitioning problem is found via column generation. In the pricing step two heuristics are applied: greedy heuristic and enumeration of closest records. We consider two options:

   Heuristic fractional solution: The column generation process is stopped if both heuristics do not yield a variable with negative reduced costs. Therefore, a feasible, not necessarily optimal, fractional solution is obtained.

   Optimal fractional solution: If both heuristics do not yield a variable with negative reduced costs, the pricing problem is solved to optimality. This procedure leads to an optimal fractional solution.

2. Given the dual information by a fractional solution, clusters are iteratively added to the returned clustering. In a greedy manner, the cluster to be added is built only from yet unassigned vertices. In each step, a cluster minimizing problem (4.4.7)–(4.4.11) is chosen. The problem is motivated from minimizing the duality gap. If less than k vertices remain, they are added to the best fitting cluster, respectively. In the end, cluster of size greater than $2k - 1$ are divided into smaller clusters and the resulting clustering is returned.

This chapter covered the first step of a microaggregation procedure, which is to find a suitable partitioning of the microdata set. In the next chapter, we discuss the subsequent aggregation step. During aggregation, a representative is selected for each cluster and all individual values are replaced by these representative values.

Chapter 5

Aggregation Step

In this chapter, we discuss the second phase of a microaggregation method, the aggregation step, in which representatives for each given cluster are identified. In the first part of a microaggregation method, the partitioning step, we determine a clustering $\mathcal{C}^*$ of a microdata set such that the clusters are of size between k and $2k-1$, where k is the parameter for k-anonymity. In the subsequent aggregation step, for each of the clusters a representative is selected. All individual records in a cluster are then replaced by the selected representative. The microdata set obtained by this procedure satisfies the k-anonymity property. In this chapter, we show a mathematical formulation of the problem to find suitable representatives and two heuristics to solve it.

We discuss several aspects of the proposed aggregation method in Section 5.1. In Section 5.2 we propose a mathematical formulation of the aggregation step. A graph-theoretical point of view to this problem is assumed in Section 5.3. Based on both formulations we present two heuristic approaches in Section 5.4. We conclude the chapter with a summary in Section 5.5.

5.1. Introduction

In this section, we motivate our approach to the aggregation methods. First, we include frequency tables in the aggregation step because (especially low-dimensional) frequency tables are of high interest for nominal data and consequently, the proposed method minimizes the information loss in (a subset of) the frequency tables. Furthermore, we motivate the use of a data-perturbative method, which can lead to compensation effects and therefore have a positive impact on errors in frequency tables. Then, we discuss the difficulties encountered in extending methods for numerical data to nominal data.

A special feature for nominal data compared to numerical data is the importance of frequency tables. Using a perturbative method on the level of the frequency tables, e.g. noise addition, can lead to inconsistencies between different tables. The anonymization on the level of frequency tables using non-perturbative methods like suppression is a well-studied field. However, anonymizing on the level of the microdata set, a non-perturbative method can lead to severe information loss in frequency tables. In contrast, using a perturbative method on the level of the microdata set can even preserve some of the original frequencies. The following example illustrates that in tables of lower dimensions frequencies can remain truthful even after the anonymization process. Note that in practice, especially frequency tables of a low dimension are of interest.

Example 5.1.1. Let the following original microdata set be given and consider a 2-anonymous microdata set corresponding to it.

Sex	*Citizenship*
female	Germany
female	Others
male	Germany
male	Others

(a) Exemplary microdata set.

Sex	*Citizenship*
female	Germany
female	Germany
male	Others
male	Others

(b) Exemplary 2-anonymous microdata set.

There are three frequency tables that can be generated from this microdata set. We consider the entries belonging to the original microdata set and, in parentheses, those belonging to the 2-anonymous microdata set depicted in Figure 5.2.

Sex	Counts
female	2 (2)
male	2 (2)

Citizenship	Counts
Germany	2 (2)
Others	2 (2)

	Citizenship	
Sex	Germany	Others
female	1 (2)	1 (0)
male	1 (0)	1 (2)

Figure 5.2.: Frequency tables corresponding to the exemplary (2-anonymous) microdata set.

In both one-dimensional frequency tables we see that even after anonymization the original cell entries could be preserved. Only in the two-dimensional table, errors occur due to the anonymization process.

Except for the trivial case, where the original microdata set already satisfies k-anonymity, the frequency table of the highest possible dimension will always include perturbed entries. This is due to the fact that this frequency table is just another representation of the microdata set since it includes all possible combinations of attribute values. In practice, frequency tables of lower dimensions are of higher interest.

In the aggregation step, typically the cluster representative is chosen to be close to the data records. For numerical data, this is achieved by using the mean values as representatives. The first extensions of microaggregation to categorical data also included the plurality mode, which is comparable to using the mean. In both cases, only the respective cluster is considered regardless of the cluster representatives of other clusters. However, taking the errors in resulting frequency tables into account, we see that selecting the representative of a cluster should not be independent from the representatives of the other clusters. The next minimal example shows this dependencies.

Example 5.1.2. Let $n = 9$ persons be asked about the attributes $\mathcal{J} = \{$*sex*, *working status*$\}$ with the value sets $V(\mathit{sex}) = \{\text{f, m}\}$ and $V(\mathit{working\ status}) = \{\text{empl, unempl, inact}\}$. Let the clustering of the microdata set be given by a set of three clusters

$$
\begin{aligned}
\mathcal{C}^* = \{&\{(\text{f, empl}), \quad (\text{f, empl}), \quad (\text{m, empl})\},\\
&\{(\text{f, unempl}), (\text{f, unempl}), (\text{m, unempl})\},\\
&\{(\text{f, inact}), \quad (\text{f, inact}), \quad (\text{m, inact})\}\}
\end{aligned}
$$

If the commonly used procedure for numerical data is adapted to nominal data by replacing the mean with the plurality mode in all clusters, the individual values for the attribute *sex* would be changed to *f*. We call this setting Scenario 1. However, we show that changing the data set to Scenario 2, as given below, would lead to less errors in the resulting frequency tables.

$$
\begin{aligned}
\{&\{(\text{f, empl}), \quad (\text{f, empl}), \quad (\text{f, empl})\},\\
&\{(\text{f, unempl}), (\text{f, unempl}), (\text{f, unempl})\},\\
&\{(\text{f, inact}), \quad (\text{f, inact}), \quad (\text{f, inact})\}\}
\end{aligned}
$$

(a) Scenario 1 using plurality mode on one cluster.

$$
\begin{aligned}
\{&\{(\text{f, empl}), \quad (\text{f, empl}), \quad (\text{f, empl})\},\\
&\{(\text{f, unempl}), (\text{f, unempl}), (\text{f, unempl})\},\\
&\{(\text{m, inact}), \quad (\text{m, inact}), \quad (\text{m, inact})\}\}
\end{aligned}
$$

(b) Scenario 2 including effects from other cluster representatives.

The entries of the one-dimensional frequency table displaying the attribute *sex* for the original microdata set and the two scenarios for 3-anonymity are depicted in Table 5.1. We see that the original cell entries can be preserved when the cluster representatives are chosen well. To identify suitable cluster representatives, we take the representatives of other clusters and the effects on the control tables into account. It is easy to verify that the entries of the frequency table displaying attribute *working status* remain unchanged in both scenarios. Note that errors occur in the two-dimensional tables in both cases because the original microdata set was not already k-anonymous.

	Counts		
Sex	original value	Scenario 1	Scenario 2
f	6	9	6
m	3	0	3

Table 5.1.: One-dimensional frequency table for the exemplary microdata set with original values, values for Scenario 1 and Scenario 2.

In a table cell, the representatives of all clusters jointly affect the resulting errors. Therefore, a cluster representative in the case of nominal data should also depend on the representatives of other clusters. To this end, we take the resulting errors in frequency tables into account. There are $\sum_{d=1}^{m} \binom{m}{d} = 2^m - 1$ frequency tables, so that the total number of frequency tables is exponential in the number of attributes. In practice, frequency tables of higher dimensions are not as important as tables of lower dimension. For this reason, the aggregation problem we analyze includes only the errors in a *subset* of all possible frequency tables.

Next to preserving the information from frequency tables, a second requirement for the cluster representative of nominal data is that the representative should have a reasonable combination of entries. By using the attribute-wise mean values for numerical data, the resulting cluster representative is a combination of attribute values that might not occur in the original microdata set in general. The same can be the case for nominal data when the plurality mode is used. For numerical data, this does not need to be a problem. However, for categorical data this can result in unreasonable combinations because logical connections or contradictions between the attributes are not taken into account when using the plurality mode. The following example illustrates this case.

Example 5.1.3. Let the data records of four persons be assigned to the same cluster that is given by the following table. Using the plurality mode, the cluster representative is given by (employed, None), which is an unreasonable combination.

Working status	*Job position*
employed	official
employed	employee
inactive	None
unemployed	None

To avoid generating unreasonable data records, one could prohibit the joint occurrence of attribute values that contradict each other. However, this approach requires a lot of a priori information about the logical connections between different entries. In contrast, our approach only allows data rows to be cluster representatives, which occur in the original microdata set. In consequence, unreasonable combinations can not be created in the anonymization process.

5.2. Formulation of the Aggregation Problem

We formulate an optimization problem that aims to find cluster representatives for each cluster. The program takes resulting errors in a subset of frequency tables into account and only allows data records occurring in the original microdata set to be selected as cluster representatives.

As described in Chapter 2, the SAFE-basic problem includes the *maximum* error in the frequency tables. However, we have discussed that focusing on the maximum error may lead to unnecessary errors in table cells. Moreover, it is reasonable to give more weight to an error in a table cell with a small original entry than to the same absolute error in a cell with a large original entry. This idea is also included in the SAFE-basic problem by the extra allowed error by parameter g. However, we decided to take the error between original values and cell values after the anonymization process relatively to the original value into account by using the χ^2-*error* in the objective function. The χ^2-error also penalizes larger errors more than smaller ones. Given the original value $\bar{n}_t > 0$ of table cell t and some value n_t, the χ^2-error is given by

$$\frac{(n_t - \bar{n}_t)^2}{\bar{n}_t}.$$

As the number of frequency tables is exponential, we can afford only to take a subset of tables in the aggregation step into account, the *control tables*. The number of control table cells directly increases the size of the problem and, therefore, the runtime required to solve the problem. In practice, tables of a low dimension would be included. We denote the set of all control table *cells* by $\mathcal{T}$. Furthermore, let $\bar{n}_t$ be the original value of table cell $t \in \mathcal{T}$.

All individual data records in a cluster are replaced by a cluster representative. To prevent the problem of creating unreasonable combinations, we only consider existing data records in each cluster as potential representatives. Remember that each data row is assigned to exactly one cluster. In the problem formulation, we will use binary variables z_i for all data rows $i \in \mathcal{I}$, which indicate whether data row $i \in C$ is selected as cluster representative of cluster C. Note that it is sufficient to only consider distinct data records in each cluster, which might decrease the number of variables.

After replacing all individual data records with representative values, all data records are counted in the same table cells after anonymization. Thus, the contribution of a cluster to a table cell is fully defined by the size of the cluster and the representative. We use an indicator to show whether a data row is counted in a table cell, which contains the cluster size. The indicator with multiplicity is defined as

$$f_{ti} := \begin{cases} |C|, & \text{if } i \in C \text{ is counted in table cell } t, \\ 0, & \text{else} \end{cases}$$

for all table cells $t \in \mathcal{T}$ and data rows $i \in C$ for all $C \in \mathcal{C}^*$. It is used to compute the value of table cell t after the anonymization process.

We formulate the aggregation problem as follows:

$$\min_{z} \sum_{\substack{t \in \mathcal{T}: \\ \bar{n}_t > 0}} \frac{(f_{t*}^T z - \bar{n}_t)^2}{\bar{n}_t} \tag{5.2.1}$$

$$\text{s.t.} \sum_{i \in C} z_i = 1 \qquad \forall C \in \mathcal{C}^*, \tag{5.2.2}$$

$$z_i \in \{0, 1\} \qquad \forall i \in \mathcal{I}. \tag{5.2.3}$$

Since only data records of the original microdata set are potential representatives, table cells with an original entry of zero will remain unchanged, i.e. the error in these cells is zero. Hence, it is sufficient to only consider table cells with an original value $\bar{n}_t$ greater than zero. After the anonymization process, each cluster is either entirely or not counted at all in a table cell. In other words, a cluster either contributes to a table cell entry with the size of the cluster or it does not contribute to the cell. A cluster is counted in a table cell if and only if the representative was originally counted in the cell. Thus, the new entry of a table cell $t \in \mathcal{T}$ can be computed by using the variables z_i and the indicators f_{ti}. It is given by $\sum_{i=1}^{n} f_{ti} z_i = f_{t*}^T z$ and it is used in the objective function. The objective function in (5.2.1) is the sum of χ^2-errors between all non-zero control table cells before and after anonymization. Constraints (5.2.2) ensure that for each cluster exactly one of its elements is selected as representative. Constraints (5.2.3) force the variables z_i to be binary.

It is sufficient to only take distinct data records in each cluster into consideration. Therefore, in a preprocessing step, all duplicates of data records in the same cluster can be eliminated to reduce the number of variables in the program. In clusters with only identical data rows, the representative is fixed by setting the variable z_i, corresponding to an arbitrary element of the cluster, to 1. In consequence, a partitioning of the microdata set into homogeneous clusters leads to a smaller number of variables in the aggregation step and, thus, to faster runtimes.

5.3. Reformulation to a Maximal Clique Problem in a Multipartite Graph

We have seen a problem formulation for finding cluster representatives, which aims at minimizing the sum of χ^2-errors between control table cells before and after anonymization. In this section, we present a reformulation of problem (5.2.1)–(5.2.3). The integer program can be interpreted as the task of finding a minimum-weighted maximal clique in a graph. This representation allows a different approach to the problem and we will use it to propose a graph-based heuristic to the aggregation problem. The basic idea is to interpret each data row as a vertex in an appropriately structured graph. By including the objective function as edge weights, the aggregation problem is equivalent to finding a minimum edge-weighted maximal clique in a multipartite graph. This problem is also described by Feremans et al. (2003) as generalized minimum clique problem and Koster et al. (1998) show that it is $\mathcal{NP}$-hard. However, the graph-based reformulation allows for another viewing angle to the problem.

For the problem reformulation, we define graph $G_{\text{aggr}} = (\mathcal{I}_{\text{aggr}}, E_{\text{aggr}})$ as follows. Let the vertex set $\mathcal{I}_{\text{aggr}} \subset \mathcal{I}$ correspond to a subset of the data rows, where all duplicates of data rows assigned to the same cluster are eliminated. To find a set of representatives according to the aggregation problem (5.2.1)–(5.2.3), we have to ensure that each cluster is represented by exactly one vertex. To this end, we define the edge set E_{aggr} as all edges between vertices that are assigned to *different* clusters. Edges between vertices in the same cluster are not contained in E_{aggr}. By this definition, each clique can contain at most one vertex from each cluster. To summarize, the considered graph consists of $|\mathcal{C}^*|$ partitions corresponding to the given clusters where there is no edge between any two vertices in the same partition and one edge between every pair of vertices from different partitions. Thus, we use a complete multipartite graph where each clique corresponds to a cluster. The following example illustrates the definition of the graph G_{aggr}.

Example 5.3.1. Let $\mathcal{I} = \{1, \dots, 7\}$ be the index set for a microdata set that contains 7 data rows. Let a clustering of the microdata set be given by $\mathcal{C}^* = \{\{1, 2\}, \{3, 4, 5\}, \{6, 7\}\}$, which is suitable for 2-anonymity. Assume that the data rows in the clusters are distinct from each other. Figure 5.4 illustrates the corresponding graph G_{aggr}.

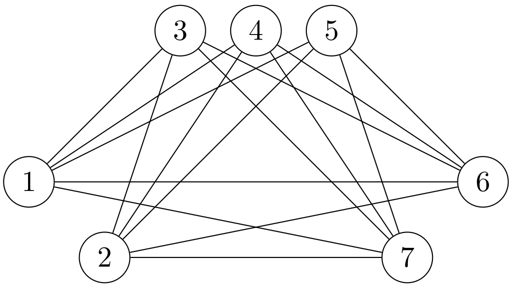


Figure 5.4.: Exemplary graph G_{aggr}.

We assign weights to the edges and to the vertices such that finding representatives, which are optimal in the sense of problem (5.2.1)–(5.2.3), is equivalent to finding a maximal clique with minimum weight in graph G_{aggr}. As a first requirement, we obtain that the objective function can be split into a quadratic part, which will be used in the edge weights, and a linear term, which can be used as vertex weights. Next, we see that the vertex weights can also be represented as edge weights, such that it is sufficient to only consider edge weights. The objective function (5.2.1) of the aggregation problem can be reformulated as follows.

$$\begin{aligned}\sum_{t\in\mathcal{T}} \frac{(f_{t*}^T z - \bar{n}_t)^2}{\bar{n}_t} &= \sum_{t\in\mathcal{T}} \frac{(f_{t*}^T z)^T (f_{t*}^T z)}{\bar{n}_t} - 2\sum_{t\in\mathcal{T}} \frac{(f_{t*}^T z)\bar{n}_t}{\bar{n}_t} + \sum_{t\in\mathcal{T}} \frac{\bar{n}_t^2}{\bar{n}_t} \\ &= \sum_{t\in\mathcal{T}} \frac{1}{\bar{n}_t} (f_{t*}^T z)^T (f_{t*}^T z) - 2\sum_{t\in\mathcal{T}} f_{t*}^T z + \sum_{t\in\mathcal{T}} \bar{n}_t \\ &= \sum_{t\in\mathcal{T}} \frac{1}{\bar{n}_t} z^T \left(f_{t*} f_{t*}^T\right) z - 2\sum_{t\in\mathcal{T}} f_{t*}^T z + \sum_{t\in\mathcal{T}} \bar{n}_t \end{aligned} \qquad (5.3.1)$$

$f_{t*}f_{t*}^T$ is a symmetric matrix.

From the term $z^T f_{t*} f_{t*}^T z$, we derive edge weights

$$\tilde{w}_{il} = 2\sum_{t\in\mathcal{T}} \frac{1}{\bar{n}_t} f_{ti} f_{tl}$$

for all edges $(i,l) \in E_{\text{aggr}}$. From the diagonal entries in the matrix $f_{t*}f_{t*}^T$ and the linear term in Equation (5.3.1), we obtain vertex weights. For all vertices $i \in \mathcal{I}_{\text{aggr}}$, we derive vertex weights

$$\tilde{w}_i = \sum_{t\in\mathcal{T}} \frac{1}{\bar{n}_t} f_{ti}^2 - 2\sum_{t\in\mathcal{T}} f_{ti}.$$

We first omit the constant term $\sum_{t\in\mathcal{T}} \bar{n}_t$ and then split the objective function into weights on the edges and the vertices of the graph G_{aggr} as given above. Due to the structure of the graph G_{aggr}, i.e. a multipartite complete graph, a maximal clique in G_{aggr} consists of exactly one vertex per component. Using the edge and vertex weights from above, minimizing the χ^2-error in the frequency tables is equivalent to finding a minimum weight maximal clique in this graph. The weight of a clique is the sum of all weights of contained edges and vertices.

The graph G_{aggr} has the property that every pair of two vertices belonging to distinct clusters is adjacent. Therefore, the size of a maximal clique is known to be $|\mathcal{C}^*|$. Moreover, it is known that for each vertex in a clique of $|\mathcal{C}^*|$ vertices, exactly $|\mathcal{C}^*| - 1$ incident edges are contained in a maximal clique. Therefore, we can divide the vertex weights among the edges. With the same explanation, we can also divide the constant term in Equation (5.3.1) among the edges. Here, we exploit the well-known fact, that a clique with $|\mathcal{C}^*|$ vertices includes $\frac{1}{2}|\mathcal{C}^*|\,(|\mathcal{C}^*| - 1)$ edges. Thus, we end up with a complete multipartite graph with edge weights only. The edge weights equal

$$w_{il} = \tilde{w}_{il} + \frac{1}{|\mathcal{C}^*| - 1}\left(\tilde{w}_i + \tilde{w}_l\right) + \frac{2}{|\mathcal{C}^*|(|\mathcal{C}^*| - 1)} \sum_{t\in\mathcal{T}} \bar{n}_t$$

for all edges $(i, l) \in E_{\text{aggr}}$.

This graphical representation allows another viewing angle.

5.4. Heuristic Approaches

In this section, we propose two heuristic approaches to the aggregation step for nominal data, which we formulated in the last chapter. The first method is a greedy algorithm based on the integer program, which is presented in Section 5.2. We describe this heuristic in Section 5.4.1. Moreover, we propose a second heuristic based on the reformulation, which is described in Section 5.3. This approach is described in Section 5.4.2.

We precede both heuristics with a preprocessing step. First, all duplicates of data rows in the same cluster are eliminated to reduce the number of variables . In a second preprocessing step, we further prescribe variables as follows. If all data rows in the same cluster have the same value for an attribute, this value can be predefined for the representative in advance. Hence, the contribution of this cluster to the table cells, counting the fixed attributes, can be fixed before starting the aggregation procedure. We show an illustrative example below.

Example 5.4.1. Let a microdata set be partitioned into $\mathcal{C}^* = C_1 \cup C_2 \cup C_3$ with

$$\begin{aligned} C_1 &= \{(\text{f, GER}), (\text{f, GER}), \ \ (\text{f, GER})\} \\ C_2 &= \{(\text{f, OTH}), (\text{f, OTH}), \ \ (\text{m, OTH})\} \\ C_3 &= \{(\text{f, GER}), (\text{m, GER}), (\text{m, OTH})\} \end{aligned}$$

and let the control tables be all one-dimensional tables. Note that duplicated elements belong to distinct statistical objects and unique identifiers are required for technical correctness, but we omit them for better readability. After preprocessing, the data set is reduced to

$$\begin{aligned} C_1 &= \{(\text{f, GER})\} \\ C_2 &= \{(\text{f, OTH}), (\text{m, OTH})\} \\ C_3 &= \{(\text{f, GER}), (\text{m, GER}), (\text{m, OTH})\} \end{aligned}$$

The cluster representative of C_1 is set to (f, GER), since all cluster elements are identical to each other. Even though the representative of cluster C_2 cannot yet be determined, the contribution to the one-dimensional frequency table showing attribute *citizenship* is already predefined. In Cluster C_3, no values are fixed in the preprocessing step. In conclusion, the initial table cell entries are set as depicted in Figure 5.5.

Sex	Counts
f	3 (6)
m	0 (3)

Citizenship	Counts
GER	3 (5)
OTH	3 (4)

Figure 5.5.: Initial (original) control table cells after preprocessing.

5.4.1. Greedy Algorithm

We propose a greedy algorithm to approach the aggregation problem. The method iterates over all clusters and for each cluster, a data row that contributes the least to the objective function is selected as cluster representative. The procedure is depicted in Algorithm 8. The initial new cell entries n_t are given after preprocessing as described above for all $t \in \mathcal{T}$. In each iteration, a representative for one cluster is fixed by choosing the data row, which causes the lowest increase with respect to the objective function. Note that the contribution of individual cluster representatives to the objective function depends on the previously selected representatives. The new values n_t are updated in each iteration according to the selected representative. The procedure terminates when it has iterated over all clusters and, thus, one representative for each cluster is fixed.

```
Input: clustering $\mathcal{C}^*$ of a microdata set, control table cells $\mathcal{T}$, initial new cell
       entries $n_t$ for all $t \in \mathcal{T}$.
Output: A $k$-anonymous microdata set.
for $C \in \mathcal{C}^*$ do
    Find $i \in C$ with minimum value for $\sum_{\substack{t \in \mathcal{T}: \\ \bar{n}_t > 0}} \frac{(f_{ti}+n_t)-\bar{n}_t}{\bar{n}_t}$;
    Update $n_t = f_{ti} + n_t$ for all $t \in \mathcal{T}$;
    Replace all individual data rows in $C$ by data row $i$;
end
return the modified microdata set.
```

Algorithm 8: Greedy heuristic for finding cluster representatives.

5.4.2. Graph-based Heuristic

In Section 5.3, we reformulated the problem of finding cluster representatives that minimize the overall χ^2-errors in the control table cells to a problem of finding a maximal clique with minimum edge-weights in a multipartite graph. The partitions of the graph correspond one-to-one to the given clusters of the microdata set. In this section, we present a heuristic approach to address this problem that can be considered a top-down approach, in which the full graph is reduced iteratively until there is only one vertex left for each partition, i.e. for each cluster. The remaining vertex is selected as cluster representative. For each vertex in a cluster, a lower bound for the weight of a maximal clique with this vertex is computed. The vertex with the largest calculated lower bound is deleted.

Since the graph is multipartite, a clique can contain at most one vertex per partition. The graph is complete, i.e. every pair of vertices from two distinct partitions is adjacent. Therefore, a maximal clique contains exactly one vertex per partition. The proposed heuristic approach is depicted in Algorithm 9 and works as follows. The cluster representatives will be given in a set R. This set is initialized to be the full graph partitioning $\mathcal{C}^*$. In each iteration, the algorithm iterates over the clusters $C \in R$ and deletes a vertex in each cluster that has more than one element. The procedure terminates when there is only one vertex left in each cluster. For each vertex $i \in C$, a lower bound of the weight of a maximal clique that contains this vertex is computed. As a maximal clique contains exactly one vertex from each cluster, if vertex i is contained in the clique, then an edge from i to exactly one vertex in each other cluster must be contained as well. A lower bound on the weight of an edge between i and a vertex in a different cluster $C' \neq C$ is given by $\min_{l \in C'}\{w_e \colon i, l \in e\}$. Since this holds for all clusters distinct from C, a lower bound for the weight of a maximal clique containing vertex i can be computed by

$$\sum_{C' \in \mathcal{C}^* \setminus C} \min\{w_e \colon i, l \in e, l \in C'\}.$$

When there is only one vertex left in a cluster, this vertex is set to be the cluster representative. Thus, this cluster can be excluded from further iterations. The algorithm iterates over all clusters and repeats this procedure until one cluster representative is found for each cluster. In the end, the set R consists of singleton sets that contain the computed cluster representatives.

```
Input: Multipartite Graph $G_{\text{aggr}} = (\mathcal{I}_{\text{aggr}}, E_{\text{aggr}})$ with edge weights $w_e$ for all
       $e \in E_{\text{aggr}}$ and clustering $\mathcal{C}^*$ of $\mathcal{I}_{\text{aggr}}$.
Output: A maximal clique in $G_{\text{aggr}}$.
Initialization: Set $R = \mathcal{C}^*$;
while there is $C \in R$ with $|C| > 1$ do
    for $C \in R$ with $|C| > 1$ do
        Find vertex $i \in C$ maximizing the term
          $\sum_{C' \in R \setminus C} \min\{w_e \colon i, l \in e, l \in C'\}$;
        Update $R$ by replacing $C$ with $C \setminus \{i\}$;
    end
end
return clique corresponding to $R$.
```

Algorithm 9: Graph-based heuristic for finding cluster representatives.

5.5. Summary of the Proposed Aggregation Step

In the partitioning step of a microaggregation procedure, the microdata set to be anonymized is partitioned into clusters of size between k and $2k-1$. The topic of this chapter is the second phase of the microaggregation method, the aggregation step. In this step, the individual records in each cluster are replaced by a representative to obtain k-anonymity. Due to the importance of frequency tables for nominal data, the proposed method takes errors in a subset of frequency tables into account. We have seen a mathematical formulation for the aggregation problem. Equivalently, the problem can be formulated as a minimum-weight maximal clique problem in a multipartite graph. Based on these different formulations, we have proposed two heuristic approaches, a greedy algorithm and a graph-based heuristic.

The proposed aggregation method selects one data row in each cluster to be the cluster representative. The resulting χ^2-errors in a set of control frequency tables is decisive for the selection of the representative. Note that all cluster representatives jointly contribute to the errors in the frequency tables. Therefore, in the proposed methods, the representatives of other clusters are included in the selection process of the representative of a cluster. Since the proposed methods work on the microdata level, all frequency tables after anonymization are generated from the derived k-anonymous microdata set and fulfill the following properties. First, only attribute combinations, which occur in the original microdata set, are contained in the anonymized microdata set. In consequence, structural zeros are preserved, i.e. table cell entries of zero remain zero after the anonymization process. However, entries that are zero after the anonymization process need not have been zero originally. Second, since the frequency tables are generated from a k-anonymous microdata set, they satisfy k-anonymity, i.e. each cell entry is either 0 or at least k. This holds for any table, regardless of whether it was used as a control table in the procedure. Additionally, all frequency tables are consistent to each other. This means that the entries of one table do not contradict any entry of another frequency table and, thus, no additional information can be revealed by comparing the frequency tables.

5.6. Exact Method Combining Both Steps

In the previous chapters 4 and 5, we focused on the idea of dividing a microaggregation procedure in two steps, the partitioning and the aggregation step. This idea is widely used especially for numerical data. Naturally, the question arises how the proposed method compares to a method that combines both steps. In this section, we present a mathematical formulation of the microaggregation problem as one combined optimization problem, which aims to minimize the χ^2-error in the control tables.

The SAFE-basic problem, described in Chapter 2, combines partitioning and aggregation steps. However, we have pointed out that it is beneficial to modify the program. Instead of including only the maximum error, we proposed to include the resulting deviations between original and anonymized frequency vector. Moreover, we have discussed the advantages of using the χ^2-error. This measurement does not only penalize larger errors more than smaller ones but it takes the original cell value into account and penalizes errors more if there are originally only very few counts in the cell. Building on these aspects, we now formulate an optimization problem that combines both steps and aims at minimizing the χ^2-error, which we also used as objective function in the aggregation step.

Similarly to the modified optimization problem as described in Section 3.2, we introduce the following formulation to minimize the sum of χ^2-errors.

$$
\begin{aligned}
\min_{y,b} \quad & \sum_{t \in \mathcal{T}} \frac{b_t^2}{(Ay^0)_t} + w \sum_{i=1}^{n} |y_i - y_i^0| && (5.6.1)\\
\text{s.t.} \quad & Ay + b = Ay^0, && (5.6.2)\\
& \sum_{i=1}^{n_a} y_i = \sum_{i=1}^{n_a} y_i^0, && (5.6.3)\\
& y_i \in \mathbb{N}_0 \setminus \{1, \dots, k-1\} \qquad \forall i = 1, \dots, n, && (5.6.4)\\
& b_t \in \mathbb{R} \qquad \forall t \in \mathcal{T}. && (5.6.5)
\end{aligned}
$$

Again, the microdata set is written in the form of a frequency vector $y^0 \in \mathbb{N}_0^{n_a}$, where n_a is the number of distinct occurring combinations of attribute values, A is a binary matrix such that Ay is the vector of control cell entries resulting from vector y, and $w > 0$ is a weighting parameter. The original frequency vector is denoted by y^0 and the constraints are similar to (2.1.2)–(2.1.5). Constraint (5.6.2) defines variable b to be the error between original and newly computed table cell entries. It is ensured by Constraint (5.6.3) that the total number of data rows is preserved. Constraints (5.6.4) enforce that the entries of the new frequency vector are not between 1 and $k-1$. Thus, they ensure k-anonymity for a given parameter k. Due to this equality constraint and the integrality of y that is enforced by Constraints (5.6.4), variables b are automatically integral.

The optimization problem is an integer program, which will be challenging to solve on large instances. However, we will use the presented exact method for comparison on smaller instances and call the method of exact solving this problem Chi-ex.

Chapter 6

Computational Experiments

The task to determine a k-anonymous microdata set to anonymize a given original microdata can be approached in different ways. We have discussed the SAFE method in Chapters 2 and 3, which was used in the last census in Germany in 2011. This algorithm is controlled by many parameters, which have to be determined. In contrast to this, the proposed algorithm presented in Chapters 4 and 5 requires only two parameters. In this chapter, we analyze the computational performance of the proposed method. We study the parameter selection for it and evaluate the single steps. We compare the proposed method to the SAFE heuristic and to exact methods, i.e. the SAFE-basic problem, the modified variant of the SAFE-basic problem and the program, which is presented in Section 5.6 and aims to minimize the χ^2-error.

In Section 6.1, we describe criteria, which we use to analyze the quality of solutions in terms of the errors in resulting frequency tables. We show computational experiments and results in Section 6.2. This section includes the description of the experiment setup, technical details of the single steps of the proposed method, and the comparison to other methods, i.e. SAFE and exact methods. To conclude, we summarize the chapter in Section 6.3.

6.1. Analysis Criteria

For categorical data, errors in frequency tables are of high interest to analyze the quality of a solution. However, there are several possibilities to interpret the resulting errors in the tables. In this section, we introduce different criteria, which we then use to analyze performance. As reported by Höhne (2015), in praxis, especially frequency tables of lower dimensions are important for many data users. Therefore, in addition to overall measurements, we also analyze results per dimension.

In the proposed method, we work with a subset of the table cells, the control table cells. However, we also evaluate the effect of the anonymization process on tables, which have not been selected as control tables. We use the notation $\mathfrak{T}$ to denote the set of analyzed table cells. We will focus on the analysis of the set of control table cells and the set of *all* table cells. Furthermore, let $\bar{n}_t$ be the original value for table cell $t \in \mathfrak{T}$ and n_t be the value after the anonymization process. For some criteria, it is useful to distinguish the tables according to their dimension. Hence, we denote by $\mathfrak{T}_d \subseteq \mathfrak{T}$ the subset of table cells that belong to d-dimensional tables.

Sum of χ^2-errors: The total sum of χ^2-errors between table cells before and after anonymization is defined by

$$\sum_{t \in \mathfrak{T}} \frac{(n_t - \bar{n}_t)^2}{\bar{n}_t}$$

and can be evaluated for each dimension d by replacing $\mathfrak{T}$ with $\mathfrak{T}_d$. The χ^2-error is used in the objective function of the proposed aggregation step, therefore, on the control table cells, the results should be particularly good.

Maximum absolute error of dimension d: The maximum absolute error for tables of dimension d is

$$\max_{t \in \mathfrak{T}_d} |n_t - \bar{n}_t|.$$

The maximum absolute error is used in the objective function of the SAFE-basic problem.

Maximum relative error of dimension d: The maximum relative error for tables of dimension d is a maximum error weighted by the original value and is defined by

$$\max_{t \in \mathfrak{T}_d} \frac{|n_t - \bar{n}_t|}{\bar{n}_t}.$$

We will compare methods $m \in \mathcal{M}$ on one instance by a *gap*, which we define as follows. For a method $m \in \mathcal{M}$, let w_m denote the objective value achieved by m. Let $w^* = \min_{m \in M} w_m$ be the minimum objective value among all methods in $\mathcal{M}$. We assume $w^* > 0$ and define the gap for method m by

$$gap_m = \frac{w_m - w^*}{w^*}. \tag{6.1.1}$$

By the assumption of $w^* > 0$, this gap is non-negative. The smaller the gap, the closer the objective value w_m returned by method m is to the best found solution w^*.

6.2. Computational Experiments

In this section, we study the computational behavior of the proposed methods. First, we analyze the effects of the parameter settings. We have presented several steps in the proposed method, where we suggested to accelerate the algorithm by using heuristics. We compare the performances of using heuristics and solving to optimality in the individual steps. Moreover, we compare the proposed method to other methods, the SAFE heuristic and exact methods, which we have presented in previous chapters. We start by describing the used data sets.

6.2.1. Data Collection

Some of the data sets used in our computer experiments are randomly drawn samples from a U.S. census database and others are synthetically generated data. We create the synthetic data by prescribing the proportion of confidentiality problems. This approach allows us to compare the performances of the methods depending on the number of confidentiality problems in the original microdata set. For each scenario, we generate and evaluate 10 instances. In the following, we go into more detail about the used data sets.

Random sampling from data set "Adult" The "Adult" data set is provided by UCI Machine Learning Repository by Dua and Graff (2019), which is a database extracted from the 1994 U.S. census. It consists of 14 attributes, 8 of which are categorical. First, we sanitize the database by removing all data rows with missing entries in any of the categorical variables. The data set includes 30162 data rows after this adjustment.

Each attribute has a different set of values. We show the number of values per attribute in Table 6.1. We have sorted the attributes in ascending order by the number of values. The attributes are *sex* (1), *race* (2), *relationship* (3), *marital-status* (4), *working class* (5), *occupation* (6), *education* (7) and *native-country* (8).

attribute j	1	2	3	4	5	6	7	8
number of values $\lvert V(j)\rvert$	2	5	6	7	8	14	16	42

Table 6.1.: Numbers of values per attribute in the "Adult" data set.

We draw 10 samples for each size. In the evaluations, we mainly focus on a size of $n = 200$ data rows and the first five attributes. The size of the frequency vector depends on the number of values per attribute, which occur in the data set. Table 6.2 shows the number of values per attribute, which actually occur in the 10 drawn samples of size $n = 200$ on the first five attributes.

attribute j	1	2	3	4	5
range of $\lvert V(j)\rvert$	$[2]$	$[4, 5]$	$[6]$	$[6, 7]$	$[6]$

Table 6.2.: Range of number of values, which actually occur in the 10 samples drawn with $n = 200$.

We denote the scenario of 200 random data rows drawn from the "Adult" data set containing attributes $1, \ldots, 5$ by *adult200m5*.

To gain insights into the data, we investigate the number of confidentiality problems since this is crucial for anonymization. The higher the number of confidentiality problems, the more data rows need to be perturbed to achieve k-anonymity. Naturally, the higher the number of considered attributes, the higher the number of confidentiality problems since then, more value combinations are possible. Table 6.3 shows the proportion of confidentiality problems for a k-anonymity parameter of $k = 3$. In this case of $k = 3$, confidentiality problems are data rows, which are either unique or identical to exactly one other data row. The table shows the proportion in the adjusted original data set, which consists of 30162 data rows, in the samples of $n = 200$ data rows and $n = 500$ data rows in dependency of the number of attributes. Note that attributes are added according to their order in the sorted attribute list as above. Due to the smaller number of data rows, the proportion of confidentiality problems in drawn smaller samples is higher compared to the full microdata set.

	Number of attributes					
	3	4	5	6	7	8
Drawn samples $n = 200$ (avg.)	0.075	0.164	0.293	0.664	0.897	0.902
Drawn samples $n = 500$ (avg.)	0.035	0.089	0.196	0.473	0.764	0.782
Adjusted "Adult" data set	0.000*	0.002	0.013	0.059	0.185	0.239

* 2 out of 30162 data rows are confidentiality problems.

Table 6.3.: (Averaged) proportion of confidentiality problems for $k = 3$ depending on the number of attributes.

Synthetic data with a predefined number of confidentiality problems The proportion of confidentiality problems in smaller samples significantly differs from the proportion in the underlying database. To control the proportion of confidentiality problems, we investigate synthetic data, for which we prescribe the proportion of confidentiality problems in advance. In addition, this allows us to specifically investigate the extreme cases with a very few or very high number of problems. The extreme case of zero confidentiality problems can be neglected since this original microdata set already fulfills k-anonymity.

Table 6.4 reports the number of values per attribute in the synthetic data.

attribute j	1	2	3	4	5
number of values $\lvert V(j) \rvert$	2	5	6	7	7

Table 6.4.: Numbers of values per attribute in the synthetic data set.

The synthetic data is generated as follows. Let $p \in [0, 1]$ be the proportion of confidentiality problems, which should be included in the data set. We first construct the microdata by generating the rows, which are not part of the confidentiality problems. We then randomly generate a data row out of the possible values, which has not been generated before. We draw a random number n_r between $k - 1$ and $0.15 \cdot n$ and duplicate this data row n_r times. The lower bound on the random number ensures that the generated data row is no confidentiality problem. We introduced the upper bound on the random number to ensure a reasonable distribution of different data rows in the data set. We repeat this procedure until $\lfloor (1 - p) \cdot n \rfloor$ data rows are generated. Then, we add $\lceil p \cdot n \rceil$ confidentiality problems by randomly generating data rows, which do not yet exist. We then either do not copy this entry or duplicate it maximal $k - 2$ times.

We focus on the following scenarios for $n = 200$ data rows: *200m5_2percent* ($p = 0.02$), *200m5_20percent* ($p = 0.2$), and *200m5_100percent* ($p = 1$). For each scenario, we create 10 instances.

6.2.2. Hard- and Software

All experiments are performed on a machine with an Intel Core i3-8100 and a random access memory capacity of 16 GB. We implemented the methods in python 3. For the column generation process, we used the non-commercial solver SCIP version 7.0.2 provided by Gamrath et al. (2020). We use Gurobi Optimizer version 9.1.2 for the exact solving of the optimization programs including the pricing step in the column generation scheme.

6.2.3. Parameter Settings in the Partitioning Step

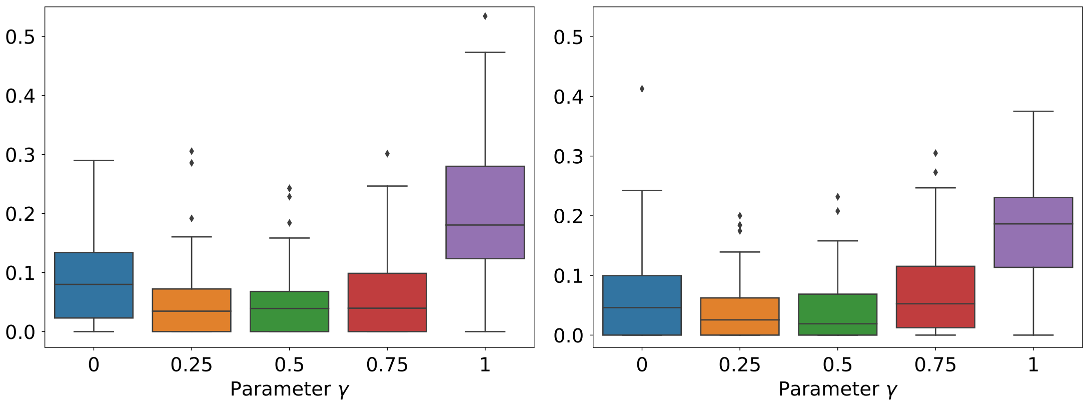
In this section, we report technical insights concerning the parameters of the proposed method. We use the gap, which is defined in (6.1.1), to compare different solutions to the partitioning problem. The set of methods $\mathcal{M}$ includes only the methods, which are shown in the respective figures. In the definition of the gap, the best objective value w^* is assumed to be positive. We can make this assumption without loss of generality because the objective value is the weight of the clustering, which is the sum of all cluster weights. Since every cluster weight is non-negative by definition, $w^* \geq 0$ holds. Moreover, an optimal objective value of $w^* = 0$ means that each cluster in an optimal clustering only contains identical data rows. This can only be the case if the original microdata set already satisfies k-anonymity. Hence, we can assume $w^* > 0$ without loss of generality.

The proposed partitioning method consists of two steps. First, the method obtains a fractional solution to the partitioning problem by using column generation. With the derived dual information, an integer clustering is constructed. We analyze the experiments separately for using an optimal or heuristically found fractional solution. The optimal fractional solution is obtained by running the column generation process until there are no more variables with negative reduced costs, i.e. the pricing problem is solved exactly and returns a non-negative optimal value. The heuristically found fractional solution is obtained by stopping the column generation when the introduced heuristics in the pricing step do not return any variable with negative reduced costs.

The proposed partitioning method uses two parameters. These parameters occur in the optimization problem for finding an integer solution. Parameter $\gamma \in [0, 1]$ weights the dual information, which is provided by the fractional solution. Parameter $\lambda > 0$ rewards the inclusion of more data rows in the same cluster and, thus, influences cluster sizes. We evaluate the objective value in the partitioning step, which is the weight of the integer clustering for different parameter settings. We compare experiments on each combination of $\lambda \in \{0.001, 0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4\}$ and $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$.

We focus on Scenario *200m5_20percent*. First, we study the parameters separately. Figure 6.1 shows the gaps according to the weight of the integer clusterings for different values of γ, given an optimal fractional solution (6.1a) and a heuristic fractional solution (6.1b), respectively. The boxes show the median and the quartiles of the data set. The whiskers extend to data points within 1.5 times the interquartile range, which is the distance between the upper quartile and the lower quartile. Data points outside this boundary are displayed as outliers. The figure includes all executions with $\lambda \in \{0.001, 0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4\}$ on 10 instances so that each boxplot contains 90 instances. The figure illustrates that the extremal value of $\gamma = 1$ performs worse than the other values. The parameter γ acts as a weighting on the dual information. The higher the value, the more weight is given to the dual variables. It is known that the dual information corresponding to an optimal *fractional* solution does not necessarily relate at all to an optimal *integer* solution. Therefore, it is reasonable that the weight γ on the dual variables should not be chosen too high as higher weights γ favor fractional solutions. Our calculations indicate that for a heuristic fractional solution also a value of $\gamma = 0.75$ is not advisable. We see a tendency that for a heuristically found fractional solution, a lower value of $\gamma \in \{0.25, 0.5\}$ is preferable. It is reasonable that the information provided by an optimal solution is more valuable to be taken into account with a larger weight γ than the information provided by a heuristic solution. Moreover, the extremal value of $\gamma = 0$ also performs worse than the interior values of $\gamma \in \{0.25, 0.5\}$ (and $\gamma = 0.75$ for an optimal fractional solution). The value $\gamma = 0$ corresponds to excluding the dual information provided by the fractional solution. Thus, the inclusion of dual information with a low or medium weight improves the search for an integer solution. The same qualitative behavior can be seen on Scenario *adult200m5*, which is depicted in the appendix in Figure B.1. For this scenario, $\gamma = 1$ does not lead to best results for any instance. Moreover, the tendency to prefer lower γ when using a heuristically found fractional solution, is even more pronounced.

We study the choice of parameter γ closer. Figure 6.2 depicts the direct comparison of the interior values $\gamma \in \{0.25, 0.5, 0.75\}$ using an optimal fractional solution. The lines connect the values, which correspond to the same instance. The subfigures include the same instances as Figure 6.1. However, the instances are divided among

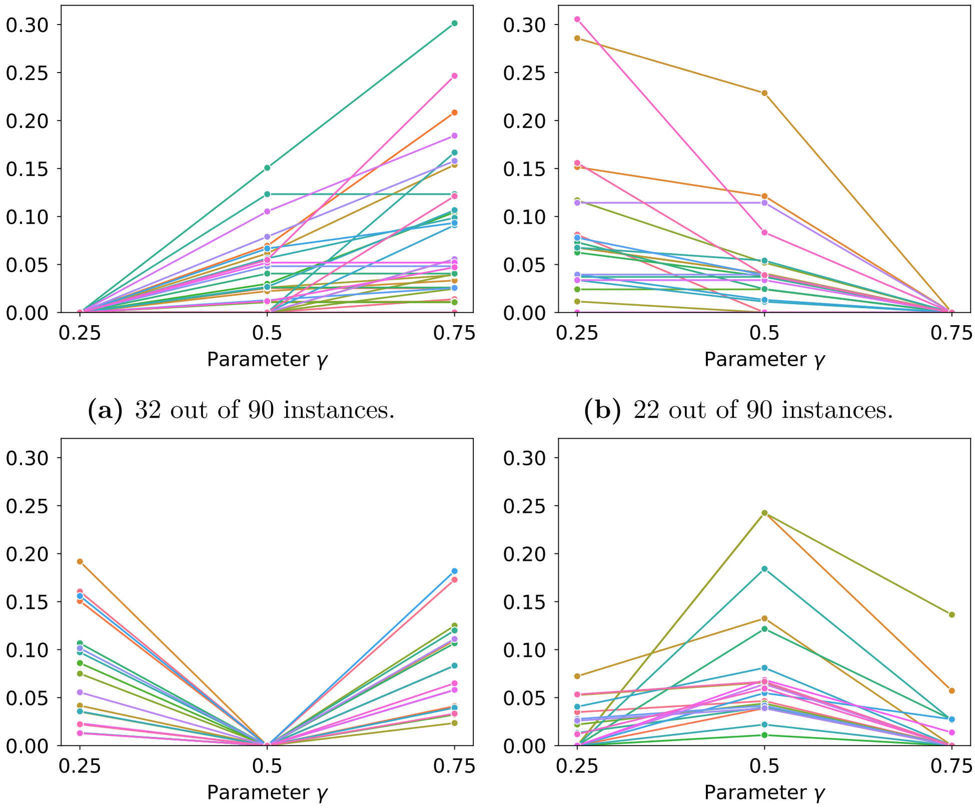


(a) Using an optimal fractional solution. **(b)** Using a heuristic fractional solution.

Figure 6.1.: Relative gaps according to the weight of the integer clustering for different values of parameter γ on Scenario *200m5_20percent*. Note that the gaps in each plot are calculated relative to the best solution of the algorithms considered in that plot, i.e. the gaps in between different plots are not directly comparable.

the subfigures according to their behavior. Figure 6.2a includes the instances with a monotone increasing behavior, i.e. for which increasing γ does not improve the results. For these instances, $\gamma = 0.25$ leads to the best results. The figure includes 32 of 90 instances. For two of these instances, all three choices of γ lead to the same solution. Figure 6.2b shows the contrary case, i.e. monotone decreasing behavior, and includes 22 out of 90 instances. Here, $\gamma = 0.75$ leads to the best results and lower γ does not improve the solution. Note that the two instances with the same result for each choice of γ are also included in this figure. Overall, a monotone behavior is encountered in 52 of 90 instances. However, there are also instances, where it is best to choose either $\gamma = 0.5$ or $\gamma \in \{0.25, 0.75\}$. Figure 6.2c shows the instances, for which $\gamma = 0.5$ is the best choice, and $\gamma \in \{0.25, 0.75\}$ leads to worse results. This behavior is observed in 18 of 90 instances. Figure 6.2d shows the remaining 20 out of 90 instances, which show neither monotonous behavior, nor have optimal values of $\gamma = 0.5$. Hence, in 38 of 90 instances the best choice shows nomonotonous behavior. Figure B.2 in the appendix is analogously structured and shows similar observations based on a heuristically found fractional solution. To conclude, the experiments show a benefit of the inclusion of the dual information ($\gamma > 0$), unless the influence of the dual information is not weighted too much ($\gamma = 1$). However, there is no clear recommendation, which inner value of $\{0.25, 0.5, 0.75\}$ to prefer.

The second parameter, λ, favors larger cluster sizes. Figure 6.3 shows the resulting gaps for different values of λ, given an optimal fractional solution (6.3a) and a heuristic fractional solution (6.3b), respectively. Since all values for $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$ and 10 distinct instances are depicted, each boxplot consists of 50 instances. The



(a) 32 out of 90 instances.

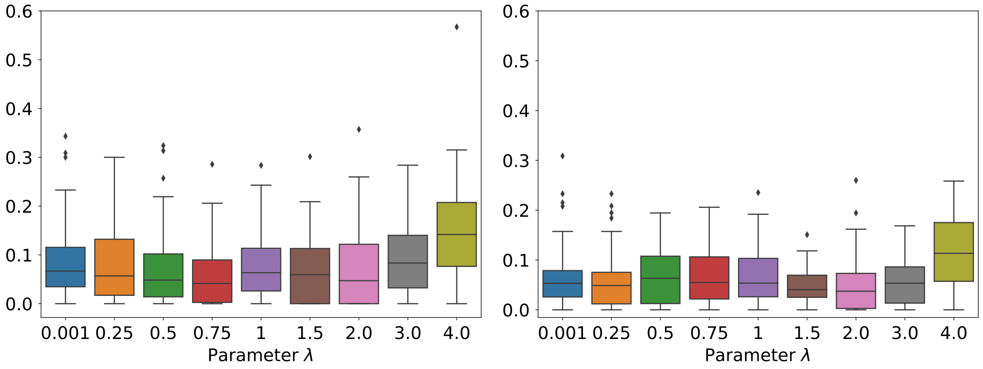
(b) 22 out of 90 instances.

(c) 18 out of 90 instances.

(d) 20 out of 90 instances.

Figure 6.2.: Relative gaps according to the weight of the integer clustering for selected values for γ on Scenario *200m5_20percent* given an optimal fractional solution.

best value for λ depends on the optimal sizes of the clusters, which are not known in advance and can significantly vary between the instances. However, choosing a too large value, i.e. $\lambda = 4$, leads to a larger gap and thus worse performance. This behavior is seen even more pronounced in Scenario *adult200m5*, which we show in Figure B.3 in the appendix. For this scenario, also $\lambda = 3$ leads to worse results. The choice of a large value for λ favors the inclusion of *many* data rows in the same cluster over the assignment of *similar* data rows. Although a larger number of vertices in a cluster never reduces the weight of the cluster, the inclusion of more vertices in the same cluster may have a positive effect on the overall weight of the cluster, as discussed in Section 4.4.2.



(a) Using an optimal fractional solution. **(b)** Using a heuristic fractional solution.

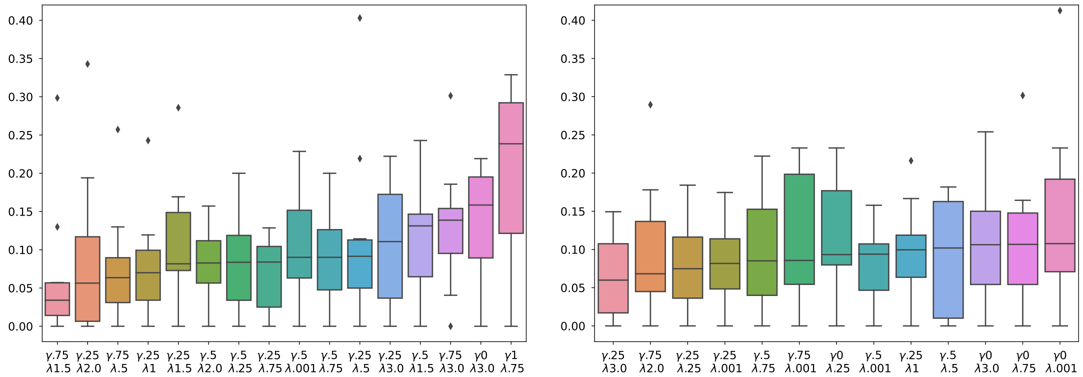
Figure 6.3.: Relative gaps according to the weight of the integer clustering for different values of parameter λ on Scenario *200m5_20percent*.

Finally, we compare the combinations of both parameters. The relative gap for selected parameter combinations is shown in Figure 6.4. For clarity, the figures only include combinations, which perform best for at least one instance. The boxplots are sorted in ascending order according to the median. We use the parameter pair with the lowest median, which is $\gamma = 0.75$, $\lambda = 1.5$ given an optimal and $\gamma = 0.25$, $\lambda = 3$ given a heuristic fractional solution for Scenario *200m5_20percent*. We follow the same procedure for Scenario *adult200m5* and report the results in Figure B.4.

We repeat the same procedure for different scenarios to obtain the parameter pairs, which result in a minimal relative gap, and report them in Table 6.5. The depicted parameter pairs are used for all further experiments.

6.2.4. Quality of the Proposed Partitioning

In the proposed method, the returned clustering of the microdata set depends on the fractional solution and the derived dual information. In this section, we investigate how the quality of integer solutions is affected if the fractional solution is optimal or only found heuristically. Moreover, we compare the objective values resulting from



(a) Using an optimal fractional solution.

(b) Using a heuristic fractional solution.

Figure 6.4.: Relative gaps according to the weight of the integer clustering for different combinations of parameters λ and γ on Scenario *200m5_20percent*.

	fractional solution	
Scenario	optimal	heuristic
adult200m5	$(0.5, 1.5)$	$(0.25, 0.5)$
200m5_2percent	$(0.75, 3.0)$	$(0.25, 0.25)$
200m5_20percent	$(0.75, 1.5)$	$(0.25, 3.0)$
200m5_100percent	$(0.75, 1.5)$	$(0.25, 1.5)$

Table 6.5.: Selected parameter pairs (γ, λ) for different scenarios.

solving the integer optimization problem to optimality in each iteration with solving it heuristically.

Table 6.6 reports the fractional and the integer objective values averaged over 10 samples for each scenario. The objective value is the sum of cluster weights and, thus, the smaller the better. First, in both steps, finding a fractional and finding an integer solution, the heuristically found solution is not as good as the exact solution, which is expected. In particular, solving the integer program to optimality performs better than the heuristic approach to the problem. For instance, for Scenario *200m5_2percent*, the exact solving leads to a clustering with a weight of 20.6 whereas the heuristic approach leads to a value of 48.6. However, the influence of using a heuristically found fractional solution instead of an optimal is low on the resulting integer solution. For instance, the objective value obtained by a heuristic fractional solution, 19.5, is even slightly better than the objective value by an optimal solution, 20.6. Due to the time savings, which a heuristic approach offers compared to solving to optimality, these observations suggest working with fractional solutions found heuristically. Another trend is that more confidentiality problems lead to higher objective values. Note that the proportion of confidentiality problems in Scenario *adult200m5* is 0.293 as shown in Table 6.3. In case of many confidentiality problems, it is necessary to assign also non-identical data rows to the same cluster. However, only elements that differ from each other but are assigned to the same cluster increase the objective value. Thus, it is expected that more confidentiality problems lead to higher objective values.

			integer solution	
Scenario	fractional solution		exact	heuristic
adult200m5	optimal	68.7	79.5	111.4
	heuristic	75.6	81.4	103.9
200m5_2percent	optimal	10.6	20.6	48.6
	heuristic	11.3	19.5	48.0
200m5_20percent	optimal	60.2	76.9	111.3
	heuristic	63.3	79.4	124.5
200m5_100percent	optimal	172.4	201.2	234.4
	heuristic	181.1	207.8	231.7

Table 6.6.: Fractional and integer objective values in the partitioning step averaged over 10 instances.

Table 6.7 breaks down the runtimes of the individual steps. It is clearly visible that by using heuristics in both steps, runtime can be remarkably decreased. In particular, the heuristic approach to the integer program leads to a large reduction of the required runtimes. The integer program iteratively determines clusters, which are added to the clustering. Only data records are taken into account, which have not been assigned yet. Hence, the integer program is called several times until all data records are assigned to some cluster. Approaching a solution heuristically can save a large amount of runtime. The depicted values are the overall runtimes required for all iterations in total. Moreover, the table indicates that the runtime is affected by the number of confidentiality problems in the original microdata set. The higher the number of confidentiality problems in the data set, the more time is required for the partitioning step. Note that the proportion of confidentiality problems in Scenario *adult200m5* is 0.293 as shown in Table 6.3.

	fractional solution		integer solution	
Scenario	exact	heuristic	exact	heuristic
adult200m5	165.11	19.00	69.58	0.47
200m5_2percent	16.16	10.55	29.04	0.40
200m5_20percent	64.48	13.91	54.66	0.44
200m5_100percent	368.02	23.55	117.06	0.55

Table 6.7.: Averaged runtimes [s] of the single phases in the partitioning step.

6.2.5. Quality of the Proposed Aggregation Methods

The task in the aggregation step of the microaggregation procedure is to find representatives for each cluster. In this section, we compare the exact and the two heuristic approaches to the aggregation problem, which are described in Chapter 5. The heuristic approaches are a greedy algorithm and a graph-based heuristic. To compare the quality, we use the χ^2-error, which is part of the objective function in the aggregation step. We name the proposed methods depending on the solution method in the fractional partitioning problem (exact/heuristic), in the integer problem in each iteration (exact/heuristic), and the used aggregation method (exact/greedy/clique heuristic). The name is generated as

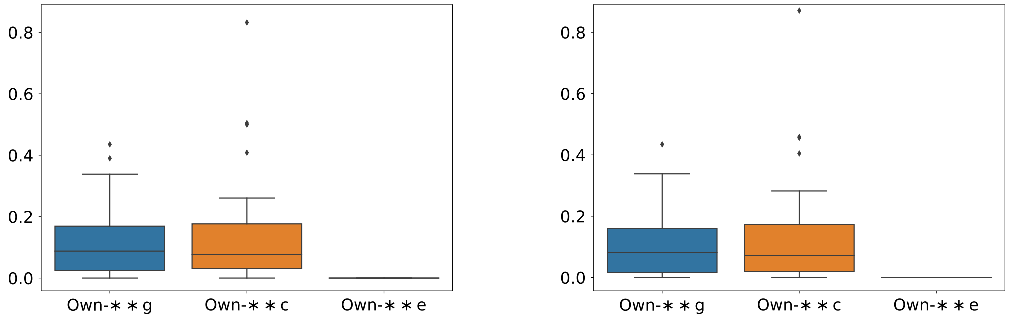
Own - *fractional solution method* - *integer solution method* - *aggregation method*

with

fractional solution method:	e - exact,	h - heuristic,	
integer solution method:	e - exact,	h - heuristic,	
aggregation method:	e - exact,	g - greedy,	c - clique heuristic.

Figure 6.5 depicts the comparison between the different aggregation approaches. The figures include all four combinations of solution methods in the partitioning step (exact/heuristic fractional solution method followed by exact/heuristic integer solution method), i.e. each boxplot contains 40 instances. The plots show the relative gap as defined in (6.1.1) in terms of the sum of χ^2-errors between table cells before and after

anonymization for Scenario *200m5_20percent*. All frequency tables up to dimension 3 serve as control tables and the respective result is shown in Figure 6.5a. Figure 6.5b shows the overall sum of χ^2-errors in *all* tables, which can be generated from the microdata set. The performances of the methods in these two cases are similar to each other. As expected, the exact solving of the aggregation problem leads to the smallest gaps and thus best results. The two heuristics yield comparable results for the majority of the instances. Figure B.5 in the appendix is analogously structured and displays the results for Scenario *adult200m5*. Notably, the exact solving does not necessarily lead to the best results in the sum of χ^2-errors over *all* tables as shown in Figure B.5b. This result does not contradict the optimality of the solution to the aggregation problem because the optimization problem only takes a subset of the frequency tables into account and aims to minimize the sum of χ^2-errors on these tables.

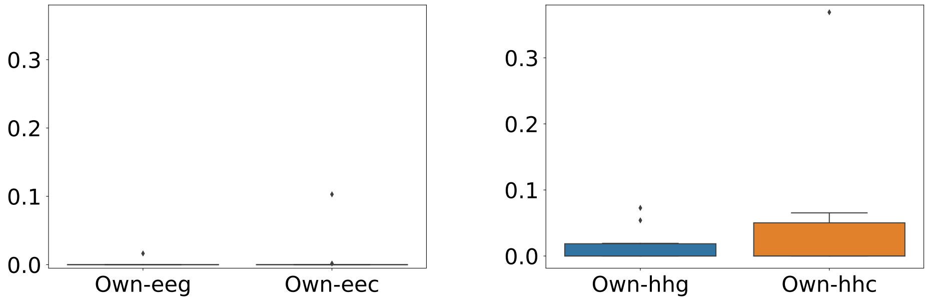


(a) On control tables, i.e. dimension ≤ 3.

(b) On all tables.

Figure 6.5.: Relative gaps according to the sum of χ^2-errors between table cells before and after anonymization for Scenario *200m5_20percent*.

We further investigate the two heuristics in a direct comparison in Figure 6.6. The relative gap is computed by only taking these two heuristics into account, i.e. it can be directly seen, which of these two returns the better solution. Section 6.2.4 points out that methods Own-∗e∗, i.e. exact solving of the integer programs, lead to a clustering of lower weight in the partitioning step compared to Own-∗h∗. Figure 6.6a shows the methods Own-ee∗, i.e. exact solving of the fractional and the integer program, and Figure 6.6b depicts the methods Own-hh∗, i.e. heuristically approaching both programs. Figure 6.6a indicates that if the clustering found in the partitioning step has a low weight and thus contains homogeneous clusters both greedy and clique heuristic algorithms lead to the same results, so that the clique heuristic is competitive to the greedy algorithm. Figure 6.6b shows that the greedy algorithm outperforms the clique heuristic, when the used clustering is of a higher weight. However, there are also instances, for which the clique heuristic outperforms the greedy algorithm, which is seen by the non-zero data points for method Own-hhg. We report results for other scenarios in the appendix in Figure B.6. For Scenarios *adult200m5* and *200m5_100percent*, the clique heuristic outperforms the greedy algorithm, when the partitioning is of good quality, i.e. methods Own-ee∗. For methods Own-hh∗, the greedy algorithm outperforms the clique heuristic. In the extreme case of only 2% confidentiality problems, which is Scenario *200m5_2percent*, both heuristics lead to the same results except for one outlier.

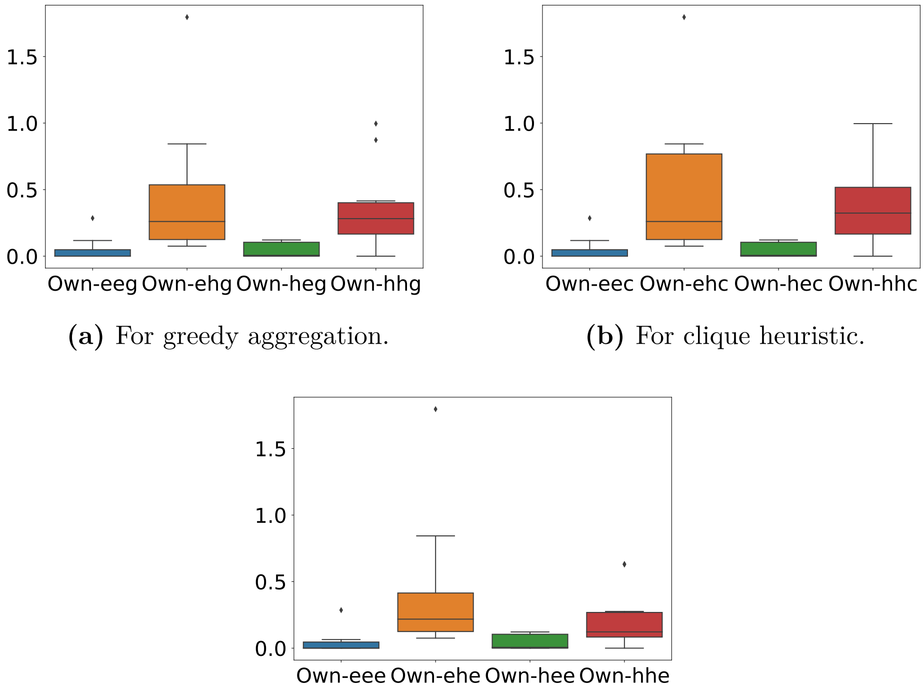


(a) Methods Own-ee∗.

(b) Methods Own-hh∗.

Figure 6.6.: Direct comparison of heuristic aggregation methods on the control tables on Scenario *200m5_20percent*.

Next, we compare the impact of the quality of the used clustering in detail. To this end, we analyze the resulting χ^2-errors in the frequency tables due to anonymization with respect to the approaches used in the partitioning step. Figure 6.7 shows the comparison between different methods in the partitioning step for the different aggregation methods, i.e. the greedy algorithm (Figure 6.7a), the clique heuristic (Figure 6.7b), and the exact aggregation method (Figure 6.7c). Again, we use the gap defined in (6.1.1) according to the resulting χ^2-errors in the control tables, which are all tables up to dimension 3. For all three aggregation methods qualitatively similar results are observed. The best results are mainly achieved by methods Own-ee∗, i.e. solving the single steps in the partitioning step exactly. However, also the methods Own-he∗ are competitive. On the other hand, approaching the integer program in the partitioning step only heuristically leads to worse results in most instances regardless of the aggregation method. These observations are consistent with the quality results for the proposed partitioning step, which showed that the quality of the resulting clustering highly depends on the approach to the integer program as discussed in Section 6.2.4. These findings suggest that the better the clustering of the microdata set, i.e. the more homogeneous the clusters, the less information is lost after the aggregation step. In order to support the claim that the quality of the clustering highly affects the quality of the anonymized microdata set, we compare our approach to a randomly found clustering in the next paragraph.



(a) For greedy aggregation.

(b) For clique heuristic.

(c) For exact aggregation.

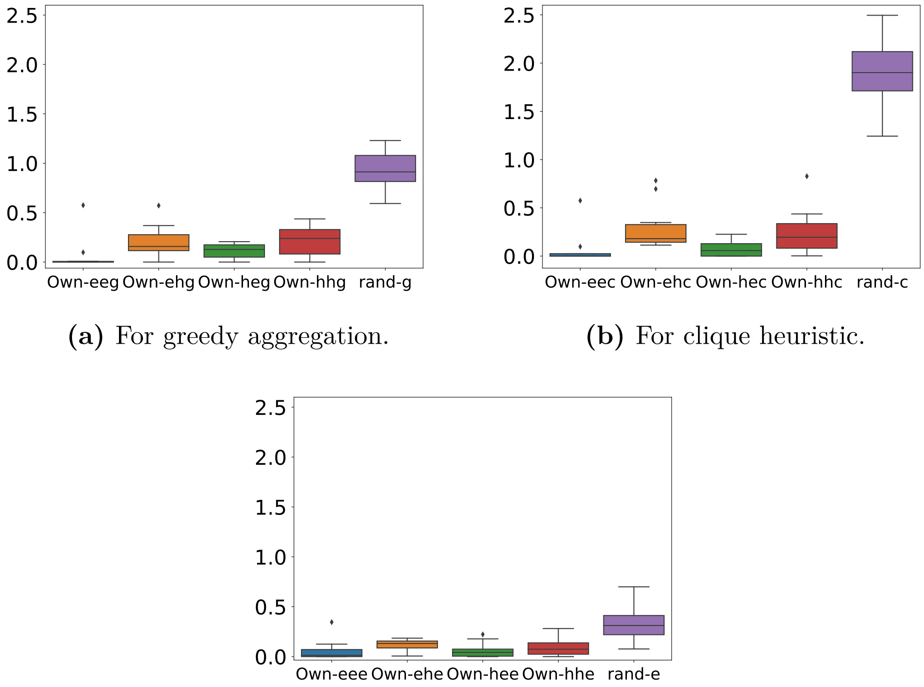
Figure 6.7.: Relative gaps according to the sum of chi squared errors in the control tables for Scenario *200m5_20percent*.

Using a randomly found clustering To verify that the quality of the clustering is crucial for the performance of the aggregation methods, we compare the proposed methods in the partitioning step to randomly found clusterings. This is an extreme scenario, in which the similarity is not taken into account at all in the partitioning step.

To find a random clustering of the microdata set we proceed as follows. We generate a random list of entries between k and $2k-1$ such that the sum of elements equals the number of data rows in the microdata set. This list contains the cluster sizes. Then, we sample the clusters by randomly drawing data rows without replacement according to the random cluster sizes. We denote the aggregation based on a randomly found clustering by rand-*aggregation method*.

Figure 6.8 indicates that for all instances, the aggregation based on a randomly found clustering is not competitive to the proposed partitioning methods. This reinforces that finding a reasonable clustering of the microdata set, i.e. clusters with similar data rows, in the first phase of the microaggregation procedure leads to better results. Another observation is that the influence of the clustering quality is stronger for heuristic approaches. In particular, the performance of the clique heuristic deteriorates when a randomly found clustering is used.

Runtimes for the aggregation step Table 6.8 breaks down the runtimes of the different aggregation methods. We have seen that the aggregation depends on the clustering quality, which itself depends strongly on the solution method (heuristic/exact) to the integer problem in the partitioning step. Solving the problem exactly leads to a clustering of a lower weight. Based on this, we grouped methods Own-hh$*$ and Own-eh$*$ together as well as methods Own-he$*$ and Own-ee$*$. Then, we averaged the runtimes for the aggregation step of all instances, which were solved within a time limit of 5000 seconds. We see that given a clustering with a lower weight, i.e. Own-$*$e$*$, the runtimes of the aggregation step are faster compared to a clustering with a larger weight, i.e. Own-$*$h$*$. The same effect can be observed for Scenario *200m5_20percent*, where the runtimes for each method increase, when a randomly found clustering is used. This behavior that the problem can be solved faster when the data rows in the same cluster are more similar can be explained by the fact that attributes, for which the data rows in the same cluster share the same value can be fixed to this value. Therefore, the more similar the data rows the faster the problem can be solved due to values that can be fixed. The reduced runtimes when values can be fixed due to similarity between data rows also explains the observed increase in runtimes with increasing number of confidentiality problems regardless of the used aggregation method. We observed that on small instances, which can be solved exactly, the exact method has faster runtimes than the heuristics because our implementation for the heuristics in python is slower than the commercial solver Gurobi used for exact solving. However, for instances with a large number of confidentiality problems, solving the problem to optimality becomes more challenging and using heuristics becomes



(a) For greedy aggregation.

(b) For clique heuristic.

(c) For exact aggregation.

Figure 6.8.: Relative gaps according to the sum of χ^2-errors after anonymization including a randomly found clustering for Scenario *200m5_20percent*.

justified. For Scenario *200m5_100percent*, only a subset of instances could be solved exactly within the time limit. After the time limit was reached for methods Own-∗h∗, the averaged duality gap is 0.05 and for methods Own-∗e∗, the averaged gap is 0.02. Moreover, we observe that the greedy algorithm scales better than the clique heuristic when instances with more confidentiality problems or more data rows are considered, e.g. on Scenario *200m5_100percent* and *500m5_20percent*.

		Aggregation method		
Scenario	Method	greedy	clique heuristic	exact
adult200m5	Own-∗h∗	3.56	5.47	0.72
	Own-∗e∗	2.71	3.69	0.54
200m5_2percent	Own-∗h∗	1.45	0.89	0.30
	Own-∗e∗	0.98	0.59	0.29
200m5_20percent	Own-∗h∗	2.85	2.92	0.53
	Own-∗e∗	2.10	2.02	0.45
	rand-∗	9.56	20.88	3.84
200m5_100percent	Own-∗h∗	6.22	16.80	942.55[*]
	Own-∗e∗	5.56	14.46	492.82[†]
500m5_20percent	Own-hh∗	6.38	26.20	3.60
	Own-he∗	4.54	16.12	1.55

[*] Averaged over 8 solved instances out of 20.

[†] Averaged over 17 solved instances out of 20.

Table 6.8.: Averaged runtimes [s] for the aggregation step, all tables up to dimension 3 are control tables.

6.2.6. Effect of Control Tables

As part of the proposed anonymization procedure, a subset of frequency tables must be designated as control tables. We show the effect of selecting frequency tables of different dimensions as control tables. In practice, tables of lower dimensions are of higher interest. Therefore, it is reasonable to determine a dimension and select all tables up to the determined dimension as control tables.

Figure 6.9 shows results on Scenario 200m5_20percent using method *Own-hhe*. It compares the χ^2-errors obtained when all tables up to dimension 2 are used as control tables, or when all tables up to dimension 3 are used. We then use the relative gap to indicate which case leads to better results. The relative gap shows that taking all tables only up to dimension 2 into account leads to less errors in the tables of dimensions 1 and 2. Consideration of all tables up to dimension 3 shows its advantages in higher dimensions. Not only the errors in 3-dimensional tables but also in the 4- and 5-

dimensional tables, which are not used as control tables, are reduced in the case of using all tables up to dimension 3.

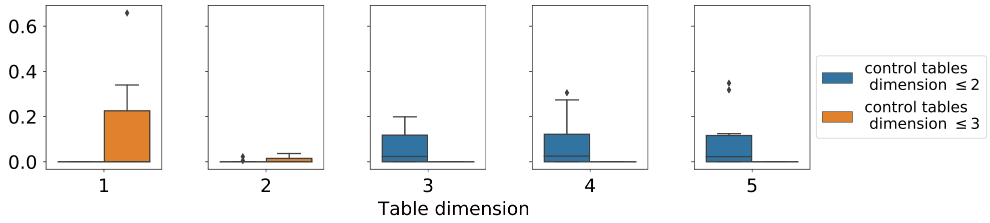


Figure 6.9.: Relative gaps according to the sum of χ^2-errors per table dimension for different control tables dimensions on Scenario *200m5_20percent* using method Own-hee.

Figure 6.10 shows that the total χ^2-error over all tables is lower when more frequency tables are selected as control tables. These results emphasize that the selection of the control tables can either lead to a focus on specific tables (e.g. of a very low dimension) or a smaller overall error. Thus, the control tables should be selected depending on the use case.

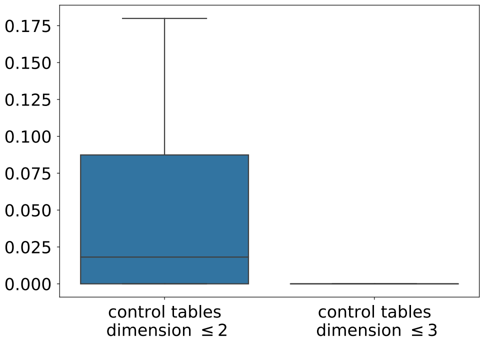


Figure 6.10.: Relative gaps according to the sum of χ^2-errors over *all* frequency tables on Scenario *200m5_20percent* using method Own-hee.

6.2.7. Comparison of Different Methods

In this section, we compare the proposed method to other methods to find k-anonymous data. The methods, which we use for comparison, are the SAFE method and exact methods, which we described in the previous chapters.

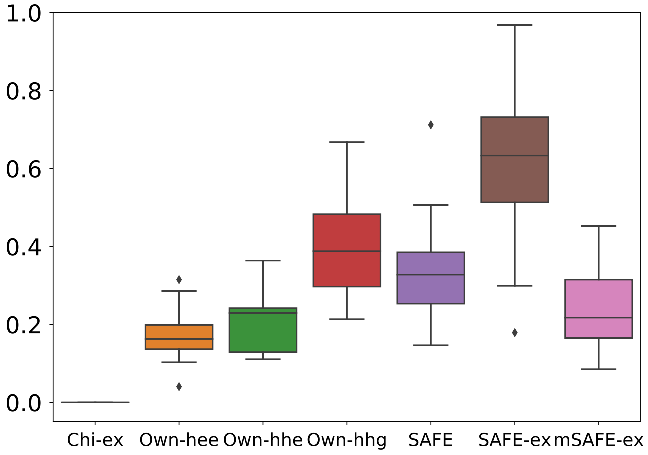
The SAFE heuristic was used in the last German census in 2011 and is described in Chapter 2. The re-implementation of the SAFE heuristic was done in python and to the best of our understanding. After contacting the author Höhne (2015), the parameters are selected according to his expertise and reported in Appendix A. Note that the objectives of the SAFE heuristic differ from the objective of the proposed method. While the proposed method targets the χ^2-errors in resulting frequency table, the SAFE method mainly aims at minimizing the maximum error. Moreover, the SAFE heuristic also takes other measurements into account like the mean error and the sum of errors. Besides the SAFE heuristic we compare the proposed method with the presented exact methods, which is only possible for sufficiently small instances. In Section 2.1, the SAFE-basic problem is described, which is the basis for the SAFE method. We refer to solving this problem exactly as SAFE-ex (SAFE-exact). We suggested a modification of the objective function to also include the deviation from the resulting frequency vector to the original frequency vector in Section 3.2. We refer to solving this modified optimization problem as mSAFE-ex (modified SAFE-exact). We showed a problem formulation minimizing the sum of χ^2-errors in Section 5.6 and refer to solving it to optimality as Chi-ex (χ^2-error-exact).

In the following, we focus on synthetic data. The results on the randomly drawn samples, i.e. Scenario *adult200m5*, are shown in the Appendix B. They are similar to the results on the synthetically generated Scenario *200m5_20percent* and can be found in Figures B.7, B.8 and B.9.

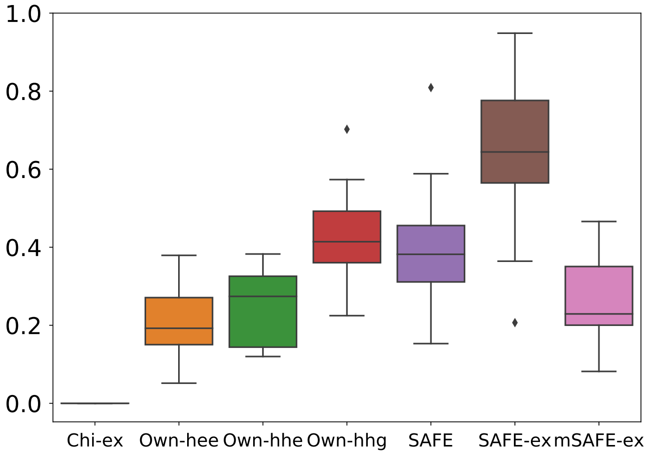
Method Comparison on Scenario *200m5_20percent*

Figure 6.11 shows the comparison of the described methods according to the relative gap defined in (6.1.1) with respect to the sum of the χ^2-errors in the control tables on Scenario *200m5_20percent*. All tables up to dimension 3 are selected as control tables. Figure 6.11a depicts the results on the control tables. Figure 6.11b shows the evaluations on all tables, i.e. including tables, which are not selected as the control tables. The results are comparable to each other. The exact method Chi-ex, which combines both steps in one optimization problem, leads to the best results in both, the set of control tables and the set of all tables. Note that the χ^2-error is included in the objective function of Chi-ex. In contrast, the exact methods SAFE-ex and mSAFE-ex aim to minimize the maximum absolute error. Therefore, Chi-ex is expected to outperform these two exact methods on this criterion. However, comparing SAFE-ex and mSAFE-ex with each other, the proposed modified version mSAFE-ex shows its benefits and clearly outperforms SAFE-ex. Moreover, the SAFE method also outperforms SAFE-ex because the SAFE heuristic takes more criteria into account than SAFE-ex. The two proposed methods Own-hee and Own-hhe are competitive to mSAFE-ex and outperform the SAFE method in terms of the χ^2-error. However, the greedy algorithm in the aggregation step, i.e. Own-hhg, loses quality compared to the exact aggregation method and the SAFE method slightly outperforms Own-hhg.

The χ^2-error is part of the objective of the proposed method whereas the methods SAFE, SAFE-ex, and mSAFE-ex focus on the minimization of the maximum absolute error in the control tables. Therefore, we compare the resulting maximum absolute errors in the frequency tables of different dimensions in Figure 6.12. All tables up to



(a) On the control tables.



(b) On all tables.

Figure 6.11.: Relative gaps according to the sum of χ^2-errors for different methods on Scenario *200m5_20percent*.

dimension 3 were selected as control tables. As described in Chapter 2, the SAFE method puts emphasis on the one-dimensional tables. This is reflected in the results as in the comparison to different methods, the SAFE method manages to generate the smallest maximum error for one-dimensional tables. It seems surprising that the exact method SAFE-ex does not outperform the SAFE method regarding the control tables. However, the objective function does not only consist of the maximum absolute error but also includes an allowed extra error g_t for each table cell $t \in \mathcal{T}$. In the SAFE heuristic, whenever a solution is found, an improvement algorithm is started, which potentially reduces the resulting maximum error. In the method SAFE-ex, these additionally allowed errors are predetermined and fixed values, which are not reduced during the process. Therefore, these results do not contradict the optimality of the solution to SAFE-ex. In terms of the maximum absolute errors, the SAFE method outperforms the proposed methods in most of the cases. However, in tables of higher dimension the proposed methods lead to competitive results even though these tables were not included in the control tables. Moreover, the maximum absolute error decreases for larger table dimensions for all methods except for SAFE, which emphasizes the one-dimensional tables. This can be explained by the fact that the lower the dimension the larger the entries of the single table cell entries in general. For methods SAFE-ex and mSAFE-ex, the additional error g_t for each control table cell $t \in \mathcal{T}$ is defined in dependence of the original cell entry and is larger for larger original entries as described in Section 2.1. Therefore, for cells with a larger original entry, a larger absolute error is allowed. Chi-ex and the proposed methods aim to minimize the χ^2-errors and allow therefore a larger absolute error for cell entries with a larger original entry.

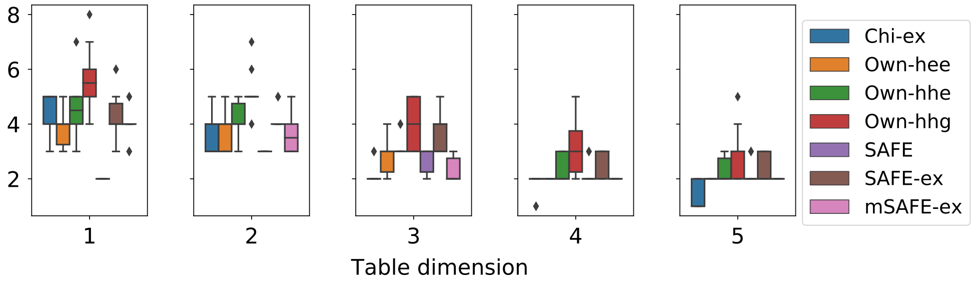


Figure 6.12.: Maximum absolute error in the frequency tables per dimension on *200m5_20percent*.

It can be argued that the maximum error is a reasonable measurement because it gives a valid upper bound for the errors in all tables of a dimension. Usually, data users are only interested in few tables. However, considering relative measurements is instructive because an error in a table field with an original very low value is weighted more heavily than the same error in a field with an original high value. Figure 6.13 depicts the relative gap according to the maximum *relative* error per table dimension.

The proposed methods show competitive results compared to other methods in terms of the maximum relative error also in the tables, which have not been selected as control tables. Moreover, in contrast to the maximum absolute error, the maximum relative error is lowest for one-dimensional tables. This behavior is expected as in general lower table dimensions correlate with larger original table cell entries and for larger original cell entries, similar absolute errors lead to smaller relative errors.

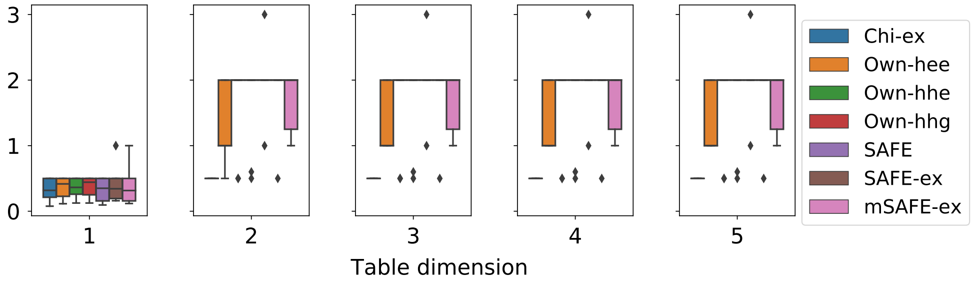
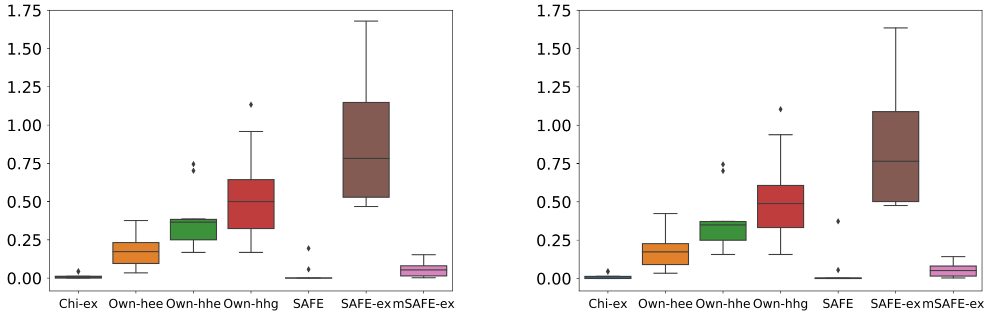


Figure 6.13.: Maximum relative error in the frequency tables per dimension on *200m5_20percent*.

Method Comparison on Scenario *200m5_2percent*

Scenario *200m5_2percent* is an extreme cases with a very low proportion of confidentiality problems. In the following, we study the comparison of different methods on Scenario *200m5_2percent*.

Figure 6.14 shows the method comparison based on the relative gap with respect to the sum of χ^2-errors in the control tables (6.14a) and all tables (6.14b), respectively. All tables up to dimension 3 are selected as control tables. In this scenario, the SAFE method shows its strength and leads to best results except for single outliers in comparison to the other methods. In particular, it might be surprising that the SAFE method outperforms Chi-ex. However, in contrast to Chi-ex, the SAFE method does not ensure that the number of data rows is preserved, as discussed in Section 3.2. Table 6.9 breaks down the number of data rows after anonymization by SAFE. Only 2 out of the 10 instances preserve the number of 200 data rows. Hence, 8 out of the 10 solutions found by SAFE are not feasible for Chi-ex due to Constraints (5.6.3). Preserving the number of data rows ensures plausibility and consistency with the original microdata set. In the case of a very small deviation from the original number, one could argue for relaxing this constraint to increase the data quality. However, it is reasonable that the number of data rows after anonymization should remain at least close to the original number to ensure reasonableness. In conclusion, the good performance of SAFE does not contradict the optimality of the solutions found by Chi-ex with respect to the optimization problem (5.6.1)–(5.6.5). Moreover, the comparison between SAFE-ex and mSAFE-ex underlines the benefits of the modification, as discussed in the previous section. The proposed methods show a similar behavior as for Scenario *200m5_20percent*.



(a) On the control tables.

(b) On all tables.

Figure 6.14.: Relative gaps according to the sum of χ^2-errors for different methods on *200m5_2percent*.

n_{new}	199	200	201	202
Number of instances	4	2	2	2

Table 6.9.: Number of instances with n_{new} data rows after anonymization by SAFE for Scenario *200m5_2percent*.

Figure 6.15 depicts the maximum error per table dimension. Note that the maximum error is not taken into account in the proposed methods. Comparing the proposed methods with each other, method Own-hhe and especially method Own-hee outperform method Own-hhg. In method Own-hhg, all steps of the proposed method are approached heuristically. For Scenario *200m5_2percent*, similar results are observed as for Scenario *200m5_20percent* described in the previous section. For all table dimensions, Chi-ex, SAFE and mSAFE-ex lead to competitive results. Again, the proposed methods lead to a smaller maximum absolute value for higher table dimensions. As discussed for Scenario *200m5_20percent*, the original entries in these tables are in general smaller and, thus, errors are penalized more due to the definition of the χ^2-error.

Figure 6.16 shows the maximum relative error in the frequency tables with respect to the table dimensions. The different methods are similar to each other and the lowest maximum relative error is observed for one-dimensional tables.

Method Comparison on Scenario *200m5_100percent*

In the previous section we evaluated different methods on microdata sets with a very low proportion of only 2% confidentiality problems. Now, we analyze Scenario

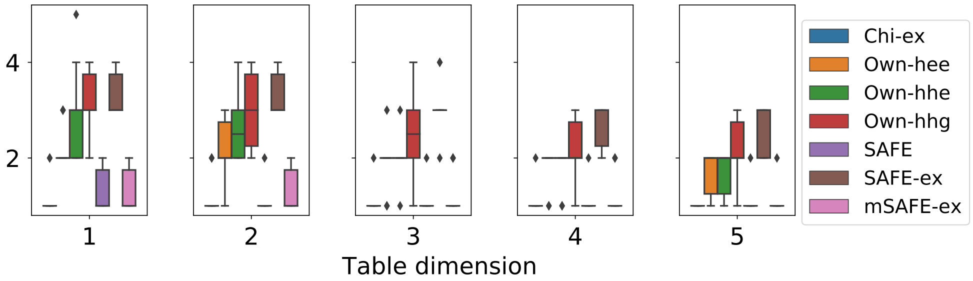


Figure 6.15.: Maximum absolute error in the frequency tables per dimension on *200m5_2percent*.

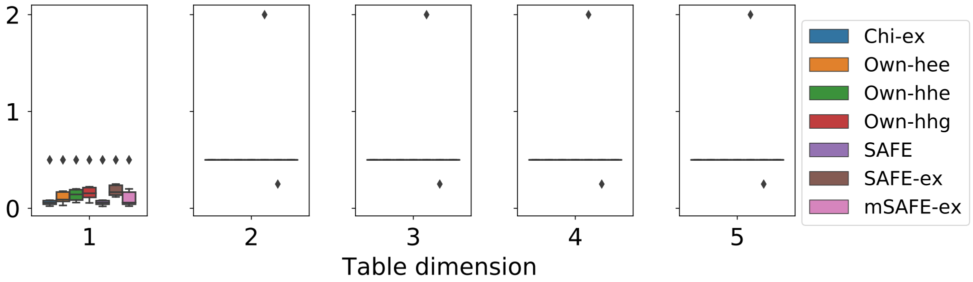
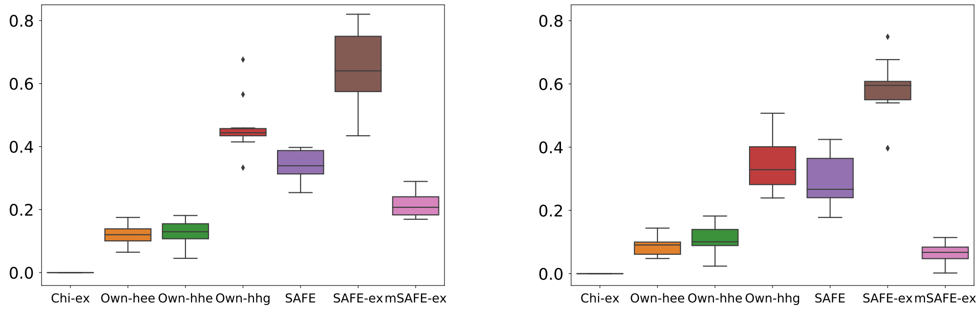


Figure 6.16.: Maximum relative error in the frequency tables per dimension on *200m5_2percent*.

200m5_100percent, which is the other extreme case with microdata sets consisting only of confidentiality problems.

Figure 6.17 displays the results with respect to the sum of χ^2-errors for selected methods. We set the time limit for all experiments to 5000s. For the method Chi-ex, the time limit was reached for all instances and the duality gap at the time of termination was on average 0.51. Nevertheless, method Chi-ex leads to the best results. Note that the duality gap provides a score on how much the found solution could theoretically be improved based on the best bounds found so far. Again, the benefits of the modification mSAFE-ex compared to SAFE-ex are visible. In contrast to the extreme scenario *200m5_2percent*, the methods Own-hee and Own-hhe outperform the SAFE method regarding the sum of χ^2-errors in the control tables as well as in all tables. However, the greedy algorithm for the aggregation problem, i.e. Own-hhg, loses quality compared to the exact aggregation in Own-hhe. These results are comparable to Scenario *200m5_20percent*.



(a) On the control tables.

(b) On all tables.

Figure 6.17.: Relative gaps according to the sum of χ^2-errors for different methods on *200m5_100percent*.

Figure 6.18 depicts the maximum absolute error for Scenario *200m5_100percent*. The results are comparable to Scenario *200m5_20percent*. The SAFE method returns the best results, in particular for low-dimensional tables. The larger the table dimension, the lower the maximum absolute error and the more competitive the proposed methods Own-hee, Own-hhe and mSAFE-ex.

The maximum relative error is depicted in Figure 6.19. The proposed methods are competitive to the other methods Chi-ex, SAFE and SAFE-ex. In particular, the proposed methods Own-hee and Own-hhe outperform the SAFE method for larger table dimensions. Similar to the previously analyzed scenarios with less confidentiality problems, also for Scenario *200m5_100percent* the maximum relative error is lowest for one-dimensional tables.

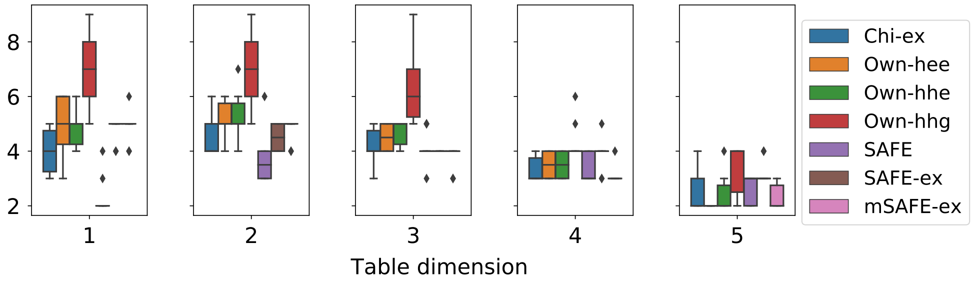


Figure 6.18.: Maximum absolute error in the frequency tables per dimension on *200m5_100percent*.

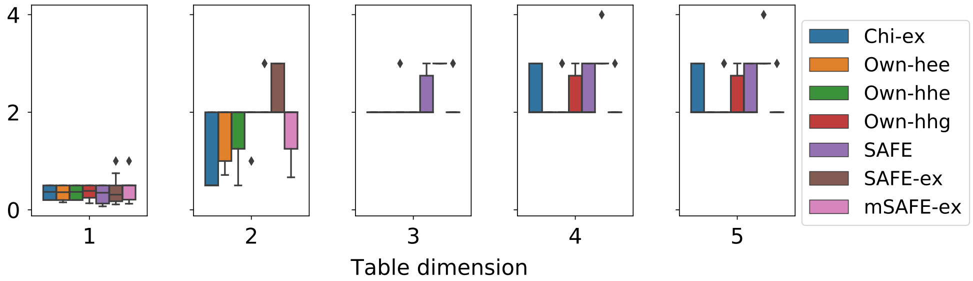


Figure 6.19.: Maximum relative error in the frequency tables per dimension on Scenario *200m5_100percent*.

Runtimes of Different Methods

Table 6.10 displays the averaged runtimes of the different methods for the experiments, which are presented in the previous sections. For the average calculation, experiments, which reached the time limit of 5000s, are excluded. For Scenario *adult200m5*, Chi-ex reached the time limit for one instance with a duality gap of 0.12. Moreover, Chi-ex reached the time limit for all instances of Scenario *200m5_100percent* with an averaged duality gap at the time of termination of 0.51. The exact methods SAFE-ex and mSAFE-ex can solve the instances faster than Chi-ex. Note that all three optimization problems are integer programs, which are in general not easy to solve. Since the optimization problem for Chi-ex includes the χ^2-error in the objective function, the problem remains non-linear even when the integer constraints are relaxed to linear constraints. Also the objective functions of SAFE-ex and mSAFE-ex contain a non-linear function, which is the absolute value. However, it is known that the absolute value can be linearized by introducing auxiliary variables. Therefore, it is not surprising that SAFE-ex and mSAFE-ex are faster than Chi-ex. A further aspect we observe is that for Scenarios *200m5_2percent* and *200m5_20percent*, method Own-hhe, which solves the aggregation problem exactly, is faster than Own-hhg, which uses the greedy algorithm in the aggregation step. Note that the heuristics are implemented in python, while the commercial solver Gurobi is used to solve the problems exactly. Therefore, it is not surprising that for instances, which are not too challenging, the exact solving can be faster. However, for Scenarios *adult200m5* and *200m5_100percent*, which have a higher number of confidentiality problems, the heuristic approach to the aggregation problem, i.e. Own-hhg, saves a large amount of time. From the table, it can be derived that all methods require a longer runtime for instances with a larger number of confidentiality problems. Note that the averaged proportion of confidentiality problems for Scenario *adult200m5* is 0.293.

Scenario	Chi-ex	Own-hee	Own-hhe	Own-hhg	SAFE	SAFE-ex	mSAFE-ex
adult200m5	1330.17*	61.87	20.54	23.37	8.65	2.25	2.62
200m5_2percent	1.07	42.78	11.34	12.55	1.27	0.99	1.00
200m5_20percent	6.80	53.71	15.07	17.39	4.45	1.95	1.98
200m5_100percent	–†	749.81‡	987.54‡	30.68	46.08	5.56	19.51

* Averaged over 9 solved instances out of 10.

† All instances reached the time limit of 5000s.

‡ Averaged over 8 and 17 solved instances out of 20, respectively, cf. Table 6.8.

Table 6.10.: Averaged runtimes [s] for the different methods (all tables up to dimension 3 are control tables).

6.2.8. Logistic Regression

Another performance indicator of anonymization methods is the behavior of the anonymized in comparison to the original microdata set in regression analysis. The logistic regression model is used to predict the probability of a categorical variable to occur depending on one or more independent variables. In this section, we examine the effects of anonymization on a multinomial logistic regression.

We draw 100 random samples from the ‘Adult’ data set according to Scenario *adult200m5*, i.e. 200 data rows and 5 attributes, and select one of the 5 attributes as dependent variable. Note that a logistic regression model is not necessarily suitable for each variable. To identify models of very low quality, we investigate the model quality using the Pseudo-R^2 value of the whole data set with 30162 data rows. In case of a very low Pseudo-R^2 value, i.e. a very low model quality, the comparison of estimated probabilities between anonymized and original microdata sets remains inconclusive because the prerequisites for the logistic regression model are not fulfilled. In all other cases, we do not interpret the Pseudo-R^2 value on its own but focus on the comparison between the estimations based on the microdata sets before and after anonymization. We compare the anonymization methods SAFE, Own-hhe and Chi-ex to each other.

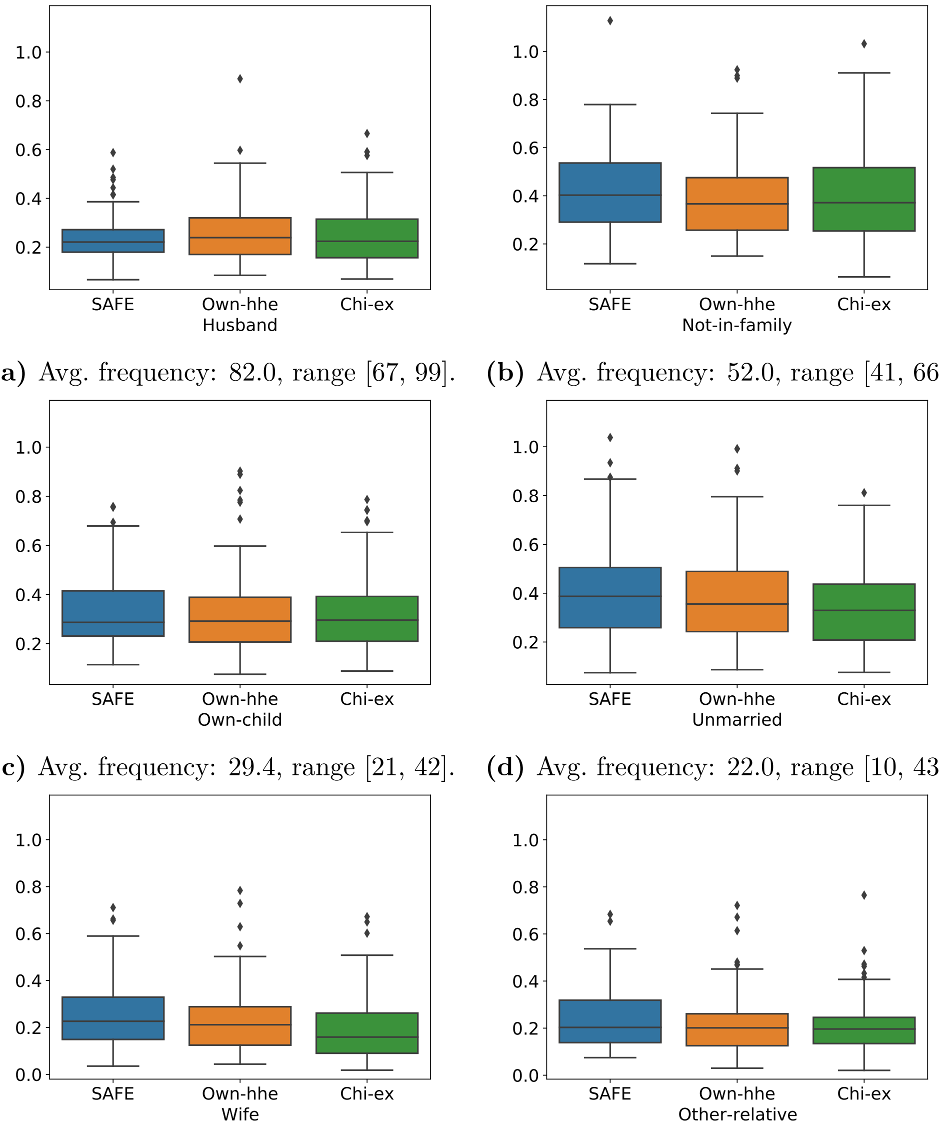
Figure 6.20 shows the ℓ_2-distances between the probability estimations calculated by a multinomial logistic regression model before and after anonymization for attribute 3 as the dependent variable. For attribute 3, the Pseudo-R^2 value with respect to the full microdata set is 0.602. This attribute has six distinct values and each subplot corresponds to one of these attribute values. The anonymization methods SAFE, Own-hhe, and Chi-ex show comparable results, where the exact method Chi-ex performs only slightly better than the two heuristic approaches. The figures include the averaged frequency and of each value and its range in the drawn samples.

Figure B.10 in the appendix is analogous with attribute 1 as the dependent variable, which has two distinct values. Again, the three methods SAFE, Own-hhe and Chi-ex are competitive with each other.

We also report the results for attribute 5 as the dependent variable in Figure B.11 in the appendix, where a supposedly surprising behavior can be observed that the exact method Chi-ex performs remarkably worse than the heuristic approaches for value *Private*. However, the respective Pseudo-R^2 value of 0.032 is very low and suggests that the multinomial logistic regression model does not fit even for the original data set. Therefore, as mentioned above, this attribute with a very low Pseudo-R^2 value should be neglected.

The experiments on the two remaining attributes are as follow. Attribute 2 can be neglected due to a very small Pseudo-R^2 value of 0.042. Attribute 4 with a Pseudo-R^2 value of 0.584 shows comparable results to attributes 1 and 5.

Overall, the two heuristic approaches SAFE and Own-hhe are competitive with each other and also with the exact method Chi-ex in terms of multinomial logistic regression.



(a) Avg. frequency: 82.0, range [67, 99]. **(b)** Avg. frequency: 52.0, range [41, 66].

(c) Avg. frequency: 29.4, range [21, 42]. **(d)** Avg. frequency: 22.0, range [10, 43].

(e) Avg. frequency: 9.3, range [4, 17]. **(f)** Avg. frequency: 5.5, range [0, 11].

Figure 6.20.: ℓ_2-distances between estimated probabilities before and after anonymization for different attribute values over 100 samples on Scenario *adult200m5* with attribute 3 as dependent variable, Pseudo-R^2 = 0.602.

6.2.9. Increasing the Size of the Instances

This section deals with the dependencies of the runtimes of the proposed methods on different aspects of the instances, which include the number of data rows in the microdata set, the parameter k for k-anonymity and the number of control tables. In this section, we show the runtimes of method Own-hhg.

First, we inspect the dependencies of the runtimes on the number of data rows, which are contained in the original microdata set. Analogously to Scenario *200m5_20percent*, we generated 10 instances for each size with 20% confidentiality problems with respect to the number of data rows. Figure 6.21 illustrates the total runtimes of method Own-hhg depending on the number of data rows. The more data rows in the microdata set, the more runtime is required for anonymization. Moreover, we added a quadratic curve fitted to the observed runtimes, which is illustrated by the orange line. The figure implies that the method Own-hhg has at least quadratic runtime with respect to the number of data rows.

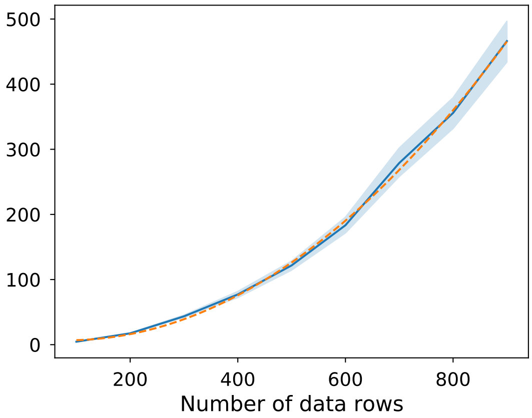
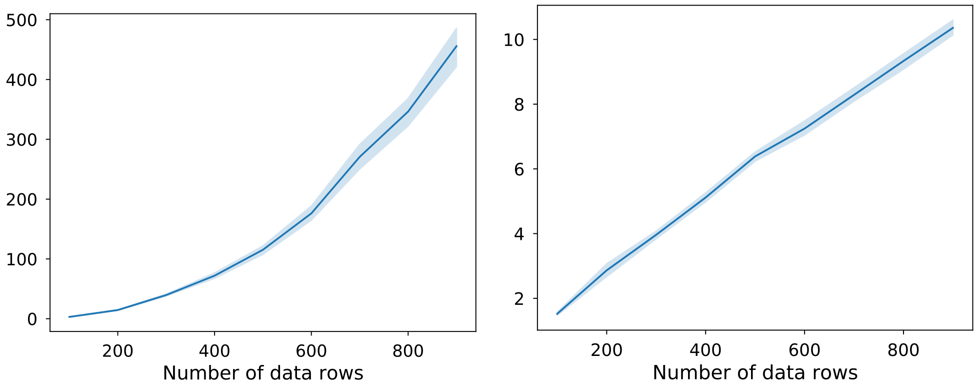


Figure 6.21.: Runtimes [s] of method Own-hhg with respect to different sizes of the microdata set.

In order to get a deeper understanding of the runtime behavior, we present the runtimes of the individual steps of the microaggregation procedure in Figure 6.22. The runtime for the partitioning step is shown in Figure 6.22a. The curve is similar to the total runtimes, which is expected as the partitioning step has the largest impact on the runtimes (see Tables 6.7 and 6.8). The aggregation step is depicted in Figure 6.22b and shows a linear behavior with increasing number of data rows.

Next, we study the effects on the runtimes for different choices for parameter k, which defines k-anonymity. In the previous experiments we focused on $k = 3$ since this value is also used in the anonymization of the German census in 2011. For the comparison of different values for parameter k, we generated 10 instances analogously to *200m5_20percent* for each parameter choice, such that the data sets contain 20% confidentiality problems with respect to parameter k. Figure 6.23 illustrates that

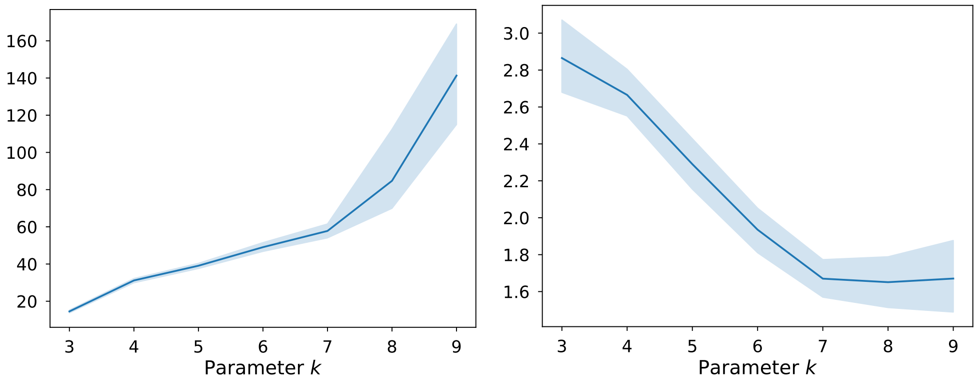


(a) Partitioning step.

(b) Aggregation step.

Figure 6.22.: Runtimes [s] of single phases of method Own-hhg with respect to different sizes of the microdata set.

parameter k has an opposite effect on the two steps of the microaggregation procedure. While the runtime of the partitioning step increases, depicted in Figure 6.23a, the runtime of the aggregation step decreases with increasing k. The increased runtime for the partitioning step is expected because the variables of the optimization program of the partitioning problem are all possible clusters, which have a size between k and $2k-1$, as discussed in Chapter 4. The reduced runtime for the aggregation step is also reasonable because for larger k, less clusters are built in the partitioning step and the greedy algorithm in the aggregation step iterates over all clusters, so that the runtime decreases with a lower number of clusters.



(a) Partitioning step (hh).

(b) Aggregation step (g).

Figure 6.23.: Runtimes [s] of single phases of method Own-hhg depending on different values for parameter k.

Finally, we also investigate the choice of the control tables, which only affects the aggregation step. Figure 6.24 shows the runtime of the aggregation step depending on the dimension, up to which all frequency tables are selected as control tables for Scenario *200m8_20percent*. Note that for each dimension d there are $\binom{8}{d}$ frequency tables. Thus, the highest increase in the runtime is observed, when the most tables are added as control tables, i.e. when all tables up to dimension 4 are selected as control tables. There is exactly one 8-dimensional table. Therefore, selecting all tables up to dimension 7 and up to dimension 8 only differs in one table and the runtime is not considerably affected.

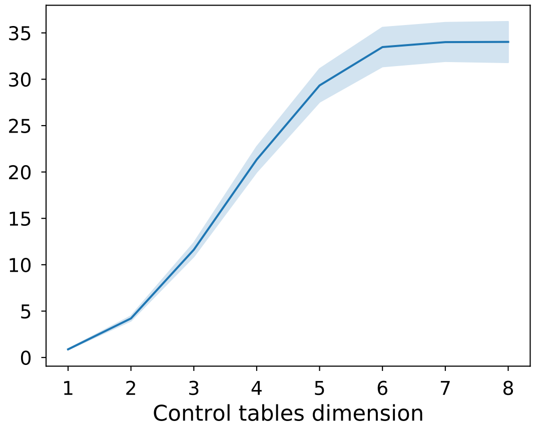


Figure 6.24.: Runtimes [s] of aggregation step in method Own-hhg with respect to the control tables dimension.

6.3. Summary

In this chapter, we presented computational experiments for the evaluation of the proposed methods. We used randomly drawn samples from census data as well as synthetic data, for which the proportion of confidentiality problems is prescribed. We focused on a proportion of 20% confidentiality problems but also included the extreme cases of a very low (2%) and a very high (100%) proportion of confidentiality problems in our experiments.

We analyzed the individual steps of the proposed microaggregation procedure. For the partitioning step, we studied the effects of the parameter choice on the result quality using the weight of the clustering and found that to find an integer clustering, it is beneficial to weight the dual information provided by an optimal fractional solution more than the dual information provided by a heuristically found fractional solution. However, in both cases, the inclusion of the dual information, i.e. $\gamma > 0$, is beneficial. Furthermore, the quality of an integer clustering mainly depends on the solution approach to the integer program. In some scenarios, a heuristically found fractional solution even leads to better integer clusterings than an optimal fractional solution. The best choice for the second parameter λ, which is a weighting parameter for the cluster sizes, highly depends on the different instances. In all cases, an excessively high value for this parameter has adverse effects, as it favors a larger cluster size at the expense of cluster homogeneity.

The heuristic approaches to the aggregation step are not as good as the exact method. However, in the direct comparison of the heuristic approaches, we have seen that there are instances, for which an approach outperforms the other approach, for both heuristics. Both heuristics perform better, when the fractional clustering has a lower weight, i.e. consists of clusters of more similar elements. However, this effect is more pronounced for the clique heuristic. This observation that both heuristics perform better on more homogeneous clusters was further supported by the comparison with a random clustering. Using a random clustering deteriorates the performances of all aggregation methods. However, the difference in performance is the highest for the clique heuristic. Regarding the runtimes, we have seen that the runtimes of all aggregation methods depend on the quality of the clustering of the microdata set.

Furthermore, we studied the effects of control tables and found that fewer selected tables lead to smaller errors in control tables. However, selecting more frequency tables as control tables can reduce the overall errors and the errors in tables of higher dimension, which are not selected as control tables.

We compared the proposed methods Own-hee, Own-hhe and Own-hhg to the SAFE heuristic and the exact methods Chi-ex, SAFE-ex and mSAFE-ex. In terms of the relative gap to the best found solution, the proposed methods showed similar results for the different scenarios with varying proportions of confidentiality problems. The exact method Chi-ex mostly returned best results for relative measurements, i.e. the sum of χ^2-errors and the maximum relative error. In these measurements, the proposed methods were competitive to or outperformed the SAFE heuristics. However, for Scenario *200m5_2percent*, the SAFE method returned the best results even compared to Chi-ex. Note that the Chi-ex method ensures the preservation of the true number of data rows while SAFE does not necessarily satisfy this constraint, so that the better results of the SAFE heuristic do not contradict the optimality of the solutions found by Chi-ex. Moreover, in all evaluations, mSAFE-ex outperformed SAFE-ex, which shows the benefit of the modification proposed in Section 3.2. The SAFE method outperformed the other methods in terms of the maximum absolute error per table dimension, in particular, for one-dimensional tables.

Finally, we also analyzed the runtimes of the proposed method Own-hhg depending on the number of data rows, the anonymization parameter k and the number of selected control tables. The runtime of the partitioning step on the one hand scales at least quadratic in the number of data rows. Also increasing parameter k leads to increasing runtimes in the partitioning step since the variables in the partitioning problem are all clusters, which have a size between k and $2k-1$. The aggregation step on the other hand shows a linear runtime in the number of data rows. With increasing parameter k, the runtime of the greedy heuristic in the aggregation step even decreases because less clusters are built for larger k and, thus, the greedy algorithm requires less iterations. Furthermore, the aggregation step requires a longer runtime, when more tables are selected as control tables. However, the overall runtime of the microaggregation procedure is dominated by the partitioning step.

Taken together, our experiments show that our modifications improve the result quality of SAFE-ex. Furthermore, while solving the problem exactly outperforms heuristic approaches for small data sets, heuristic approaches are necessary for larger data sets. Moreover, we also experimentally compared the effect of different parameters on the result quality and performance and found for example that cluster homogeneity is crucial for high quality and performance, higher proportion of confidentiality problems decreases result quality and performance and that it is beneficial to include dual information in the optimization problem.

Chapter 7

Conclusion

In this research, an integer programming approach is proposed to generate a k-anonymous microdata set with nominal data with the aim to lose as little information as possible. We proposed a new method to achieve k-anonymity for nominal data, which follows the classical procedure of microaggregation methods but is tailored to nominal data.

In the first part of this thesis, we outlined the SAFE heuristic, which was used to anonymize the last German census data from 2011 as described by Höhne (2015). The author presents an optimization problem, which is the basis for the SAFE heuristic. We analyzed the complexity of this SAFE-basic problem. Specifically, we proved $\mathcal{NP}$-hardness for a slightly modified variant of the SAFE-basic problem. Based on this complexity result, we provided justification for using heuristic approaches for larger data sets. Furthermore, we pointed out that the SAFE-basic problem does not include the minimization of some criteria like the mean error in frequency tables or the deviation from the original frequency vector, which are nevertheless addressed by the SAFE heuristic. Based on these findings, we proposed a modification of the SAFE-basic problem, which includes the deviation between the frequency vectors before and after anonymization. In our computational experiments, the advantages of the proposed modified optimization problem over the SAFE-basic problem with respect to resulting errors in frequency tables were profound. Moreover, we discussed that the SAFE heuristic does not ensure that the number of data rows is preserved, which is, however, required for a solution to the SAFE-basic problem. On the one hand, this requirement preserves consistency with the original microdata set. On the other hand, we have seen cases, i.e. Scenario *200m5_2percent*, where it might be worth considering to relax the requirement in order to increase the quality of the anonymized data. However, we find it useful to ensure that the number of data rows remains at least close to the original number for data plausibility.

In the subsequent part of this thesis, we proposed a data-perturbative microaggregation method for nominal data. In the past, microaggregation methods were primarily developed for numerical data and only later, extensions to categorical data became research focus. However, we pointed out crucial differences between numerical and categorical data, which complicate direct extensions of approaches for numerical data to nominal data. Specifically, common approaches for numerical data work with the Euclidean distances between data records and cluster centroids. However, we discussed that for nominal data, two data records, which are close to the same centroid with respect to the Hamming-distance, are not necessarily close to each other. Therefore, the dissimilarities between *all* data records in the same cluster need to be taken into account for nominal data. Hence, we build our approach on the recent work by Castro et al. (2022), who focus on numerical data and propose a method, which includes the overall dissimilarity. To this end, they formulate the partitioning problem in the first step of microaggregation, such that its linear relaxation is suitable for a column generation scheme. Our work builds on their idea of using column generation and the work by Zhao et al. (2018) to present a new method for k-anonymity on nominal data. Zhao et al. (2018) propose a method to derive an integer solution based on the dual information provided by a fractional solution, which is obtained by a column generation scheme. This method iteratively constructs an integer solution with the objective to minimize the duality gap. We showed the application to the partitioning step of the microaggregation procedure. In contrast to the SAFE heuristic, which is controlled by many parameters, the proposed method requires only two parameters to be determined. The first parameter $0 \leq \gamma \leq 1$ determines the weight given to the dual information during the construction of an integer solution. We found empirically that neither 0 nor 1 are recommendable, i.e. it is useful to include dual information but it should not be overemphasized. The second parameter gives weight on the cluster size. However, the size of the cluster should not outweigh the similarities of the cluster elements, i.e. a large value of $\lambda > 3$ was not recommended for our experiments. Furthermore, our experiments reveal that the better the quality of the clustering found in this partitioning step, the lower the objective function of the subsequent aggregation step, i.e. the better the quality of the resulting k-anonymous microdata set. These findings suggest that particular importance should be attached to the partitioning problem.

Moreover, we mathematically formulated the aggregation problem, which is the second step of the microaggregation procedure. The task in the aggregation step is to find cluster representatives to minimize the information loss, which results from replacing individual data records in each cluster with the respective representative record. For numerical data, typically, only the respective cluster is taken into account in the selection of its representative. However, for nominal data, frequency tables are crucial in practice and we, therefore, include a subset of frequency tables in the anonymization process. Since all cluster representatives influence the frequency table entries jointly, the selection of a cluster representative to minimize the error in frequency tables also depends on the other cluster representatives. Moreover, we decided to use the χ^2-error between table cell entries before and after anonymization in the objective function since it takes the error in frequency tables relatively to the original entry into account. Thus, it penalizes an error in a table cell with a small original entry more than the same error in a cell with a large original entry. Based on these insights, we presented a mixed-integer program minimizing the sum of χ^2-errors in the cells of control tables and proposed a greedy algorithm to approach a solution. Our experimental results show that this greedy approach can save runtime, as expected, but comes with a notable quality loss. Moreover, we reformulated this program as a minimum-weighted maximal clique problem in a multipartite graph. Even though the reformulation does not make the problem less challenging, it allows a different viewing angle. Based on this interpretation, we proposed another heuristic approach to the aggregation problem. This clique heuristic showed its strength, when the clustering, which is found in the partitioning step, has a low weight, i.e. the clusters consist of similar data rows. This representation of the problem also allows for further developments of heuristic approaches to the aggregation step involving the χ^2-error in selected frequency tables.

In both, SAFE and the proposed methods, a subset of frequency tables is selected as control tables. As shown in our experiments, the selection of only a few tables can lead to a lower information loss on these tables. Therefore, practitioners should select the control tables depending on their use case to either focus on specific frequency tables or to minimize the overall error of a larger set of frequency tables.

On the one hand, our computational experiments showed the strengths of the SAFE method with respect to maximum *absolute* errors, evaluations on one-dimensional frequency tables and a very low number of confidentiality problems in the data set. On the other hand, our proposed methods Own-hee and Own-hhe outperformed the SAFE method and SAFE-ex with respect to *relative* errors, i.e. sum of χ^2-errors and maximum relative error. In all experiments, the proposed methods Own-hee and Own-hhe outperformed Own-hhg, which is the fastest method. However, on samples of a real-world data set, also Own-hhg outperformed the SAFE method and SAFE-ex. In regard of regression analysis, the heuristic approaches SAFE and Own-hhe revealed to be competitive with the exact method Chi-ex.

Our experimental study showed that the runtime of our proposed method is dominated by the runtime of the partitioning step, which depends on the number of data rows and parameter k. We proposed accelerations of the method by heuristic approaches of the individual steps instead of exact solving. Further ideas for accelerations could be possible preprocessing steps. For example, fixing clusters of identical data rows or deleting edges between data rows that do not have any value in common. However, this should be done carefully since, depending on the microdata set, it might be beneficial to assign even identical data rows to different clusters or also to assign completely different data rows to the same cluster.

To conclude, based on the insights into the properties of nominal data, e.g. the importance of the overall dissimilarities in the clusters and of frequency tables, our proposed methods reduce the information loss when generating k-anonymous microdata sets from microdata sets with nominal data. Overall, particularly for nominal data, the inclusion of integer programming to the field of Statistical Disclosure Control is beneficial.

Appendix A

Parameter Selection for the SAFE Method

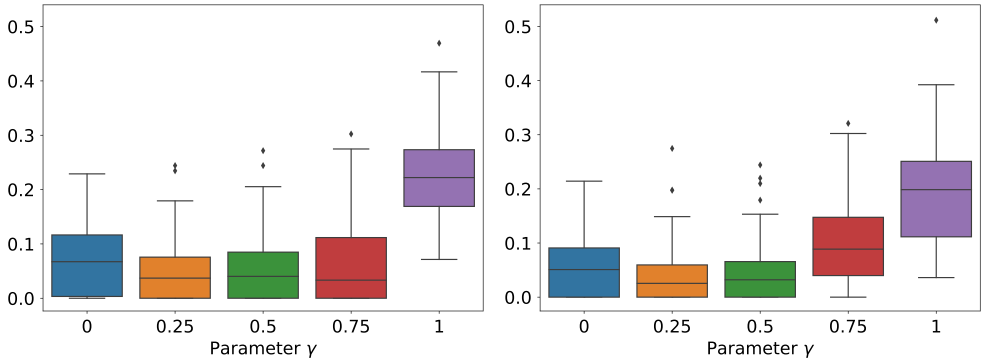
We use the following parameters for the SAFE method.

allowed_bounds_for_one_dim_table_errors = 2
allowed_bounds_for_higher_dim_table_errors = 3
max_gap_between_bounds_for_one_and_for_higher_dim_tables = 3
allowed_deviation_from_original_frequency_vector = 3
size_of_first_part_to_consider = 5000
size_of_parts = 500
size_of_slices = 3
stagnation_parameter = 0.02
target_bound_for_one_dim_tables = 2
target_bound_for_higher_dim_tables = 3
maximum_number_iterations_improvement_steps = 20

$$\text{allowed\_extra\_error\_on\_table\_cell\_t} = \begin{cases} 0, & \text{if } \bar{n}_t < 10 \\ 1, & \text{if } 10 \leq \bar{n}_t < 20 \\ 2, & \text{if } 20 \leq \bar{n}_t < 50 \\ 3, & \text{if } 50 \leq \bar{n}_t < 100 \\ 4, & \text{if } 100 \leq \bar{n}_t < 200 \\ 5, & \text{if } 200 \leq \bar{n}_t < 1{,}000 \\ 6, & \text{if } 1{,}000 \leq \bar{n}_t < 10{,}000 \\ 7, & \text{if } 10{,}000 \leq \bar{n}_t < 100{,}000 \\ 8, & \text{if } 100{,}000 \leq \bar{n}_t < 1{,}000{,}000 \\ 9, & \text{if } 1{,}000{,}000 \leq \bar{n}_t \end{cases}$$

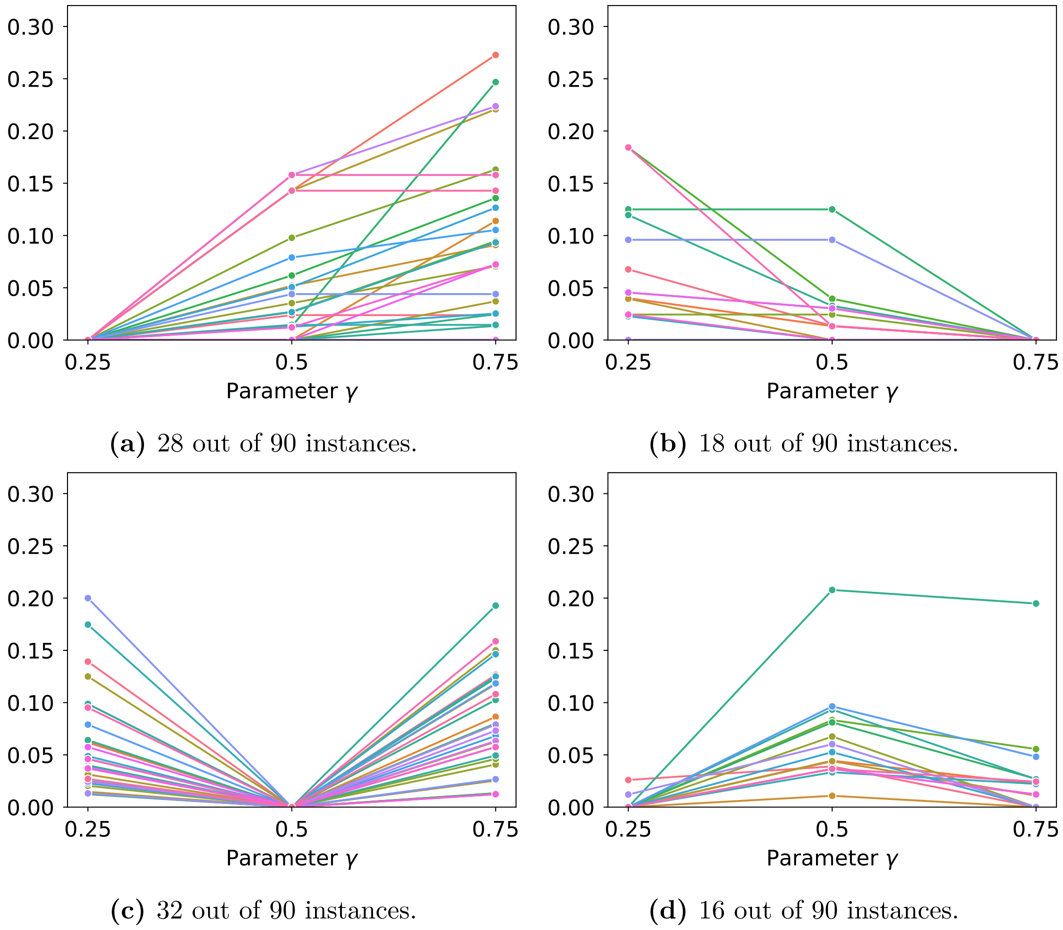
Appendix B

Further Computational Results



(a) Using an optimal fractional solution. **(b)** Using a heuristic fractional solution.

Figure B.1.: Relative gaps according to the weight of the clustering for different values of parameter γ on Scenario *adult200m5*.



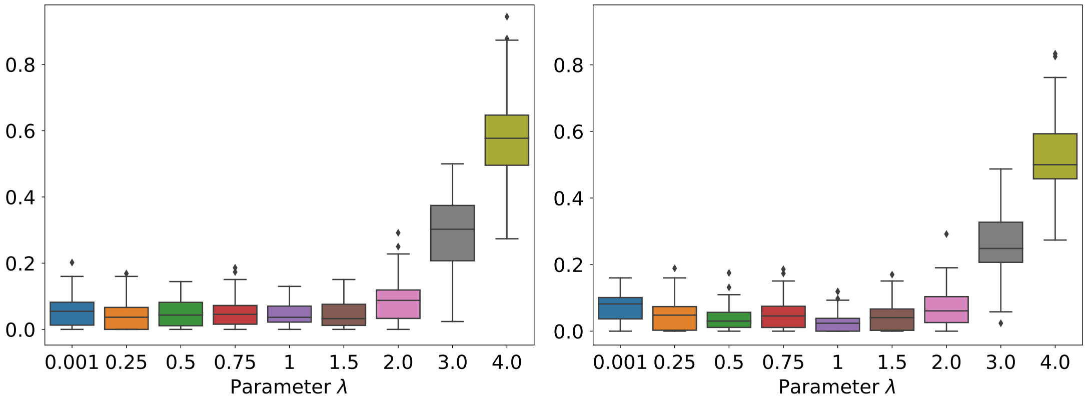
(a) 28 out of 90 instances.

(b) 18 out of 90 instances.

(c) 32 out of 90 instances.

(d) 16 out of 90 instances.

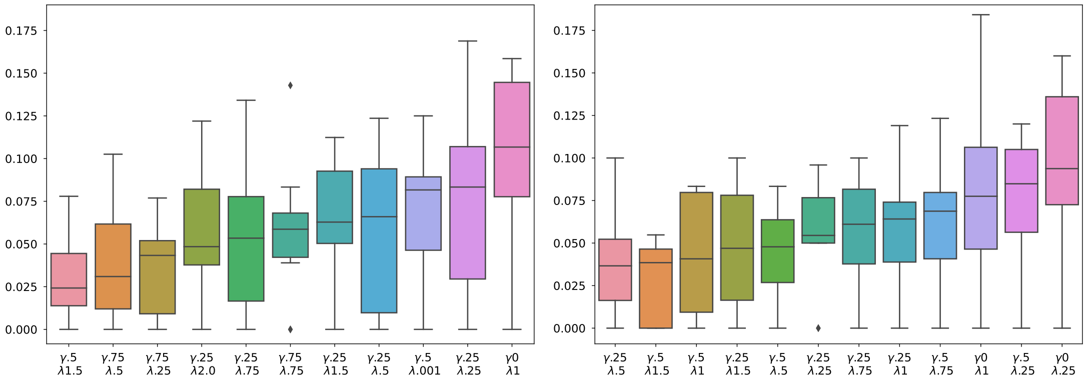
Figure B.2.: Relative gap for weight of integer clustering for selected values for γ on Scenario *200m5_20percent* given a heuristically found fractional solution.



(a) Using an optimal fractional solution.

(b) Using a heuristic fractional solution.

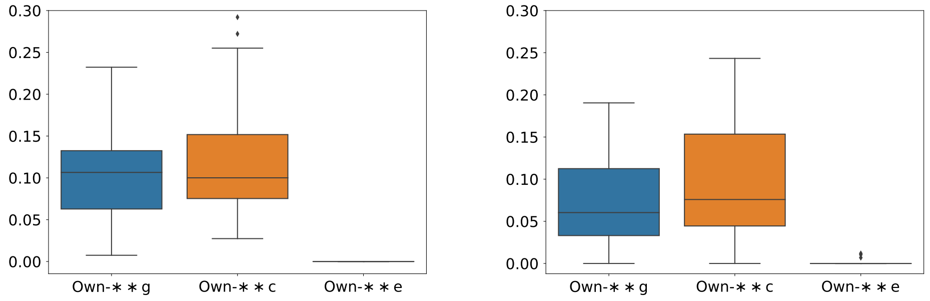
Figure B.3.: Relative gaps according to the weight of the integer clustering for different values of parameter λ on Scenario *adult200m5*.



(a) Using an optimal fractional solution.

(b) Using a heuristic fractional solution.

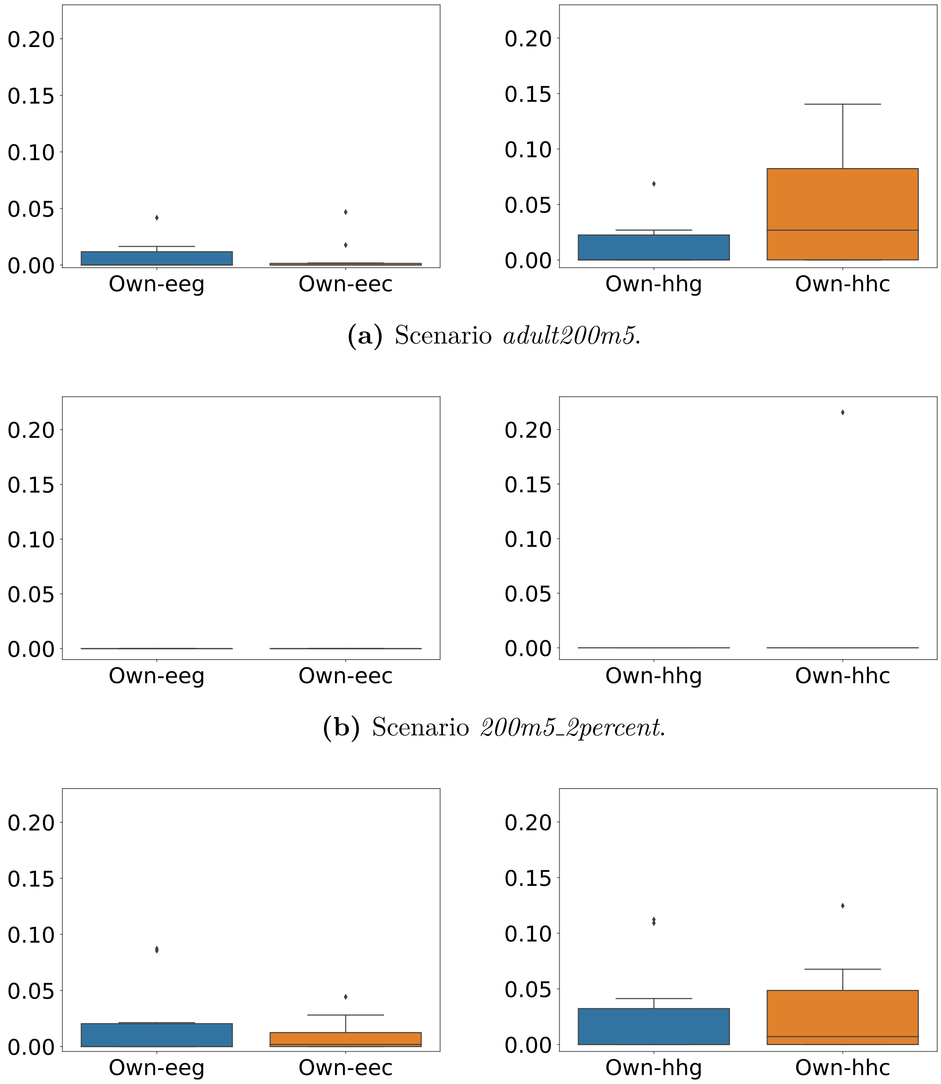
Figure B.4.: Relative gaps according to the weight of the integer clustering for different combinations of parameters λ and γ on Scenario *adult200m5*.



(a) On control tables, i.e. dimension ≤ 3.

(b) On all tables.

Figure B.5.: Relative gaps according to the sum of χ^2-errors between table cells before and after anonymization for Scenario *adult200m5*.

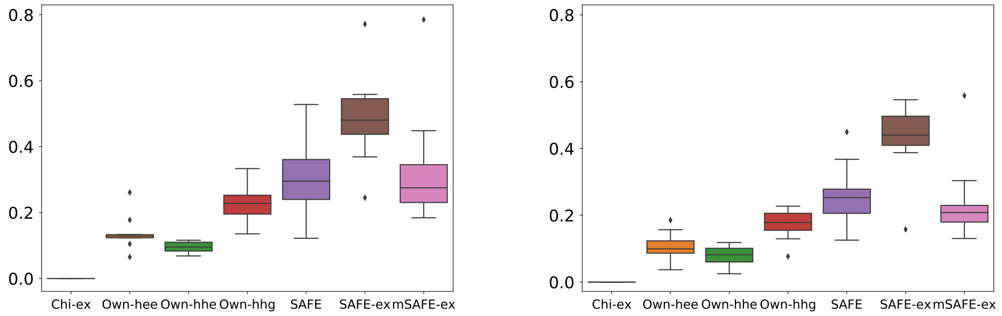


(a) Scenario *adult200m5*.

(b) Scenario *200m5_2percent*.

(c) Scenario *200m5_100percent*.

Figure B.6.: Direct comparison of heuristic aggregation methods on the control tables on different scenarios.



(a) On the control tables.

(b) On all tables.

Figure B.7.: Relative gaps according to the sum of χ^2-errors for different methods on Scenario *adult200m5*.

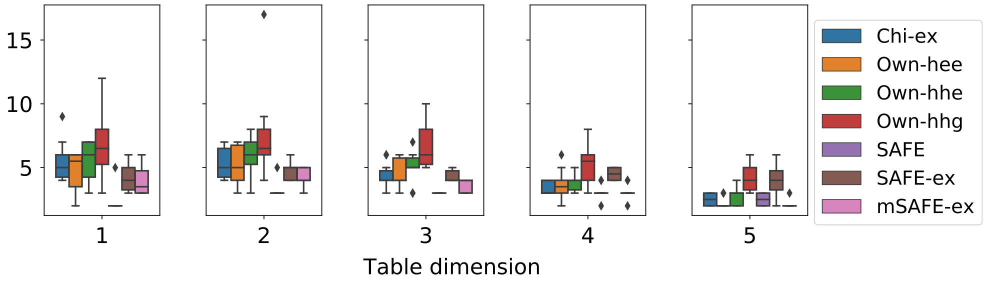


Figure B.8.: Maximum absolute error in the frequency tables per dimension on *adult200m5*.

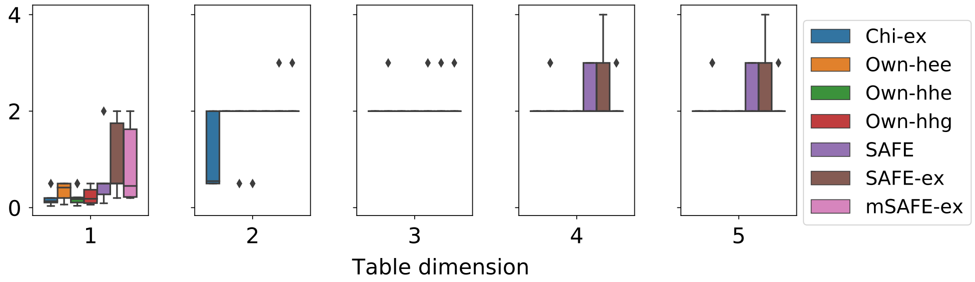
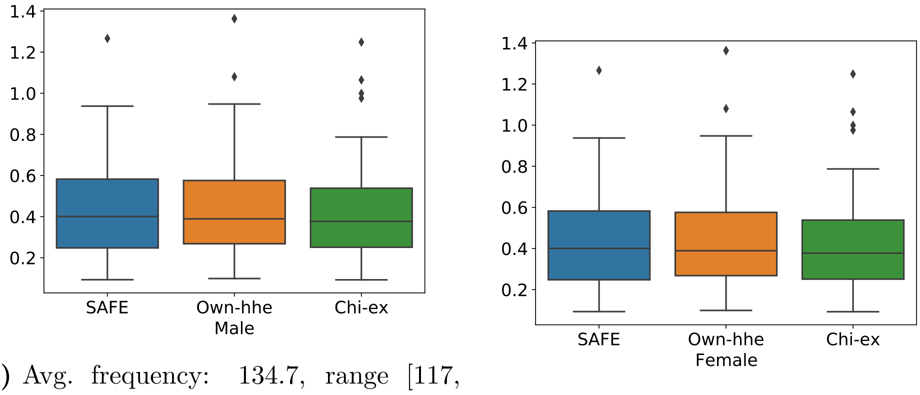


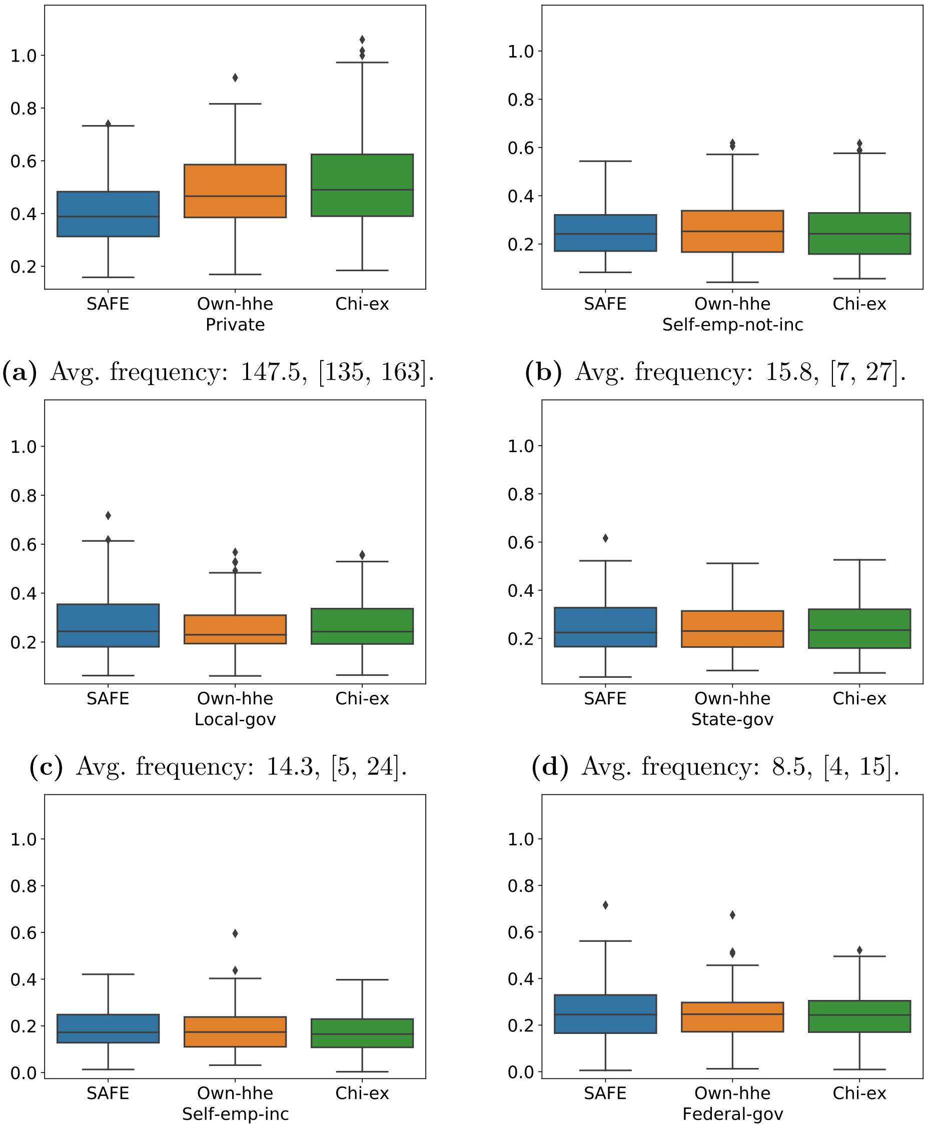
Figure B.9.: Maximum relative error in the frequency tables per dimension on *adult200m5*.



(a) Avg. frequency: 134.7, range [117, 151].

(b) Avg. frequency: 65.3, range [8, 49].

Figure B.10.: ℓ_2-distances between estimated probabilities before and after anonymization for different attribute values over 100 samples on Scenario *adult200m5* with attribute 1 as dependent variable, Pseudo-R^2 = 0.438.



(a) Avg. frequency: 147.5, [135, 163].

(b) Avg. frequency: 15.8, [7, 27].

(c) Avg. frequency: 14.3, [5, 24].

(d) Avg. frequency: 8.5, [4, 15].

(e) Avg. frequency: 7.5, [2, 14].

(f) Avg. frequency: 6.5, [2, 13].

Figure B.11.: ℓ_2-distances between estimated probabilities before and after anonymization for different attribute values over 100 samples on Scenario *adult200m5* with attribute 5 as dependent variable, Pseudo-R^2 = 0.032.

Bibliography

Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., & Zhu, A. (2005). Anonymizing tables. *International Conference on Database Theory*, 246–258.

Aghdam, M. R. S., & Sonehara, N. (2016). Achieving high data utility k-anonymization using similarity-based clustering model. *IEICE TRANSACTIONS on Information and Systems*, *99*(8), 2069–2078.

Bayardo, R. J., & Agrawal, R. (2005). Data privacy through optimal k-anonymization. *21st International conference on data engineering (ICDE'05)*, 217–228.

Brand, R. (2002). Microdata protection through noise addition. In *Inference control in statistical databases* (pp. 97–116). Springer.

Byun, J.-W., Kamra, A., Bertino, E., & Li, N. (2007). Efficient k-anonymization using clustering techniques. *International Conference on Database Systems for Advanced Applications*, 188–200.

Castro, J., Gentile, C., & Spagnolo-Arrizabalaga, E. (2022). An algorithm for the microaggregation problem using column generation. *Computers & Operations Research*, *144*, 105817.

Chen, L., & Wang, S. (2013). Central clustering of categorical data with automated feature weighting. *Twenty-Third International Joint Conference on Artificial Intelligence.*

Clifton, C., & Tassa, T. (2013). On syntactic anonymity and differential privacy. *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, 88–93.

Cox, L. H. (1995). Network models for complementary cell suppression. *Journal of the American Statistical Association*, *90*(432), 1453–1462.

Cox, L. H., & Ernst, L. (1982). Controlled rounding. *INFOR: Information Systems and Operational Research*, *20*(4), 423–432.

Dalenius, T., & Reiss, S. P. (1982). Data-swapping: A technique for disclosure control. *Journal of statistical planning and inference*, *6*(1), 73–85.

Desrosiers, J., & Lübbecke, M. E. (2005). A primer in column generation. *Column Generation*, (March), 1–32. https://doi.org/10.1007/0-387-25486-2_1

Dinur, I., & Nissim, K. (2003). Revealing information while preserving privacy. *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 202–210.

di Vimercati, S. d. C., Foresti, S., Livraga, G., & Samarati, P. (2011). Anonymization of statistical data. *IT-Information Technology*, *53*(1), 18–25.

Domingo-Ferrer, J. (2008). A survey of inference control methods for privacy-preserving data mining. In *Privacy-preserving data mining* (pp. 53–80). Springer.

Domingo-Ferrer, J., & Mateo-Sanz, J. M. (1999). Resampling for statistical confidentiality in contingency tables. *Computers & Mathematics with Applications*, *38*(11-12), 13–32.

Domingo-Ferrer, J., & Mateo-Sanz, J. M. (2002). Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and data Engineering*, *14*(1), 189–201.

Domingo-Ferrer, J., Sánchez, D., & Soria-Comas, J. (2016). Database anonymization: Privacy models, data utility, and microaggregation-based inter-model connections. *Synthesis Lectures on Information Security, Privacy, & Trust*, *8*(1), 1–136.

Domingo-Ferrer, J., & Torra, V. (2005). Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, *11*(2), 195–212.

Dua, D., & Graff, C. (2019). UCI machine learning repository. http://archive.ics.uci.edu/ml

Dwork, C. (2008). Differential privacy: A survey of results. *International conference on theory and applications of models of computation*, 1–19.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. *Theory of cryptography conference*, 265–284.

Dwork, C., & Nissim, K. (2004). Privacy-preserving datamining on vertically partitioned databases. *Annual International Cryptology Conference*, 528–544.

Feremans, C., Labbé, M., & Laporte, G. (2003). Generalized network design problems. *European Journal of Operational Research*, *148*(1), 1–13.

Fischetti, M., & Salazar, J. J. (2001). Solving the cell suppression problem on tabular data with linear constraints. *Management Science*, *47*(7), 1008–1027.

Ford Jr, L. R., & Fulkerson, D. R. (1958). A suggested computation for maximal multi-commodity network flows. *Management Science*, *5*(1), 97–101.

Fung, B. C., Wang, K., & Yu, P. S. (2005). Top-down specialization for information and privacy preservation. *21st international conference on data engineering (ICDE'05)*, 205–216.

Gamrath, G., Anderson, D., Bestuzheva, K., Chen, W.-K., Eifler, L., Gasse, M., Gemander, P., Gleixner, A., Gottwald, L., Halbig, K., Hendel, G., Hojny, C., Koch, T., Le Bodic, P., Maher, S. J., Matter, F., Miltenberger, M., Mühmer, E., Müller, B., ... Witzig, J. (2020, March). *The SCIP Optimization Suite 7.0* (Technical Report). Optimization Online. http://www.optimization-online.org/DB_HTML/2020/03/7705.html

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability* (Vol. 174). freeman San Francisco.

Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. *Journal of computer and system sciences*, *28*(2), 270–299.

Höhne, J. (2003). Safe-a method for statistical disclosure limitation of microdata. *Monographs of official statistics*, 1–3.

Höhne, J. (2010). Verfahren zur anonymisierung von einzeldaten. *Statistik und Wissenschaft Wiesbaden*, (16).

Höhne, J. (2015). Das Geheimhaltungsverfahren SAFE. *Zeitschrift für amtliche Statistik Berlin Brandenburg*, *1987*(5+6), 16–33.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, *2*(3), 283–304.

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., & De Wolf, P.-P. (2012). *Statistical disclosure control* (Vol. 2). Wiley New York.

Hurkens, C., & Tiourine, S. (1998). Models and methods for the microdata protection problem. *Journal of Official Statistics*, *14*(4), 437.

Ji, X., & Mitchell, J. E. (2007). Branch-and-price-and-cut on the clique partitioning problem with minimum clique size requirement. *Discrete Optimization*, *4*(1), 87–102. https://doi.org/10.1016/j.disopt.2006.10.009

Kelly, J. P., Golden, B. L., & Assad, A. A. (1992). Cell suppression: Disclosure protection for sensitive tabular data. *Networks*, *22*(4), 397–417.

Kooiman, P. (1997). Pram: A method for disclosure limitation of microdata. *Report, Department of Statistical Methods*.

Koster, A. M., van Hoesel, S. P., & Kolen, A. W. (1998). The partial constraint satisfaction problem: Facets and lifting theorems. *Operations research letters*, *23*(3-5), 89–97.

Larsson, T., & Patriksson, M. (2006). Global optimality conditions for discrete and nonconvex optimization - with applications to lagrangian heuristics and column generation. *Operations Research*, *54*(3), 436–453.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2005). Incognito: Efficient full-domain k-anonymity. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 49–60.

Li, N., Li, T., & Venkatasubramanian, S. (2007). T-closeness: Privacy beyond k-anonymity and l-diversity. *2007 IEEE 23rd international conference on data engineering*, 106–115.

Lloyd, S. (1982). Least squares quantization in pcm. *IEEE transactions on information theory*, *28*(2), 129–137.

Lübbecke, M. E. (2010). Column generation. *Wiley encyclopedia of operations research and management science. Wiley, New York*, 1–14.

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *1*(1), 3–es.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, *1*(14), 281–297.

Marés, J., & Torra, V. (2012). Clustering-based categorical data protection. *International Conference on Privacy in Statistical Databases*, 78–89.

Meyerson, A., & Williams, R. (2004). On the complexity of optimal k-anonymity. *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 223–228.

Moore, R. (1996). *Controlled data-swapping techniques for masking public use microdata sets*. US Census Bureau [custodian].

Nguyen, T.-H. T., Dinh, D.-T., Sriboonchitta, S., & Huynh, V.-N. (2019). A method for k-means-like clustering of categorical data. *Journal of Ambient Intelligence and Humanized Computing*, 1–11.

Samarati, P., & Sweeney, L. (1998). Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression.

San, O. M., Huynh, V.-N., & Nakamori, Y. (2004). An alternative extension of the k-means algorithm for clustering categorical data. *International journal of applied mathematics and computer science*, *14*, 241–247.

Singh, A., Yu, F., & Dunteman, G. (2004). Massc: A new data mask for limiting statistical information loss and disclosure. *Proceedings of the Joint UNECE/EUROSTAT Work Session on Statistical Data Confidentiality*, 373–394.

Solanas, A., Martinez-Balleste, A., & Domingo-Ferrer, J. (2006). V-mdav: A multivariate microaggregation with variable group size. *17th COMPSTAT Symposium of the IASC, Rome*, 917–925.

Soria-Comas, J., Domingo-Ferrer, J., & Mulero, R. (2019). Efficient near-optimal variable-size microaggregation. *International Conference on Modeling Decisions for Artificial Intelligence*, 333–345.

Sweeney, L. (2002). K-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570.

Torra, V. (2004). Microaggregation for categorical variables: A median based approach. *International Workshop on Privacy in Statistical Databases*, 162–174.

Willenborg, L., & De Waal, T. (2012). *Elements of statistical disclosure control* (Vol. 155). Springer Science & Business Media.

Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., & Fu, A. W.-C. (2006). Utility-based anonymization using local recoding. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 785–790.

Zhao, Y., Larsson, T., & Rönnberg, E. (2018). An integer programming column generation principle for heuristic search methods. *International Transactions in Operational Research*, *27*(1), 665–695. https://doi.org/10.1111/itor.12521

Zhu, T., Li, G., Zhou, W., & Philip, S. Y. (2017). Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *29*(8), 1619–1638.

Zigomitros, A., Casino, F., Solanas, A., & Patsakis, C. (2020). A survey on privacy properties for data publishing of relational data. *IEEE Access*, *8*, 51071–51099.