Raffael Bild*, Klaus A. Kuhn, and Fabian Prasser

SafePub: A Truthful Data Anonymization Algorithm With Strong Privacy Guarantees

Abstract: Methods for privacy-preserving data publishing and analysis trade off privacy risks for individuals against the quality of output data. In this article, we present a data publishing algorithm that satisfies the differential privacy model. The transformations performed are truthful, which means that the algorithm does not perturb input data or generate synthetic output data. Instead, records are randomly drawn from the input dataset and the uniqueness of their features is reduced. This also offers an intuitive notion of privacy protection. Moreover, the approach is generic, as it can be parameterized with different objective functions to optimize its output towards different applications. We show this by integrating six well-known data quality models. We present an extensive analytical and experimental evaluation and a comparison with prior work. The results show that our algorithm is the first practical implementation of the described approach and that it can be used with reasonable privacy parameters resulting in high degrees of protection. Moreover, when parameterizing the generic method with an objective function quantifying the suitability of data for building statistical classifiers, we measured prediction accuracies that compare very well with results obtained using state-of-the-art differentially private classification algorithms.

Keywords: Data privacy, differential privacy, anonymization, disclosure control, classification.

DOI 10.1515/popets-2018-0004
Received 2017-05-31; revised 2017-09-15; accepted 2017-09-16.

1 Introduction

There is a strong tension between opportunities to leverage ever-growing collections of sensitive personal data for business or research on one hand, and potential dangers to the privacy of individuals on the other. Methods for privacy-preserving data publishing and analysis aim to find a balance between these conflicting goals by trading privacy risks off against the quality of data [2].

Data published by statistical agencies usually describes a sample from a specific population. The sampling process performed during data acquisition, as well as additional random sampling sometimes performed before data is released, provides an intuitive but weak notion of privacy protection [53]. In addition, statistical data is typically sanitized using methods of *disclosure control* which includes modifying, summarizing, or perturbing (i.e. randomizing) the data. In this process, "principles-based" approaches defined by experts and rules of thumb are typically used [50].

An additional line of research, which we will call *data anonymization*, has formulated *syntactic requirements* for mitigating risks in the form of *privacy models*. The most well-known model is *k-anonymity*, which requires that each record in a dataset is indistinguishable from at least $k-1$ other records regarding attributes which could be used for re-identification attacks [52].

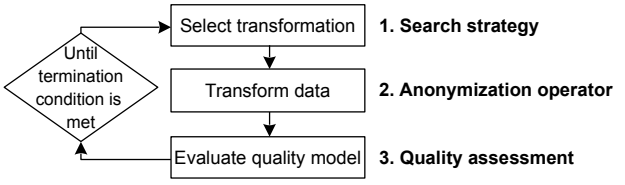


Fig. 1. Common components of data anonymization algorithms.

Based on such formal requirements, privacy protection can be implemented with *anonymization algorithms* which transform data to ensure that the requirements are met while reductions in data quality are quantified and minimized [2]. As is sketched in Figure 1, anonymization algorithms can be modelled as a process in which a set of available data transformations is being *searched*, while an anonymization operator is used to make sure that privacy requirements are *satisfied* and quality is *assessed* to guide the search process. We emphasize that this is a very high-level overview and that the design of concrete algorithms often depends on the privacy models, quality models, and, most importantly, the types of data transformation implemented.

Differential privacy [10] takes a different approach to privacy protection, as it is not a property of a dataset,

***Corresponding Author: Raffael Bild:** Technical University of Munich, Germany, E-mail: raffael.bild@tum.de
Klaus A. Kuhn: Technical University of Munich, Germany, E-mail: klaus.kuhn@tum.de
Fabian Prasser: Technical University of Munich, Germany, E-mail: fabian.prasser@tum.de

but a property of a data processing method. Informally, it guarantees that the probability of any possible output of a probabilistic algorithm (called *mechanism*) does not change "by much" if data of an individual is added to or removed from input data. Implementing differential privacy does not require making strong assumptions about the background knowledge of attackers, e.g. about which attributes could be used for re-identification. Moreover, differential privacy provides strong protection, while syntactic models are much less reliable [9].

Differential privacy, however, has also been criticized for various reasons. First, implementations are often non-truthful, i.e. perturbative, as they rely on noise addition [5, 6]. Truthfulness can be a desirable property in many fields [3]. Examples include governmental or industrial applications [21] and the medical domain, in which implausible data created by perturbation (e.g. combinations or dosages of drugs which are harmful for a patient) have led to challenges for introducing noise-based mechanisms [6]. Second, the semantics of differential privacy are complex and it has been argued that the approach is much more difficult to explain to decision makers, e.g. to ethics committees and policy makers, than the idea of *hiding in a crowd* often implemented by syntactic models [6]. Finally, differentially private mechanisms are typically special-purpose algorithms developed for specific applications, see e.g. [17, 31, 32]. Many of them serve the *interactive* scenario, i.e. they provide perturbed answers to (limited sets of) queries. In contrast, microdata publishing methods aim to release a sanitized dataset that supports a variety of use cases. The development of such *non-interactive* methods which satisfy differential privacy while retaining sufficient data quality has remained challenging.

1.1 Contributions and Outline

Previous work has shown that algorithms which draw a random sample of data followed by k-anonymization can fulfill differential privacy [26, 39, 40]. These results are notable, as they combine statistical disclosure control, data anonymization and differential privacy.

In this article, we build upon this approach to implement a traditional data anonymization algorithm (see Figure 1) with differentially private components. The result is a practical method for non-interactive microdata publishing that fulfills differential privacy. The method is truthful, as randomization is implemented via sampling only and attribute values are transformed with truthful methods. Moreover, it is intuitive, as privacy is protected by sampling records and reducing the uniqueness of their features. Finally, the approach employs a flexible search strategy which can be parameterized with a wide variety of data quality models to optimize its output towards different applications. While developing the approach, we had to overcome multiple challenges.

On the theoretical level, we have completed and extended the proofs presented in [39] and [40] to develop a method for obtaining the exact privacy guarantees obtained by the approach instead of loose upper bounds. This enabled us to strengthen a theorem about the privacy guarantees provided, to study the relationships between random sampling, k-anonymization and differential privacy in more detail and to show that the approach can be used with reasonable parameterizations providing strong privacy protection. Moreover, we have transformed six common data quality models into a form suitable for integration into the approach.

On the practical level, we have performed an extensive experimental evaluation and a comparison with related work. We have evaluated general-purpose data quality and, as an application example, performed experiments with statistical classification. Our evaluation shows that the approach is practical in terms of runtime complexity and output quality. Moreover, when our generic method is parameterized with an according data quality model, it can be used to create classifiers which are en-par with, and sometimes significantly outperform, state-of-the-art approaches. This is notable, as these competitors are pertubative special-purpose implementations of the differential privacy model.

The remainder of this paper is structured as follows: We provide background information in Section 2. Then, we give a high-level overview of the method in Section 3. The anonymization operator is presented in Section 4. Section 5 describes the objective functions. In Section 6 we introduce the search strategy. Section 7 presents analytical evaluations of the method. In Section 8 we present results of experimental analyses, including comparisons with related approaches. Section 9 reviews related work. Section 10 concludes and summarizes this article and Section 11 discusses future work.

2 Background and Preliminaries

2.1 Dataset

For an arbitrary dataset D with m attributes we will denote the domains of attribute 1 to m by $\Omega_1, ..., \Omega_m$. Then, we can regard D to be a multiset $D \subseteq \Omega_1 \times ... \times \Omega_m$, and we will denote the universe of all datasets $D \subseteq \Omega_1 \times ... \times \Omega_m$ with $\mathcal{D}_m$. Analogously to other articles we will assume that each individual who contributed data

to a dataset is represented by exactly one record $r = (r_1, ..., r_m) \in D$ and refer to such datasets as *microdata*.

2.2 Transformation Models

Data anonymization is typically performed by reducing the distinguishability of records. Common methods for doing so are clustering and aggregation of data items [28], the introduction of noise and the generalization and suppression of attribute values [2].

In this paper we focus on attribute generalization through user-specified hierarchies, which describe rules for replacing values with more general but semantically consistent values on increasing *levels* of generalization. Figure 2 shows two examples.

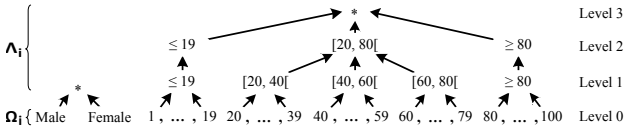


Fig. 2. Example generalization hierarchies.

Without loss of generality we will assume that a generalization hierarchy is provided for each attribute $i = 1, ..., m$ so that the values on level 0 form the domain Ω_i while we denote the set of values on levels greater than 0 by Λ_i. For a given value $r'_i \in \Omega_i \cup \Lambda_i$ we will call each value on level 0 which is an element of the subtree rooted at r'_i a *leaf node* of r'_i. For example, the leaf nodes of "$[20, 80[$" in Figure 2 are "20", ..., "79". We will indicate the removal of a record by replacing it with the placeholder $* = (*, ..., *)$. Since generalizing a value to the highest level effectively suppresses the value we will also denote the root values of all hierarchies with $*$.

2.3 Solution Spaces and Search Strategies

Most anonymization algorithms can be described as search algorithms through all possible outputs defined by the data transformation model. While they are obviously not always implemented this way (e.g. clustering algorithms typically use heuristics to guide the clustering process [28]) search algorithms are often implemented in combination with generalization hierarchies. The exact nature of the search space then depends on the generalization method.

For example, *full-domain* generalization generalizes all values of an attribute to the same level. With *subtree* generalization different values of an attribute can be generalized to different levels [2]. In this article we will focus on full-domain generalization, which results in search spaces that can be described with *generalization lattices*. An example is shown in Figure 3. An arrow denotes that a transformation is a direct *successor* of a more specialized transformation, i.e. it can be derived from its *predecessor* by incrementing the generalization level of exactly one attribute. The number of transformations in a generalization lattice grows exponentially with the number of attributes [15] and a wide variety of globally-optimal and heuristic search algorithms for generalization lattices have been proposed [15, 33–35].

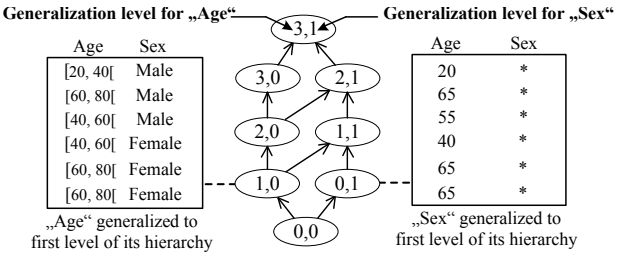


Fig. 3. Example generalization lattice and output datasets.

In this article we will use the following notion. A *generalization scheme* is a function $g : \Omega_1 \times ... \times \Omega_m \to (\Omega_1 \cup \Lambda_1) \times ... \times (\Omega_m \cup \Lambda_m)$ mapping records to (possibly) generalized records. Obviously, every transformation which performs full-domain generalization can be formalized as a generalization scheme. Unless otherwise noted, we define the solution space $\mathcal{G}_m$ to be the set of all full-domain generalization schemes which is determined by the generalization hierarchies of all attributes of a given dataset.

2.4 Anonymization Operators

An anonymization operator implements a privacy model. It determines whether or not a record or dataset satisfies the privacy requirements and may also modify the data. For example, in clustering algorithms, the anonymization operator may merge the records within a cluster into an equivalence class that satisfies k-anonymity, which we define as follows:

Definition 1 (k-Anonymity [52])**.** For a given dataset $D \subseteq (\Omega_1 \cup \Lambda_1) \times ... \times (\Omega_m \cup \Lambda_m)$, we define an *equivalence class* $E \subseteq D$ to be the multiset of all records in D which share a given combination of attribute values. An equivalence class E satisfies k-anonymity if $|E| \geq k$ holds. D satisfies k-anonymity if each record $r \in D$ cannot be distinguished from at least $k - 1$ other records, i.e. if all equivalence classes $E \subseteq D$ are k-anonymous.

As a part of algorithms implementing full-domain generalization, the anonymization operator typically sup-

presses records which do not satisfy the privacy requirements [52]. This principle can not only be implemented for k-anonymity but also for further privacy models, including l-diversity [1], t-closeness [28] and δ-presence [47], which have been proposed for protecting data from threats that go beyond re-identification.

2.5 Quality Assessment

Measuring reductions in data quality due to anonymization is non-trivial as usefulness depends on the use case.

When it is unknown in advance how the data will be used, *general-purpose* quality models can be employed. They typically estimate data quality by quantifying the amount of information loss, e.g. by measuring similarities between the input and the output dataset [2]. Models can roughly be classified as measuring information loss on the attribute-level, cell-level or record-level. Typical examples for changes on these levels are differences in the distributions of attribute values (attribute-level), reductions in the granularity of data (cell-level) and differences in the sizes of equivalence classes (record-level).

Special-purpose (or *workload-aware*) quality models quantify data quality for a specific application scenario, e.g. statistical classification. Thereby the task is to predict the value of a predefined *class attribute* from a given set of values of *feature attributes*. This is implemented with *supervised learning* where a model is created from a *training set* [54]. Specific quality models have been developed for optimizing data for this purpose [25, 37].

2.6 Differential Privacy

Differential privacy requires that any output of a mechanism is almost as likely, independent of whether or not a record is present in the input dataset [10]. (ϵ, δ)-*Differential privacy* can be formally defined with respect to two datasets D_1 and D_2 satisfying $|D_1 \oplus D_2| = 1$, which means that D_2 can be obtained from D_1 by either adding or removing one record:

Definition 2 ((ϵ, δ)-differential privacy [6])**.** A randomized function $\mathcal{K}$ provides (ϵ, δ)-differential privacy if for all datasets $D_1, D_2 \in \mathcal{D}_m$ with $|D_1 \oplus D_2| = 1$, and all measurable $S \subseteq \mathit{Range}(\mathcal{K})$,

$$P[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \cdot P[\mathcal{K}(D_2) \in S] \qquad (1)$$

holds with a probability of at least $1 - \delta$.

$(\epsilon, 0)$-Differential privacy is usually just called ϵ-differential privacy. For $\delta > 0$, (ϵ, δ)-differential privacy is then a relaxation of ϵ-differential privacy.

Sequences of differentially private computations are also differentially private:

Theorem 1. *For $i = 1, ..., n$, let the mechanism $\mathcal{M}_i$ provide ϵ_i-differential privacy. Then the sequence $\mathcal{M}_1^{r_1}(D), ..., \mathcal{M}_n^{r_n}(D)$, where $\mathcal{M}_i^{r_i}$ denotes mechnism $\mathcal{M}_i$ supplied with the outcomes of $\mathcal{M}_1, ..., \mathcal{M}_{i-1}$, satisfies $\left(\sum_{i=1}^{n} \epsilon_i\right)$-differential privacy [45].*

A common method to achieve differentially privacy is the *exponential mechanism* [44]. It ranks all potential outputs $r \in \mathcal{R}$ for a given input dataset D using a real-valued *score function* s. It then randomly chooses one according to a specific probability distribution which assigns higher probabilities to outputs with higher scores:

Definition 3 (Exponential mechanism [44])**.** For any function $s : (\mathcal{D}_m \times \mathcal{R}) \to \mathbb{R}$, the *exponential mechanism* $\mathcal{E}_s^{\epsilon}(D, \mathcal{R})$ chooses and outputs an element $r \in \mathcal{R}$ with probability proportional to $\exp\left(\frac{s(D,r)\epsilon}{2\Delta s}\right)$, where the *sensitivity* Δs of the function s is defined as

$$\Delta s := \max_{r \in \mathcal{R}} \max_{D_1, D_2 \in \mathcal{D}_m : |D_1 \oplus D_2| = 1} |s(D_1, r) - s(D_2, r)|.$$

It can be seen that it is important to use score functions which assign higher scores to outputs with higher quality while having a low sensitivity. The privacy guarantees provided are as follows:

Theorem 2. *For any function $s : (\mathcal{D}_m \times \mathcal{R}) \to \mathbb{R}$, $\mathcal{E}_s^{\epsilon}(D, \mathcal{R})$ satisfies ϵ-differential privacy [44].*

3 Overview of the Approach

Prior work has shown that randomization via sampling can be used to achieve (ϵ, δ)-differentially privacy [26, 39, 40]. We build upon and extend these results to implement the *SafePub* algorithm. It comprises a search strategy, an anonymization operator and various methods for quality assessment, similar to many anonymization algorithms. The overall privacy budget ϵ is split up into two parts ϵ_{anon}, which is used by the anonymization operator, and ϵ_{search}, which is used by the search strategy. SafePub satisfies $(\epsilon_{anon} + \epsilon_{search}, \delta)$-differential privacy, where δ and the number of iterations performed by the search strategy (*steps*) can also be specified.

Figure 4 shows the high-level design of the approach. It also indicates the parameters which are relevant for the individual steps. First, SafePub performs pre-processing by random sampling, selecting each

Input: Dataset D, Parameters ϵ_{anon}, ϵ_{search}, δ, *steps*
Output: Dataset S

1: Draw a random sample D_s from D ▷ (ϵ_{anon})
2: Initialize set of transformations G
3: **for** (Int $i \leftarrow 1, ..., steps$) **do**
4: Update G
5: **for** $(g \in G)$ **do**
6: Anonymize D_s using g ▷ $(\epsilon_{anon}, \delta)$
7: Assess quality of resulting data
8: **end for**
9: Probabilistically select solution $g \in G$ ▷ (ϵ_{search})
10: **end for**
11: **return** Dataset D_s anonymized using the best solution selected in Line 9 ▷ $(\epsilon_{anon}, \delta)$

Fig. 4. High-level design of the SafePub mechanism. The search strategy is implemented by the loop in lines 3 to 10.

record independently with probability $\beta = 1 - e^{-\epsilon_{anon}}$. This leads to provable privacy guarantees as we will see in the next section. Then, a search through the space of all full-domain generalization schemes is performed. It comprises multiple iterations which are implemented by the for-loop in lines 3 to 10. In each iteration the sample is anonymized using every full-domain generalization scheme in the set G. The quality of the resulting data is assessed and a good solution is selected in a probabilistic manner. Finally, the mechanism returns the best transformation which has been selected during all iterations. In the following sections we will describe each component in greater detail.

4 Anonymization Operator

An overview of the anonymization operator is shown in Figure 5. It builds upon prior work by Li et al. [39] which we have extended with a parameter calculation so that the operator satisfies (ϵ, δ)-differential privacy for arbitrary user-specified parameters. The operator first generalizes the (sampled) input dataset using the generalization scheme g, and then suppresses every record which appears less than k times. Thereby the integer k is derived from the privacy parameters δ and ϵ_{anon}. We will simply denote ϵ_{anon} with ϵ in this section.

Every output of the operator obviously satisfies k-anonymity. Moreover, Li et al. have shown that:

Theorem 3. *Random sampling with probability β followed by attribute generalization and the suppression of every record which appears less than k times satisfies*

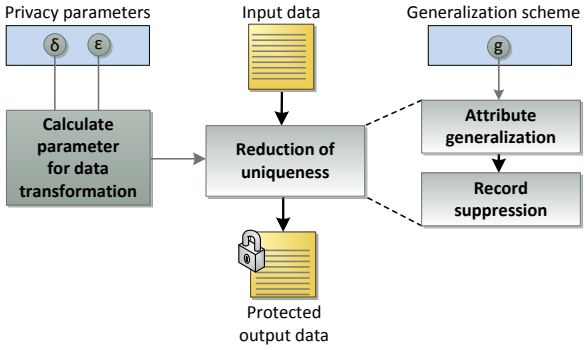


Fig. 5. Overview of the anonymization operator.

(ϵ, δ)-differential privacy for every $\epsilon \geq -\ln(1-\beta)$ with

$$\delta = d(k, \beta, \epsilon) \coloneqq \max_{n:n \geq n_m} \sum_{j > \gamma n}^{n} f(j; n, \beta) \tag{2}$$

where $n_m \coloneqq \left\lceil \frac{k}{\gamma} - 1 \right\rceil$, $\gamma \coloneqq \frac{e^{\epsilon} - 1 + \beta}{e^{\epsilon}}$ and $f(j; n, \beta) \coloneqq \binom{n}{j} \beta^j (1-\beta)^{n-j}$, which is the probability mass function of the binomial distribution [39].

It can be seen that the calculation of β described in Section 3 follows from Theorem 3:

$$\epsilon \geq -\ln(1-\beta) \Rightarrow \beta \leq 1 - e^{-\epsilon} \coloneqq \beta_{max}$$

We will explain why we set $\beta = \beta_{max}$ in Section 8.2.

To derive a practical anonymization operator from Theorem 3, it is necessary to calculate a value for k from given values of ϵ, δ and β. This is not trivial since Equation (2) requires to find the maximum of an infinite non-monotonic sequence. In the following we will show how this is implemented in SafePub. To do so, we will first introduce some definitions for notational convenience and recapitulate some important prior results.

For ease of notation we define the sequence:

$$a_n \coloneqq \sum_{j > \gamma n}^{n} f(j; n, \beta). \tag{3}$$

It follows that $d(k, \beta, \epsilon) = \max_{n:n \geq n_m} a_n$. Furthermore, we will use the following sequence:

$$c_n \coloneqq e^{-n(\gamma \ln(\frac{\gamma}{\beta}) - (\gamma - \beta))}. \tag{4}$$

Li et al. have shown in [40] that c_n is strictly monotonically decreasing with $\lim_{n \to \infty} c_n = 0$ and that it is an upper bound for a_n, i.e. it satisfies:

$$\forall n \in \mathbb{N} : a_n \leq c_n. \tag{5}$$

From these results we can conclude:

$$\delta = d(k, \beta, \epsilon) = \max_{n:n \geq n_m} a_n \underset{(5)}{\leq} \max_{n:n \geq n_m} c_n \leq c_{n_m}. \tag{6}$$

The sequence a_n consists of sums which are, except for multiplicative factors, partial sums of a row in Pascal's triangle. For such sums no closed-form expressions are known [23]. However, we will show that the function d can still be evaluated by using the following simplified representation:

Theorem 4. *The function d has the representation*

$$d(k,\beta,\epsilon) = \max\{a_{n_m},...,a_{\tilde{n}}\}$$

where $\tilde{n} \coloneqq \min\{N \geq n_m : c_N \leq a_{n_m}\}$.

Proof. From $\lim_{n\to\infty} c_n = 0$ and $a_{n_m} > 0$ we can conclude:

$$\begin{aligned}&\forall \xi > 0\ \exists N \geq n_m\ \forall n \geq N:\ c_n \leq \xi\\ \Rightarrow\ &\exists N \geq n_m\ \forall n \geq N:\ c_n \leq a_{n_m}\\ \Rightarrow\ &\exists N \geq n_m:\ c_N \leq a_{n_m}.\end{aligned}$$

Hence $\tilde{n}$ exists. Since the sequence c_n is monotonically decreasing with increasing n it follows that:

$$\forall n > \tilde{n}:\ a_n \underset{(5)}{\leq} c_n \leq c_{\tilde{n}} \leq a_{n_m} \leq \max\{a_{n_m},...,a_{\tilde{n}}\}.$$

We can conclude:

$$\max\{a_{n_m},...,a_{\tilde{n}}\} = \max_{n:n\geq n_m} a_n = d(k,\beta,\epsilon)\ . \qquad \square$$

Theorem 4 allows to derive δ from k, β and ϵ by calculating both a_n and c_n for increasing values of $n \geq n_m$ until an index $\tilde{n}$ satisfying $c_{\tilde{n}} \leq a_{n_m}$ is reached. δ is then the maximum of the finite sequence $a_{n_m},...,a_{\tilde{n}}$. This strategy is schematically illustrated in Figure 6.

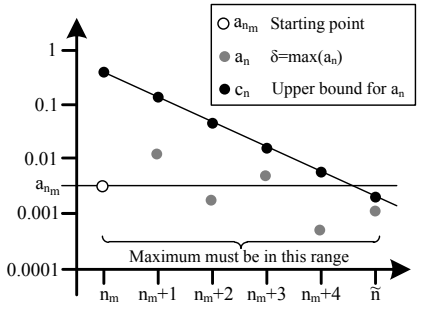


Fig. 6. Schematic plot of a_n and c_n in the range n_m to $\tilde{n}$.

For fixed values of β and ϵ we obtain the function $d(\cdot,\beta,\epsilon) : \mathbb{N} \to [0,1]$. In order to use this function to compute a value of k so that (ϵ,δ)-differential privacy is provably satisfied, we will first prove that $d(\cdot,\beta,\epsilon)$ converges:

Theorem 5. *For arbitrary $\epsilon > 0$ and $0 < \beta < 1$,* $\lim_{k\to\infty} d(k,\beta,\epsilon) = 0$ *is satisfied.*

Proof. Note that n_m is a function of k which satisfies:

$$n_m = n_m(k) = \left\lceil \frac{k}{\gamma} - 1 \right\rceil \to \infty,\ k \to \infty\ .$$

Using the strict monotonicity of c_n we can conclude:

$$\begin{aligned}0 \leq\ & d(k,\beta,\epsilon) = \max\left\{a_{n_{m(k)}},...,a_{\tilde{n}}\right\}\\ &\underset{(5)}{\leq} \max\left\{c_{n_{m(k)}},...,c_{\tilde{n}}\right\} = c_{n_{m(k)}} \to 0,\ k \to \infty.\end{aligned}$$

The claim follows according to the squeeze theorem. $\square$

From this result we can conclude:

$$\forall \delta > 0\ \exists k \in \mathbb{N} : d(k,\beta,\epsilon) \leq \delta.$$

In order to find the smallest such k for a given value of δ, we can evaluate $d(k,\beta,\epsilon)$ as described above for increasing values of $k \in \mathbb{N}$ until $d(k,\beta,\epsilon) \leq \delta$ is satisfied. More formally, k can be computed using the function:

$$d'(\delta,\beta,\epsilon) \coloneqq \min\{k \in \mathbb{N} : d(k,\beta,\epsilon) \leq \delta\}.$$

We denote the output of the operator with $S(D) \coloneqq$ *suppress*$(g(D),k)$, where $g(D) \coloneqq \bigcup_{r\in D}\{g(r)\}$ and *suppress* denotes a function that suppresses every record which appears less than k times.

5 Quality Assessment

The output of the anonymization operator must be assessed to determine a good solution. For this purpose the search strategy employs the exponential mechanism. In this section we will present implementations of common quality models as score functions and discuss their sensitivities. They comprise five general-purpose models which are frequently used in the literature [28, 56] and which have been recommended in current data de-identification guidelines [11] as well as a special-purpose model for building statistical classifiers. For proofs we refer to Appendix B.

5.1 Granularity and Intensity

Data Granularity is a cell-level, general-purpose model. It measures the extent to which the values in a dataset cover the domains of the respective attributes [25]. Since the model already has a low sensitivity, we can multiply its results with -1 to obtain a score function which measures data quality rather than information loss:

Definition 4. For $i = 1,...,m$, let $leaves_i : \Omega_i \cup \Lambda_i \to \mathbb{N}$ denote the function which returns the number of leaf

nodes for each value r'_i within the generalization hierarchy of the i-th attribute. For every $k \in \mathbb{N}$, we define the score function $gran_k : (\mathcal{D}_m \times \mathcal{G}_m) \to \mathbb{R}$ as follows:

$$gran_k(D, g) \coloneqq - \sum_{(r'_1,\ldots,r'_m)\in S(D)} \sum_{i=1}^{m} \frac{leaves_i(r'_i)}{|\Omega_i|}.$$

The sensitivity of $gran_k$ is as follows (see Appendix B.1):

Theorem 6. *For every $k \in \mathbb{N}$, the following holds:*

$$\Delta gran_k \leq \begin{cases} (k-1)m, & \text{if } k > 1 \\ m, & \text{if } k = 1 \end{cases}.$$

Generalization Intensity is another cell-level, general-purpose model which sums up the relative generalization level of values in all cells [52]. A score function $intensity_k : (\mathcal{D}_m \times \mathcal{G}_m) \to \mathbb{R}$ which is tailored to this model, and which has the same sensitivity as $gran_k$, can be constructed analogously.

5.2 Discernibility

Discernibility is a record-level, general-purpose model which penalizes records depending on the size of the equivalence class they belong to [3]. Let $EQ(D)$ denote the set of all equivalence classes of D, except of $\{* \in D\}$, which contains the suppressed records in D. We first define the following normalized variant of the model:

$$\phi(D) \coloneqq \left(\sum_{E\in EQ(D)} \frac{|E|^2}{|D|} \right) + |\{* \in D\}|. \tag{7}$$

We note that suppressed records are considered separately from the other records in Equation (7) as this improves the sensitivity of the function. The score function $disc_k : (\mathcal{D}_m \times \mathcal{G}_m) \to \mathbb{R}$ is defined as follows:

Definition 5. $disc_k(D, g) \coloneqq -\phi(S(D))$.

The sensitivity of $disc_k$ is as follows (see Appendix B.2):

Theorem 7. *For every $k \in \mathbb{N}$, the following holds:*

$$\forall k \in \mathbb{N} : \Delta disc_k \leq \begin{cases} 5, & \text{if } k = 1 \\ \frac{k^2}{k-1} + 1, & \text{if } k > 1 \end{cases}.$$

5.3 Non-Uniform Entropy

Non-Uniform Entropy is an attribute-level, general-purpose model which quantifies the amount of information that can be obtained about the input dataset by observing the output dataset [7]. According to this model information loss increases with increasing homogeneity of attribute values in the output dataset. Hence we will base the score function on a measure of homogeneity.

Let $p_i(D)$ denote the projection of D to its i-th attribute. We can then measure the homogeneity of attribute values in D using the function ϕ (see Equation (7)) by calculating $\sum_{i=1}^{m} \phi(p_i(D))$ and thus define:

Definition 6. For every $k \in \mathbb{N}$, the score function $ent_k : (\mathcal{D}_m \times \mathcal{G}_m) \to \mathbb{R}$ is defined as:

$$ent_k(D, g) \coloneqq - \sum_{i=1}^{m} \phi(p_i(S(D))).$$

The sensitivity of ent_k is as follows (see Appendix B.3):

Theorem 8. *For every $k \in \mathbb{N}$, we have*

$$\Delta ent_k \leq \begin{cases} 5m, & \text{if } k = 1 \\ (\frac{k^2}{k-1} + 1)m, & \text{if } k > 1 \end{cases}.$$

5.4 Group Size

Group Size is a record-level, general-purpose model which measures the average size of equivalence classes [36]. We derive a score function which is inversely correlated to this model as follows:

Definition 7. For every $k \in \mathbb{N}$, the score function $groups_k : (\mathcal{D}_m \times \mathcal{G}_m) \to \mathbb{R}$ is defined as:

$$groups_k(D, g) \coloneqq |EQS(D)|.$$

Since the addition of a single record can lead to at most one additional equivalence class, it is easy to see that $\forall k \in \mathbb{N} : \Delta groups_k \leq 1$ holds.

5.5 Statistical Classification

Iyengar has proposed a special-purpose model which measures the suitability of data as a training set for statistical classifiers [25]. It penalizes records which do not contain the most frequent combination of feature and class attribute values. Since the model already has a low sensitivity, we can derive a practical score function by giving weights to records which are not penalized:

Definition 8. For every $k \in \mathbb{N}$, the score function $class_k : (\mathcal{D}_m \times \mathcal{G}_m) \to \mathbb{R}$ is defined as follows:

$$class_k(D, g) \coloneqq \sum_{r'\in S(D)} w(S(D), r') \,.$$

Let $fv(r')$ denote the the sub-vector of a record r' which consists of the feature attribute values in r'. The record

r' is given a weight by the function w if $fv(r')$ is not suppressed and if the class attribute value $cv(r')$ of r' is equal to the most frequent class value $cv_{maj}(S(D), r')$ among all records in $S(D)$ which share the same combination of feature values. More precisely, we define:

$$w(S(D), r') := \begin{cases} 1, & \text{if } fv(r') \text{ is not suppressed and} \\ & cv(r') = cv_{maj}(S(D), r') \text{ holds} \\ 0, & \text{otherwise} \end{cases}.$$

The sensitivity of $class_k$ is as follows (see Appendix B.4):

Theorem 9. *For every $k \in \mathbb{N}$, $\Delta class_k \leq k$ holds.*

6 Search Strategy

The search strategy implements a (randomized) top-down search through the generalization lattice using the scores which are calculated according to the given quality model. Traversal is implemented by iterative applications of the exponential mechanism which exponentially favors transformations with high scores, and thus likely returns transformations resulting in good output data quality (see Section 2.6). For ease of notation we will denote ϵ_{search} with ϵ in this section.

Input: Dataset $D \in \mathcal{D}_m$, Real ϵ, Integer *steps*,
1: ScoreFunction $s : (\mathcal{D}_m \times \mathcal{G}_m) \to \mathbb{R}$
Output: Scheme $g \in \mathcal{G}_m$
2: Real $\tilde{\epsilon} \leftarrow \epsilon / steps$
3: Scheme $pivot \leftarrow \top$
4: Scheme $optimum \leftarrow \top$
5: SchemeSet $candidates \leftarrow \{\top\}$
6: **for** (Int $i \leftarrow 1, ..., steps$) **do**
7: $\quad candidates \leftarrow candidates \cup predecessors(pivot)$
8: $\quad candidates \leftarrow candidates \setminus \{pivot\}$
9: $\quad pivot \leftarrow \mathcal{E}_s^{\tilde{\epsilon}}(D, candidates)$
10: $\quad$ **if** $(s(D, pivot) > s(D, optimum))$ **then**
11: $\quad\quad optimum \leftarrow pivot$
12: $\quad$ **end if**
13: **end for**
14: **return** *optimum*

Fig. 7. Detailed presentation of the search strategy.

Figure 7 shows a more detailed presentation of the search strategy which is also outlined in the high-level overview in Figure 4 (the loop in lines 6 to 13 of Figure 7 corresponds to the loop in lines 3 to 10 of Figure 4). The function *predecessors* maps a transformation to the set of its direct predecessors. The search starts with the transformation $\top \in \mathcal{G}_m$ which generalizes every attribute to the highest level available. The scores of all direct predecessors of $\top$ are calculated and the transformations are put into the set *candidates*. In each iteration a pivot element is selected from the set using the exponential mechanism with a privacy budget of $\tilde{\epsilon} = \epsilon / steps$, the scores of all its direct predecessors are calculated, and the predecessors are being added to *candidates*. The pivot element is then removed from the set. After a predefined number of steps the method returns the pivot element with the best score.

We note that using $steps = 0$ is possible but impractical, as this results in the deterministic selection of the transformation $\top$ that suppresses all data.

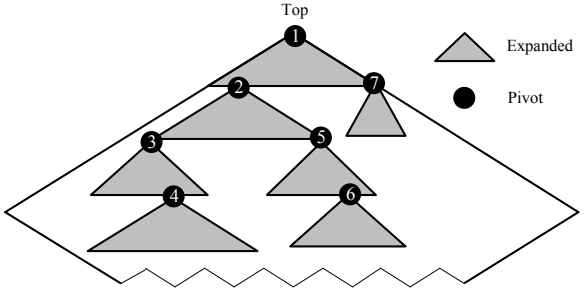


Fig. 8. Schematic illustration of the search strategy.

Figure 8 schematically illustrates the method. A black circle represents a pivot element and the gray triangle below it represents its direct predecessors. The method is likely to perform a best-first search, following a path of transformations with increasing score values (e.g. the path from transformation no. 1 to no. 4). We note that it is not likely that the algorithm will be trapped in a local minimum, i.e. that it continues following a path of elements with non-optimal score values. The reason is that the predecessors of all previously selected pivot elements are left in the set *candidates*. For example, if all predecessors of pivot element no. 4 have a lower score than transformation no. 5, then transformation no. 5 will likely be selected as the next pivot element. Moreover, following a non-optimal path is unlikely to negatively affect the quality of the overall output, as the final solution is selected deterministically. The privacy guarantees provided are as follows:

Theorem 10. *For every parameter $steps \in \mathbb{N}_0$ and $\epsilon > 0$, the search strategy satisfies ϵ-differential privacy.*

Proof. If $steps = 0$ holds the search strategy returns $\top$ in a deterministic manner and hence trivially satisfies ϵ-differential privacy. In the following we will assume that $steps > 0$ holds. We note that the only instructions which modify the content of the variables *pivot* and *candidates* are located in lines seven to nine.

For every iteration $i \in \{1, ..., steps\}$ of the enclosing loop, let $\mathcal{M}_i^{r_i}(D)$ denote the sequence of operations performed by these three lines during the i-th iteration. Let $r_i = (pivot_i, candidates_i)$ denote the content of the variables *pivot* and *candidates* before the i-th iteration of the loop. Then each r_i is determined by $\mathcal{M}_1^{r_1}(D), ..., \mathcal{M}_{i-1}^{r_{i-1}}(D)$ and supplied to $\mathcal{M}_i^{r_i}(D)$ which outputs r_{i+1} in a manner that satisfies $\tilde{\epsilon}$-differential privacy according to Theorem 2. We can conclude from Theorem 1 that the sequence $\mathcal{M}_1^{r_1}(D)$, ..., $\mathcal{M}_{steps}^{r_{steps}}(D)$ satisfies ϵ-differential privacy since $\sum_{i=1}^{steps} \tilde{\epsilon} = \epsilon$ holds. Finally, the algorithm returns the generalization scheme with the highest score value amongst all pivot elements selected by the differentially private operations $\mathcal{M}_1^{r_1}(D)$, ..., $\mathcal{M}_{steps}^{r_{steps}}(D)$ in a deterministic manner. Hence the algorithm satisfies ϵ-differential privacy. □

7 Analytical Evaluation

7.1 Complexity Analysis

Let $n = |D|$ denote the number of records, each consisting of m attributes. Each basic operation, i.e. drawing a random sample, executing the anonymization operator and evaluating a score function, has a runtime complexity of $O(n \cdot m)$. In each step of the search process, the anonymization operator and the score function are being evaluated once for at most m predecessors of the current pivot element. Hence each step has a time complexity of $O(n \cdot m^2)$. The number of steps performed is a user-defined parameter and we will derive recommendations experimentally in Section 8.

We note that the method for calculating the parameters of the algorithm described in Section 4 is of non-trivial runtime complexity. Unfortunately a detailed analysis is complex and out of the scope of this work. We have, however, performed experimental evaluations using a wide variety of common parameterizations which showed that the approach is practical. We will present the results in the next section.

7.2 Parameter Analysis

In this section we analyze dependencies between parameters of SafePub. We will focus on ϵ_{anon} and δ since they determine k and β in a non-trivial manner. For ease of of notation, we will denote ϵ_{anon} with ϵ.

Figure 9 shows the values of β and k obtained for various values of ϵ and δ as described in Section 4. We focus on common values of ϵ [6]. Later we will set δ to 10^{-m} with $m \in \mathbb{N}$ such that $\delta < 1/n$, where n is the size of the dataset, and at least $\delta \leq 10^{-4}$ holds. This is a recommended parameterization [38, 40]. We focus on ranges of δ relevant to our evaluation datasets (see Section 8.1).

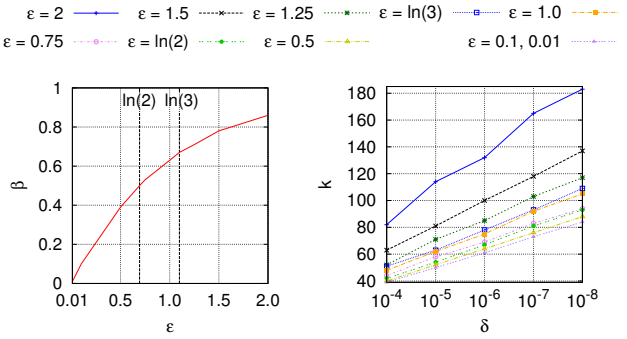


Fig. 9. Overview of values for k and β derived from ϵ and δ.

As can be seen, for fixed values of ϵ, decreasing δ increases k and thus potentially reduces data quality. Decreasing ϵ, however, has two consequences with possibly opposing impacts: On one hand, β decreases, but on the other hand, for fixed values of δ, k decreases as well. The value of β decreases rapidly for smaller values of ϵ which indicates that our approach is not practical with such parameterizations. When ϵ increases, the increase of β flattens, while k increases further.

We also measured the time required to calculate β_{max} and k for every ϵ discussed here and $10^{-4} \leq \delta \leq 10^{-20}$ on a desktop PC with a quad-core 3.1 GHz Intel Core i5 CPU. We measured between 0.1s and 37s with an average of 4.5s. This shows that the method presented in Section 4 terminates quickly for realistic privacy parameters.

7.3 Smooth Privacy

While the (ϵ, δ)-differential privacy model guarantees that the bound $\exp(\epsilon)$ in Inequation (1) may be exceeded with a probability of at most δ, it does not restrict the permitted degree of exceedance.

Li et al. have suggested that the mechanism studied here has the property that the higher such an exceedance is, the more unlikely it is to occur [40]. However, their results just provide upper bounds for these probabilities based on Inequation (6) which are very conservative: For example, for the values $\epsilon = 1$, $\beta = 0.632$ and $k = 75$, they overestimate δ by more than four orders of magnitude ($3.7 \cdot 10^{-2}$ vs. 10^{-6}). Based on our results, we can calculate the exact probabilities:

Theorem 11 (Smoothness property)**.** *For arbitrary parameters $\epsilon = \epsilon_{anon} > 0$ and $\delta > 0$, let β and k be the parameters derived as described in Section 4. Then the*

combination of random sampling with probability β and the anonymization operator satisfies $(\epsilon', d(k, \beta, \epsilon'))$-differential privacy simultaneously for every $\epsilon' \geq \epsilon$ while $d(k, \beta, \epsilon')$ is monotonically decreasing when ϵ' increases.

The proof can be found in Appendix A.

Figure 10 illustrates the smoothness property for $\epsilon = 1$ and various values of δ. As can be seen, the probability of exceeding ϵ decreases exponentially for increasing degrees of exceedance. The smaller δ, the steeper are the curves, which means that the smoothness effect is stronger. Hence, when δ is set based on the size of the dataset as described in Section 7.2, the degree of protection increases with increasing size of the dataset.

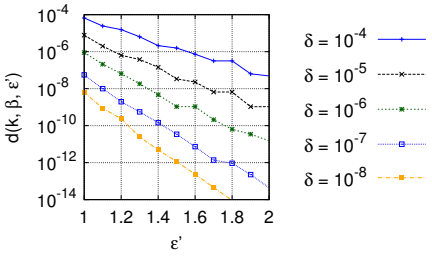


Fig. 10. Semi-log plot showing the smoothness property for $\epsilon = 1$ and various values of δ.

8 Experimental Evaluation

We have implemented our method using the open source ARX Data Anonymization Tool[1]. In this section we present experimental analyses of each individual component of SafePub and develop recommendations for parameterizations. Furthermore, we present results of comparisons with related methods.

8.1 Datasets and Setup

We used four different datasets (see [48]) in our experiments: *1) US Census (USC)*, an excerpt of records from the 1994 U.S. Census database which is often used for evaluating anonymization algorithms, *2) Crash statistics (CS)*, a database about fatal traffic accidents, *3) Time use survey (TUS)*, a dataset consisting of responses to a survey on individual time use in the U.S. and *4) Health interviews (HI)*, a database of records from a survey on the health of the U.S. population.

Label	No. of Attributes	No. of Records	Size of Lattice	$\epsilon = 1$ δ	$\epsilon' = 2$ δ'
USC	9	30,162	19,440	10^{-5}	1×10^{-9}
CS	8	100,937	15,552	10^{-6}	2×10^{-11}
TUS	9	539,253	34,992	10^{-6}	2×10^{-11}
HI	8	1,193,504	14,580	10^{-7}	4×10^{-14}

Table 1. Overview of the evaluation datasets.

The datasets have increasing volumes, ranging from about 30,000 to more than a million records. All include sensitive data such as demographics (e.g. sex, age) or health-related data. Table 1 provides an overview of the datasets and parameterizations we used in our experiments. It also shows results of the smoothness property, i.e. the probability δ' of violating 2-differential privacy.

8.2 Analysis of the Anonymization Operator

First we examine the amount of records preserved (i.e. not removed by random sampling or record suppression) by the anonymization operator, which is a generic utility estimate. We set $\epsilon_{search} = 0$ and used three full-domain generalization schemes defining *low*, *medium* or *high* relative generalization levels for the attributes in the datasets. We note that the parameter ϵ determines the degree of privacy provided together with δ while the relative generalization level balances the loss of information resulting from generalization against the loss of information resulting from record suppression – the higher the degree of generalization is, the more records are likely to become indistinguishable, and hence the fewer records have to be removed for violating k-anonymity. We focus on the parameters also investigated in Section 7.2. Figure 11 shows averages of 10 executions. All standard deviations were less than 1%.

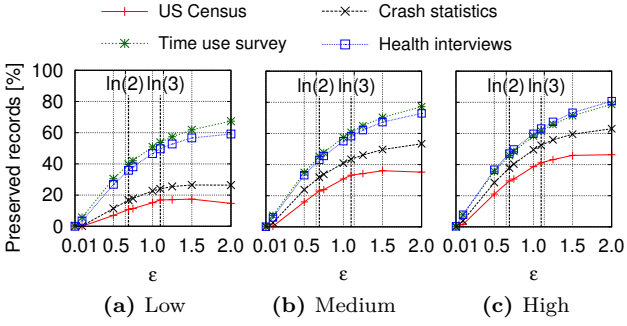
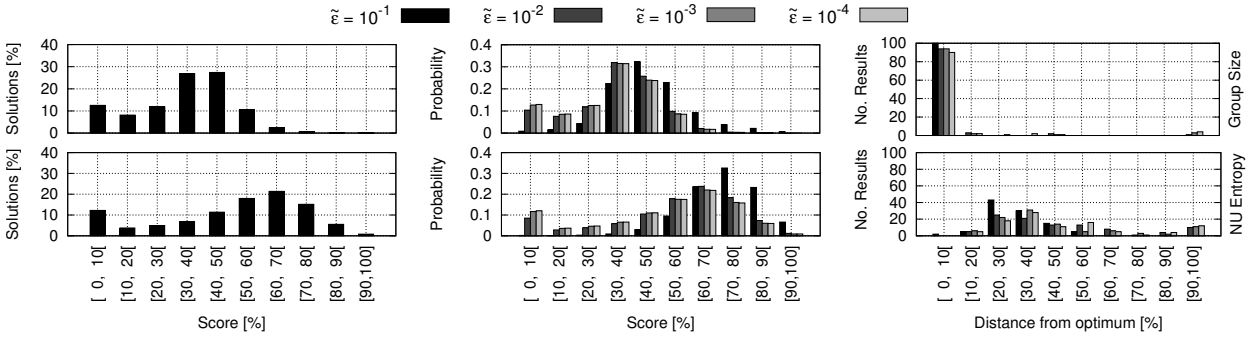


Fig. 11. Average number of records preserved by SafePub for each generalization scheme using various values of ϵ.

As can be seen, lower values of ϵ, and the resulting reduction of β and k (see Section 7.2), tendentially led to fewer records being preserved. Only for small datasets, low degrees of generalization and high values of ϵ the decrease of k did outweigh the decrease of β so that more records were preserved (see the "US Census" and "Crash statistics" datasets for $\epsilon = 2$ and $\epsilon = 1.5$). In all other cases the lower sampling probability dominated, especially for realistic values of $\epsilon \leq 1.5$.

1 http://arx.deidentifier.org/



(a) Distribution of quality **(b)** Probability mass function **(c)** Distribution of output quality

Fig. 12. (a) Distribution of data quality for different score functions. (b) Probability mass functions obtained with varying values of $\tilde{\epsilon}$. (c) Distributions of the quality of results of 100 executions of the exponential mechanism using various values of $\tilde{\epsilon}$.

We note that SafePub uses the highest possible value of β (see Section 3) for a given privacy budget (see Theorem 3). The results presented here justify this choice. They also indicate that values of ϵ_{anon} in the order of one are a good choice. Unless noted otherwise, we will use an overall budget of $\epsilon = 1$ in the following sections, which is a common setup [10, 46] and, as we will show, a good parameterization for our method as well.

8.3 Analysis of the Optimization Functions

We now investigate the effectiveness of the score functions and the quality of transformations selected by the exponential mechanism. We focus on Non-Uniform Entropy and Group Size, because the results obtained for the other score functions lied in between. We further focus on "US Census" and point out differences obtained using the larger datasets where applicable.

Figure 12a shows the distribution of (normalized) scores within the solution space. We note that the y-axis represents the probability of selecting a transformation with a score value in a given range when drawing from the uniform distribution. For the other datasets, the fraction of transformations with higher scores increased with growing volume, because the more records are contained, the less records are likely to be suppressed because they appear less than k times.

Figure 12b shows the probability mass functions used by the exponential mechanism when drawing a solution from the whole solution space using $\epsilon_{anon} = 1$ and various values of $\tilde{\epsilon}$ between 10^{-1} and 10^{-4}. We focus on relatively small values since the search strategy executes the exponential mechanism several times so that higher budgets for each execution would add up to an unusably high overall budget. For $\tilde{\epsilon} = 0.1$ the resulting probability distributions were significantly better than the distributions obtained when drawing from the uniform distribution (see Figure 12a). The improvements decreased with decreasing $\tilde{\epsilon}$ and increased significantly with increasing data volumes. The main reason is that larger datasets often lead to broader ranges of score values in the solution space so that the application of the exponential function according to Definition 3 yields higher differences between probabilities for good and bad solutions.

Figure 12c shows the results of 100 executions of the exponential mechanism. For each transformation selected, we calculated the difference to the optimal solution in terms of data quality using the model for which the score function has been designed. On average, we measured very good results of less than 4% for the Group Size model, even though solutions with a score in the range $[30\%, 50\%[$ were selected with the highest probability. This is because the according score function is not directly proportional to the quality model, but rather inversely proportional. Hence data quality increases significantly with increasing scores. The results for Non-Uniform Entropy were not as good with averages ranging from 31% ($\tilde{\epsilon} = 10^{-1}$) to 49% ($\tilde{\epsilon} = 10^{-4}$). The reason is that the according score function does not resemble the corresponding quality model as closely as the other score functions do. The results imply that a budget which is very small compared to the one required by the anonymization operator can suffice to achieve good results using the exponential mechanism.

8.4 Analysis of the Search Strategy

Next we analyze the influence of the number of steps performed by the search strategy on the quality of output data. We executed SafePub 10 times for each dataset and score function using varying numbers of steps. Since the previous results imply that a budget in the order of one is a good choice for the anonymization operator while a significantly smaller budget is sufficient for the exponential mechanism, we used an overall budget of $\epsilon = 1$ which we have split into various combinations of ϵ_{search} and ϵ_{anon}. Figure 14 shows the results

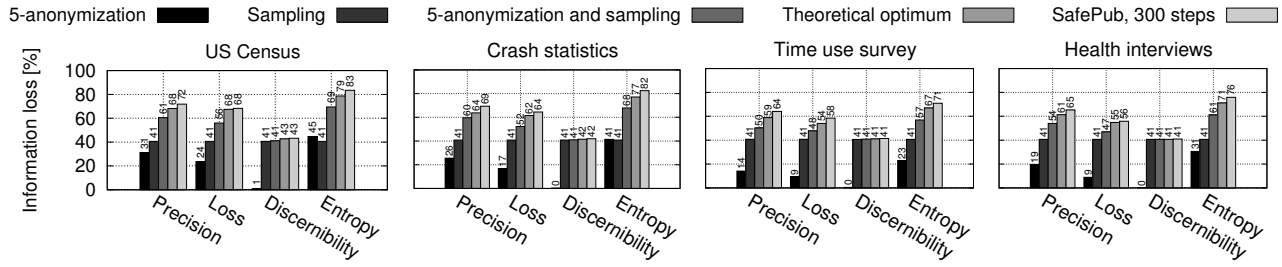


Fig. 13. Average information loss induced by SafePub for $\epsilon = 1$ compared with the (average) results of various baseline methods.

obtained for the "Health interviews" dataset, which are representative for the other datasets. The results for the Discernibility model were comparable to the results for the Granularity model. We normalized all values so that 0% corresponds to the input dataset and 100% to a dataset from which all information has been removed. All standard deviations were less than 12%.

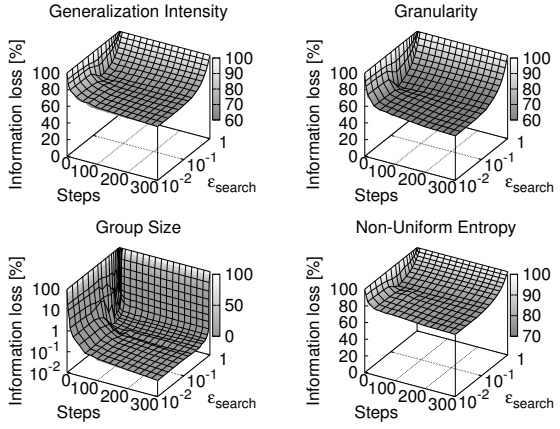


Fig. 14. Average information loss induced by SafePub for various step values and various values of ϵ_{search} with $\epsilon_{anon} = 1 - \epsilon_{search}$. The ϵ_{search} axis and the information loss values for the Group Size model are scaled logarithmically.

We note that increasing the number of steps performed by SafePub has two consequences: The number of executions of the exponential mechanism increases, while the budget $\tilde{\epsilon}$, which is used for each execution, decreases. It can be observed that the former effect tends to outweight the latter so that increasing the number of steps improves data quality. In all experiments the effect flattened at around 300 steps. Decreasing ϵ_{search} from 1 to 10^{-1} generally improved the results. Further reductions decreased data quality in some experiments and had no significant effects in the others. Hence, in the following sections, we will use a default parameterization of 300 steps, $\epsilon_{search} = 0.1$ and $\epsilon_{search} = 0.9$ unless noted otherwise. These values result in a budget of $\tilde{\epsilon} \approx 10^{-4}$ which did not perform as well as higher values when drawing from the whole solution space $\mathcal{G}_m$ (see Section 8.3). However, since the search strategy draws repeatedly out of subsets of $\mathcal{G}_m$, it can still select very good solutions as we will see in the next section.

8.5 Analysis of the Quality of Output

Here we analyze output data quality for the default parameterization and compare it with the quality obtained using various baseline methods: The optimal quality obtained with k-anonymization, by only using random sampling and by random sampling combined with k-anonymization. We also measured the quality of the theoretical optimum which can be obtained with SafePub by deterministically selecting the optimal generalization scheme rather than using the search strategy. Each of these methods constitutes a baseline in terms of output quality for (combinations of) transformations performed by SafePub, and hence illustrates their impact on output data quality. We note that none of them satisfies differential privacy but that all approaches have been implemented such that the optimal transformation according to a given quality model is selected. To establish a strict baseline we set $k = 5$, which is common in the literature [12, 13] but less conservative than other values, e.g. $k = 11$ which has been recommended by the European Medicines Agency (EMA) [14].

The results are shown in Figure 13. Numbers for the Group Size model are not included as we measured values of less than 2% for all approaches. It can be seen that SafePub removed a significant amount of information from the datasets, i.e. between 83% and 71% according to the Non-Uniform Entropy model and between 41% and 43% according to the Discernibility model. It can further be observed that random sampling contributed the most to these reductions (41%). The average difference between results of SafePub and the theoretical optimum was very small (less than 3%). We note that, even though SafePub produced near-optimal results, the fraction of the solution space which has been traversed by the search strategy was relatively small. When using the "Crash statistics" dataset, this fraction was about 5%. In the other cases, it was about 10%. This confirms that the search strategy performs very well using the

default parameters. In particular, it also achieves very good results for the Non-Uniform Entropy model, for which the exponential mechanism alone did not perform as well as for the other models (see Section 8.3).

8.6 Analysis of the Utility of Output

As there is not necessarily a strong correlation between loss of information and the actual usefulness of data, we now evaluate the performance of statistical classifiers built with the output of SafePub. This is the most common benchmarking workload for methods of privacy-preserving data publishing. We have used the class attributes listed in Table 2, which resulted in both binomial and multinomial classification problems.

Dataset	Class attributes	Number of instances
USC	(1) Marital status	8
	(2) Salary class	2
CS	(1) Hispanic origin	10
	(2) Race	20
TUS	(1) Marital status	7
	(2) Sex	3
HI	(1) Marital status	10
	(2) Education	26

Table 2. Overview of the class attributes used in our evaluations.

For each dataset and class attribute, we executed 100 runs of SafePub with varying numbers of steps, varying values of ϵ_{anon} and $\epsilon_{search} = 0.1$. We focused on ϵ_{anon}, since the previous results showed that small values of ϵ_{search} are sufficient and that ϵ_{anon} thus primarily determines the overall trade-off between privacy and utility provided by SafePub. We configured SafePub to use the score function which optimizes output data for training statistical classifiers (see Section 5.5). All attributes besides the class attribute were used as features, and we used generalization schemes which do not generalize the class attribute.

As a classification method we used decision trees generated with the well-known C4.5 algorithm [49] because this is the most frequently used method in our context. We point out that it is obviously possible to use other classification methods with our approach and that we have obtained comparable results using logistic regression classifiers [54]. We created the classifiers from output data and evaluated their prediction accuracy with input data using the approach presented in [16] and 10-fold cross-validation. We report *relative prediction accuracies*, which means that all values have been normalized so that 0% represents the accuracy of the trivial ZeroR method, which always returns the most frequent value of the class attribute, while 100% corresponds to the accuracy of C4.5 decision trees trained on input data.

Figure 15 shows the results of varying ϵ_{anon} using 300 steps. As can be seen, the impact of ϵ_{anon} was relatively small considering the strong effect on the number of preserved records (see Section 8.2). As expected, small values of ϵ_{anon} often resulted in sub-optimal accuracies. Values of about $\epsilon_{anon} = 0.9$ generally resulted in good performance. Further increasing the parameter decreased the accuracies obtained. The reason is that, although increasing ϵ_{anon} increases the number of preserved records, k also increases, which eventually causes a high degree of generalization.

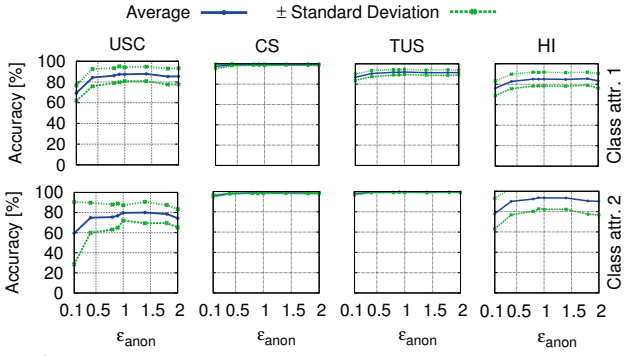


Fig. 15. Relative classification accuracies obtained using various values of ϵ_{anon}, $\epsilon_{search} = 0.1$ and 300 steps.

Figure 16 shows results obtained for varying numbers of steps and $\epsilon_{anon} = 0.9$. As can be seen, the performance of the classifiers improved with an increasing number of steps. The average accuracies obtained using 300 steps ranged from 82% when predicting the second class attribute of "US Census" to about 99% when predicting the class attributes of "Crash statistics". Results were rather stable (standard deviations of about 5%).

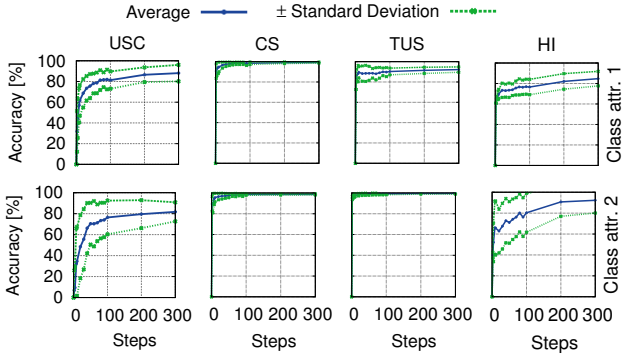


Fig. 16. Relative classification accuracies obtained using various numbers of steps, $\epsilon_{anon} = 0.9$ and $\epsilon_{search} = 0.1$.

Using the same setup, we also evaluated classification accuracies obtained using the output of the baseline methods discussed in Section 8.5 (sampling only, k-anonymization only), also optimized for building classifiers. All accuracies achieved were at least 97%.

In summary, these experiments justify our default parameterization, and we conclude that the differences

between the performance of classifiers trained with unmodified input or the output of baseline methods and classifiers trained with the output of SafePub are small. This indicates that although SafePub removes a significant amount of information, it does so in a controlled manner which preserves frequent patterns hidden in the data.

8.7 Comparison With Prior Work

In this section we will put our method into perspective by experimentally comparing it to related approaches. We have performed all experiments using the default configuration (300 steps, $\epsilon = 1$) and we have calculated δ as described in Section 7. Where applicable, we used the same hierarchies as in the previous experiments.

8.7.1 Comparison With Other Approaches for Differentially Private Statistical Classification

We compared SafePub to the following state-of-the-art algorithms: DiffGen [46], DiffP-C4.5 [19], LDA [55], SDQ [57] and DPNB [29]. We have exactly replicated the setups reported in the respective publications and refer to them for exact specifications. All evaluations used (variants of) the "US Census" dataset (see Section 8.1) and the "Nursery" dataset [42]. We point out that the other methods implement ϵ-differential privacy while SafePub satisfies the slight relaxation (ϵ, δ)-differential privacy, which potentially allows for higher data quality. However, unlike the other methods which output classifiers or synthetic microdata, SafePub outputs truthful microdata using a less flexible but truthful transformation technique.

Algorithm	DiffP-C4.5	LDA	DPNB	DPNB	SDQ
Dataset	US Census			Nursery	
Competitor	82.1%	80.8%	82%	90%	79.9%
SafePub	80.9%	81.5%	81.2%	83.7%	83.8%

Table 3. Comparison of absolute prediction accuracies for $\epsilon = 1$.

The results for all mechanisms except DiffGen, which we will address below, are listed in Table 3. As can be seen, the accuracies obtained using C4.5 and SafePub were comparable to the results of DiffP-C4.5, LDA and DPNB for the "US Census" dataset. For the "Nursery" dataset, SafePub outperformed SDQ, while DPNB outperformed SafePub by 6.3%. In all experiments, we measured standard deviations of $< 2\%$.

DiffGen is particularly closely related to SafePub because it also produces microdata using concepts from data anonymization (i.e. attribute transformation based on generalization hierarchies). Hence we have performed a more detailed analytical and experimental comparison. DiffGen employs a more flexible transformation model, subtree generalization, where values of an attribute can be transformed to different generalization levels (see Section 2.3). Analogously to SafePub, it also selects a transformation based on a user-specified number of iterative applications of the exponential mechanism (steps). However, in contrast to our approach, it does not achieve differential privacy by random sampling and k-anonymization, but rather by probabilistically generating synthetic records.

Using the implementation provided by the authors and our evaluation datasets we compared SafePub and DiffGen using C4.5 decision trees which were evaluated using 2/3 of the records as training data and the remaining 1/3 as test data (as proposed by the authors of DiffGen [46]). We used a privacy budget of $\epsilon = 1$ for both methods and increasing numbers of steps. The number of steps DiffGen can perform has a limit which depends on the heights of the generalization hierarchies and which was around 20 in our setup. For SafePub, we used between 0 and 300 steps since higher values did not improve the quality of results (see Section 8.4). We performed every experiment 20 times. Table 4 lists average execution times and standard deviations for the maximal number of steps measured on the hardware described in Section 7.2. Moreover, we included the optimal accuracies obtained using any number of steps.

Label	Class Attribute	Execution times		Max. Accuracies	
		SafePub	DiffGen	SafePub	DiffGen
USC	1	4.8 ± 1.0s	16.2 ± 0.7s	92.0%	85.0%
	2	5.1 ± 1.3s	21.9 ± 0.6s	87.3%	79.2%
CS	1	8.8 ± 0.7s	18.5 ± 1.6s	99.7%	97.9%
	2	8.9 ± 0.6s	6.5 ± 2.5s	99.9%	98.3%
TUS	1	54.2 ± 4.5s	28.7 ± 0.7s	93.6%	91.0%
	2	55.3 ± 2.0s	30.9 ± 0.6s	99.9%	99.7%
HI	1	98.0 ± 5.8s	61.1 ± 2.2s	87.7%	94%
	2	103.5 ± 9.2s	65.0 ± 2.1s	99.1%	64.0%

Table 4. Comparison of absolute execution times and maximal relative accuracies achieved for $\epsilon = 1$.

SafePub outperformed DiffGen regarding maximal accuracies in seven out of eight experiments. The accuracies obtained by SafePub when predicting the second class attribute of "Health interviews" were 35% higher than the results obtained by DiffGen. The minimal and maximal execution times of SafePub varied from between 4s and 7s ("US Census") to between 90s and 128s ("Health interviews"). The corresponding times of DiffGen varied from between 15s and 18s to between 62s and 70s. In summary, SafePub was faster than DiffGen for smaller datasets while DiffGen was faster than SafePub for larger datasets.

A more detailed analysis is provided in Figure 17, which shows execution times and relative accuracies obtained using different numbers of steps.

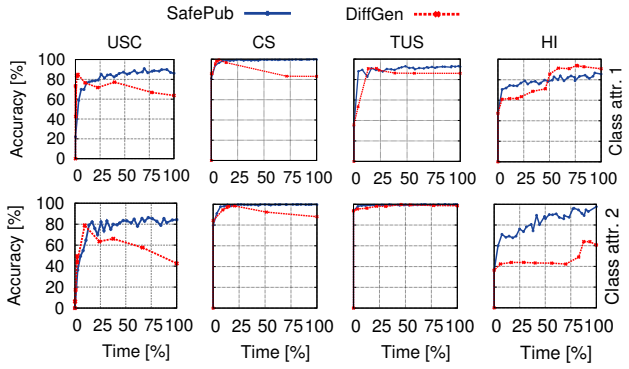


Fig. 17. Relative classification accuracies obtained for $\epsilon = 1$.

It can be seen that the accuracies achieved by SafePub improved monotonically over time (apart from minor fluctuations which are a result of randomization) while no such relationship can be observed for DiffGen. We explain this by the fact that SafePub is not likely to be trapped in a local minimum (see Section 6) while DiffGen can only keep on specializing a transformation once it has been selected. This implies that SafePub is easier to parameterize and enables trading execution times off against data quality.

8.7.2 Comparison With the Approach by Fouad et al.

We conclude our experimental evaluation by presenting a comparison with the approach which is most closely related to ours. Fouad et al. have also proposed a truthful (ϵ, δ)-differentially private microdata release mechanism using random sampling and generalization [18, 43].

Their algorithm replaces each record independently with a generalized record which is *t-frequent*, i.e. a generalization of at least t records from the input dataset. The authors show that the mechanism satisfies (ϵ, δ)-dif-ferential privacy, however, with unknown δ. They further show that an *upper bound* for δ can be calculated when t is chosen greater than a threshold $\lfloor T \rfloor$ [43, Theorem 4]. Knowing δ is, however, crucial for guaranteeing a known degree of privacy.

We analyzed $\lfloor T \rfloor$ and the resulting values of δ for various common input parameters. We emphasize that we chose all parameters in such a way that $\lfloor T \rfloor$ is as small as possible. Figure 18 shows the results for $\epsilon = 1$. As can be seen, δ decreases very quickly for an increasing number of attributes, while $\lfloor T \rfloor$ increases exponentially. For datasets with three attributes, $\lfloor T \rfloor$ equals 76, while for datasets with seven attributes, $\lfloor T \rfloor$ equals $1,217$ already. Hence, a very high degree of generalization is required to obtain known privacy guarantees.

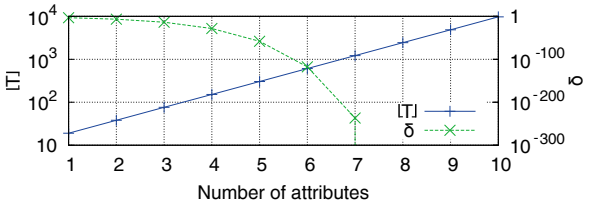


Fig. 18. Analysis of the approach by Fouad et al. The figure shows $\lfloor T \rfloor$ and corresponding values of δ for $\epsilon = 1$ and $h = 2$.

We experimentally evaluated the method choosing $\epsilon = 1$ and $t = \lfloor T \rfloor + 1$ so that the approach satisfies (ϵ, δ)-differential privacy. We performed the experiments ten times and report average results (standard deviations $< 1\%$). All information was removed from all datasets but "Health interviews" for which some information was preserved. However, 68% of records were removed and seven out of eight attributes were completely suppressed. With the models considered in this article, we measured reductions in data quality of between 97% and 99%, which renders the approach impractical.

9 Related Work

Other works have also investigated relationships between syntactic privacy models and differential privacy. Domingo-Ferrer and Soria-Comas have shown that there is a theoretical relationship between ϵ-differential privacy and a stochastic extension of t-closeness and that satisfying t-closeness can imply ϵ-differential privacy under certain assumptions [8]. Moreover, Soria-Comas et al. and Jafer et al. have also combined k-anonymity and differential privacy [27, 30]. While our approach uses k-anonymity in order to create a differentially private mechanism, these works employ k-anonymization to reduce the amount of noise that must be added.

Moreover, further differential privacy mechanisms have been proposed that use random sampling. Fan and Jin [17] as well as Jorgensen et al. [32] have used non-uniform random sampling to produce aggregate data. Hong et al. have used random sampling for protecting search logs [24]. These are all special-purpose mechanisms while SafePub is a generic microdata release algorithm.

For further differentially private microdata release methods see the surveys [6, 38]. Unlike SafePub, most of them are not truthful or use methods that are very different from those typically used in data anonymization. We have compared our approach to the notable

exception, i.e. the approach by Fouad et al. [18, 43], in Section 8.7.2 and found that it is not practical.

Differentially private machine learning is also an ongoing field of research (see the surveys [31, 51]). We have compared our approach to five different state-of-the-art methods in Section 8.7.1. We have performed a detailed experimental comparison with DiffGen [46] because of its conceptual similarities to our approach. Our results showed that our method, which is the only generic and truthful approach in the field, achieves accuracies that compare well to those of special-purpose mechanisms.

Gehrke et al. have also studied the approach by Li et al. [26], albeit from a purely theoretical perspective. They showed that it satisfies a privacy model called crowd-blending privacy. Informally, this model guarantees that every record r from the input dataset either blends into a "crowd" of at least k records or that r is essentially being ignored by the mechanism. Their work also indicates that the mechanism satisfies a relaxation of another model called zero-knowledge privacy [22].

10 Summary and Discussion

In this paper we have presented a flexible differentially private data release mechanism that produces truthful output data, which is important in some data publishing scenarios [3] and domains such as medicine [6]. While it has been argued that differential privacy is difficult to explain to non-experts the approach offers an intuitive notion of privacy protection: with a probability determined by ϵ the data of an individual will not be included at all and even if it is included it will only be released in a generalized form such that it cannot be distinguished from the similarly generalized data of at least $k-1$ other individuals, where k is determined by ϵ and δ.

Our evaluation showed that the method is practical and that values in the order of $\epsilon = 1$ are a good parameterization. The current implementation uses full-domain generalization and the k-anonymity privacy model, methods which have frequently been criticized for being too inflexible and too strict to produce output data of high quality [2]. However, our experiments have shown that statistical classifiers trained with the output of the generic method parameterized with an appropriate objective function perform as well as non-truthful differential privacy mechanism designed specifically for this use case. The reason is that while the approach indeed removes a significant amount of information it does so in a controlled manner which extracts frequent patterns. Compared to prior work, however, our approach provides slightly lower privacy guarantees.

While developing the score functions introduced in Section 5, we learned that optimization functions which have the form of sums to which every record or cell contributes a non-negative summand tend to have a low sensitivity. According score functions can often be obtained easily (see score functions for Data Granularity, Intensity and Classification). If the sensitivity is high, it can be possible to reduce it by division through the size of the dataset or by forming reciprocals (see score functions for Discernibility and Group Size). If this is not the case, it can be worthwhile to try to find functions with lower sensitivities which have related properties (see score function for Non-Uniform Entropy).

11 Future Work

An interesting line of future research is to develop score functions tailored to further quality models which address learning tasks such as regression or time-to-event analysis [54]. Based on our experiences presented in the previous section we are confident that, for example, the workload-aware quality models presented by LeFevre et al. in [37] can be integrated into the method.

Another potential direction for further work is to try to consider the effects of random sampling which may have been performed during data acquisition to reduce the amount of explicit random sampling that needs to be used by the mechanism.

In its current form SafePub is suited for protecting dense data of low to medium dimensionality as high-dimensional data is often sparse and hence cannot be k-anonymized while retaining sufficient data quality. We plan to investigate methods for vertically partitioning high-dimensional data, such that disassociated subsets of correlated attributes can be processed independently. Moreover, future work could investigate the crowd-blending and the zero-knowledge privacy models which provide other means of formalizing the notion of "hiding in a group" than our implementation. We point out that these models can also make it possible to prefer certain records, e.g. for publishing control or test data using random sampling which is slightly biased [26].

Finally, a variety of unified frameworks have been proposed for comparing the trade-off between privacy and utility provided by algorithms which implement privacy models, including syntactic ones and ϵ-differential privacy [4, 20, 41]. As the mechanism presented here is the first practical implementation of differential privacy for the release of truthful microdata, it would be interesting to compare it to other methods using such frameworks.

References

[1] A. Machanavajjhala et al. l-diversity: Privacy beyond k-anonymity. *Transactions on Knowledge Discovery from Data*, 1(1):3, 2007.

[2] B. C. M. Fung et al. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. CRC Press, 2010.

[3] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *International Conference on Data Engineering*, pages 217–228, 2005.

[4] J. Brickell and V. Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 70–78, 2008.

[5] C. Clifton and T. Tassa. On syntactic anonymity and differential privacy. In *International Conference on Data Engineering Workshops*, pages 88–93, 2013.

[6] F. K. Dankar and K. El Emam. Practicing differential privacy in health care: A review. *Transactions on Data Privacy*, 6(1):35–67, 2013.

[7] T. de Waal and L. Willenborg. Information loss through global recoding and local suppression. *Netherlands Official Statistics*, 14:17–20, 1999.

[8] J. Domingo-Ferrer and J. Soria-Comas. From t-closeness to differential privacy and vice versa in data anonymization. *Knowledge-Based Systems*, 74:151–158, 2015.

[9] C. Dwork. An ad omnia approach to defining and achieving private data analysis. In *International Conference on Privacy, Security, and Trust in KDD*, pages 1–13, 2008.

[10] C. Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19, 2008.

[11] K. El Emam and L. Arbuckle. *Anonymizing Health Data*. O'Reilly Media, 2013.

[12] K. El Emam and F. K. Dankar. Protecting privacy using k-anonymity. *Jama-J Am. Med. Assoc.*, 15(5):627–637, 2008.

[13] K. El Emam and B. Malin. Appendix b: Concepts and methods for de-identifying clinical trial data. In *Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk*, pages 1–290. National Academies Press (US), 2015.

[14] European Medicines Agency. External guidance on the implementation of the european medicines agency policy on the publication of clinical data for medicinal products for human use. EMA/90915/2016, 2016.

[15] F. Prasser et al. Lightning: Utility-driven anonymization of high-dimensional data. *Transactions on Data Privacy*, 9(2):161–185, 2016.

[16] F. Prasser et al. A tool for optimizing de-identified health data for use in statistical classification. In *IEEE International Symposium on Computer-Based Medical Systems*, 2017.

[17] L. Fan and H. Jin. A practical framework for privacy-preserving data analytics. In *International Conference on World Wide Web*, pages 311–321, 2015.

[18] M. R. Fouad, K. Elbassioni, and E. Bertino. A supermodularity-based differential privacy preserving algorithm for data anonymization. *IEEE Transactions on Knowledge and Data Engineering*, 26(7):1591–1601, 2014.

[19] A. Friedman and A. Schuster. Data mining with differential privacy. In *International Conference on Knowledge Discovery and Data Mining*, pages 493–502, 2010.

[20] G. Cormode et al. Empirical privacy and empirical utility of anonymized data. In *IEEE International Conference on Data Engineering Workshops*, pages 77–82, 2013.

[21] G. Poulis et al. Secreta: a system for evaluating and comparing relational and transaction anonymization algorithms. In *International Conference on Extending Database Technology*, pages 620–623, 2014.

[22] J. Gehrke, E. Lui, and R. Pass. Towards privacy for social networks: A zero-knowledge based definition of privacy. In *Theory of Cryptography Conference*, pages 432–449, 2011.

[23] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley publishing company, 2nd edition, 1994.

[24] Y. Hong, J. Vaidya, H. Lu, and M. Wu. Differentially private search log sanitization with optimal output utility. In *International Conference on Extending Database Technology*, pages 50–61, 2012.

[25] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *International Conference on Knowledge Discovery and Data Mining*, pages 279–288, 2002.

[26] J. Gehrke et al. Crowd-blending privacy. In *Advances in Cryptology*, pages 479–496. Springer, 2012.

[27] J. Soria-Comas et al. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *VLDB J.*, 23(5):771–794, 2014.

[28] J. Soria-Comas et al. t-closeness through microaggregation: Strict privacy with enhanced utility preservation. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):3098–3110, 2015.

[29] J. Vaidya et al. Differentially private naive bayes classification. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, pages 571–576, 2013.

[30] Y. Jafer, S. Matwin, and M. Sokolova. Using feature selection to improve the utility of differentially private data publishing. *Procedia Computer Science*, 37:511–516, 2014.

[31] Z. Ji, Z. C. Lipton, and C. Elkan. Differential privacy and machine learning: a survey and review. *CoRR*, abs/1412.7584, 2014.

[32] Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In *IEEE International Conference on Data Engineering*, pages 1023–1034, April 2015.

[33] K. El Emam et al. A globally optimal k-anonymity method for the de-identification of health data. *J. Am. Med. Inform. Assn.*, 16(5):670–682, 2009.

[34] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn. Flash: efficient, stable and optimal k-anonymity. In *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Conference on Social Computing (SocialCom)*, pages 708–717, 2012.

[35] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *International Conference on Management of Data*, pages 49–60, 2005.

[36] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *International Conference on Data Engineering*, pages 25–25, 2006.

[37] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization techniques for large-scale datasets. *ACM Transactions on Database Systems*, 33(3):1–47, 2008.
[38] D. Leoni. Non-interactive differential privacy: A survey. In *International Workshop on Open Data*, pages 40–52, 2012.
[39] N. Li, W. Qardaji, and D. Su. On sampling, anonymization, and differential privacy: Or, k-anonymization meets differential privacy. In *ACM Symposium on Information, Computer and Communications Security*, pages 32–33, 2012.
[40] N. Li, W. H. Qardaji, and D. Su. Provably private data anonymization: Or, k-anonymity meets differential privacy. *CoRR*, abs/1101.2604, 2011.
[41] T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *International Conference on Knowledge Discovery and Data Mining*, pages 517–526, 2009.
[42] M. Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.
[43] M. R. Fouad, K. Elbassioni, and E. Bertino. Towards a differentially private data anonymization. CERIAS Tech Report 2012-1, Purdue Univ., 2012.
[44] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2007.
[45] F. D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *International Conference on Management of Data*, pages 19–30, 2009.
[46] N. Mohammed et al. Differentially private data release for data mining. In *International Conference on Knowledge Discovery and Data Mining*, pages 493–501, 2011.
[47] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *International Conference on Management of Data*, pages 665–676, 2007.
[48] F. Prasser, F. Kohlmayer, and K. A. Kuhn. The importance of context: Risk-based de-identification of biomedical data. *Methods of information in medicine*, 55(4):347–355, 2016.
[49] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.
[50] F. Ritchie and M. Elliott. Principles- versus rules- based output statistical disclosure control in remote access environments. *IASSIST Quarterly*, 39(2):5–13, 2015.
[51] A. D. Sarwate and K. Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE Signal Processing Magazine*, 30(5):86–94, 2013.
[52] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, Oct. 2002.
[53] L. Willenborg and T. De Waal. *Statistical disclosure control in practice*. Springer Science & Business Media, 1996.
[54] I. H. Witten and F. Eibe. *Data mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
[55] X. Jiang et al. Differential-private data publishing through component analysis. *Transactions on Data Privacy*, 6(1):19–34, Apr. 2013.
[56] Z. Wan et al. A game theoretic framework for analyzing re-identification risk. *PloS one*, 10(3):e0120592, 2015.
[57] N. Zhang, M. Li, and W. Lou. Distributed data mining with differential privacy. In *IEEE International Conference on Communications*, pages 1–5, 2011.

A Proof of Theorem 11

Proof. For the purpose of this proof we will use the following representation of the function d which is obtained as an intermediate result in the proof of [40, Theorem 1]:

$$d(k,\beta,\epsilon') = \max_{n\in\mathbb{N}} \sum_{\{j\in\mathbb{N} \mid j\geq k\wedge j>\gamma n\wedge j\leq n\}} f(j;n,\beta).$$

Let us fix an arbitrary $\epsilon' \geq \epsilon$ and recall that $\gamma = \gamma(\epsilon')$ is actually a function of ϵ'. It is easy to see that $\epsilon' \geq \epsilon$ implies:

$$\gamma(\epsilon') = \frac{\mathrm{e}^{\epsilon'} - 1 + \beta}{\mathrm{e}^{\epsilon'}} \geq \frac{\mathrm{e}^{\epsilon} - 1 + \beta}{\mathrm{e}^{\epsilon}} = \gamma(\epsilon).$$

Hence we have:

$$\forall n \in \mathbb{N} : \{j \in \mathbb{N} \mid j \geq k \wedge j > \gamma(\epsilon')n \wedge j \leq n\} \subseteq \{j \in \mathbb{N} \mid j \geq k \wedge j > \gamma(\epsilon)n \wedge j \leq n\}.$$

This implies

$$\begin{aligned} d(k,\beta,\epsilon') &= \max_{n\in\mathbb{N}} \sum_{\{j\in\mathbb{N} \mid j\geq k\wedge j>\gamma(\epsilon')n\wedge j\leq n\}} f(j;n,\beta) \\ &\leq \max_{n\in\mathbb{N}} \sum_{\{j\in\mathbb{N} \mid j\geq k\wedge j>\gamma(\epsilon)n\wedge j\leq n\}} f(j;n,\beta) \\ &= d(k,\beta,\epsilon) \end{aligned}$$

which proofs the monotonicity. Furthermore, we have $\epsilon' \geq \epsilon = -\ln(1-\beta)$ so that $(\epsilon', d(k,\beta,\epsilon'))$-differential privacy is indeed satisfied according to Theorem 3. □

B Proofs of Sensitivities

B.1 Granularity (Theorem 6)

Proof. Let $k \in \mathbb{N}$ be an arbitrary integer, let $g \in \mathcal{G}_m$ be an arbitrary generalization scheme and let $D_1, D_2 \in \mathcal{D}_m$ be arbitrary datasets satisfying $|D_1 \oplus D_2| = 1$. Without loss of generality we assume $D_1 = D_2 \cup \{r\}$. We will use the notation $g(r) = (\tilde{r}_1, ..., \tilde{r}_m)$ and point out that $\forall i = 1, ..., m : \ 0 \leq \frac{leaves_i(\tilde{r}_i)}{|\Omega_i|} \leq \frac{leaves_i(*)}{|\Omega_i|} = 1$ holds.

– If $g(r) = *$ holds or $g(r) \neq *$ appears less than k times in $g(D_1)$, then $g(r)$ is suppressed in both $S(D_1)$ and $S(D_2)$ with $S(D_1) = S(D_2) \cup \{*\}$. We can conclude:

$$\begin{aligned} &|gran_k(D_1,g) - gran_k(D_2,g)| = \\ &\left| \left(\sum_{(r'_1,...,r'_m)\in S(D_2)} \sum_{i=1}^{m} \frac{leaves_i(r'_i)}{|\Omega_i|} \right) + \left(\sum_{i=1}^{m} \underbrace{\frac{leaves_i(*)}{|\Omega_i|}}_{=1} \right) \right. \\ &\left. - \left(\sum_{(r'_1,...,r'_m)\in S(D_2)} \sum_{i=1}^{m} \frac{leaves_i(r'_i)}{|\Omega_i|} \right) \right| = m \ . \end{aligned}$$

– If $g(r) \neq *$ appears k times in $g(D_1)$, then it is not suppressed in $S(D_1)$ but in $S(D_2)$ with

$$S(D_1) = (S(D_2) \setminus \underbrace{\{*, ..., *\}}_{k-1-times}) \cup \underbrace{\{g(r), ..., g(r)\}}_{k-times}.$$

We can conclude:

$$\begin{aligned}
&|gran_k(D_1, g) - gran_k(D_2, g)| \\
&= \left| \left(\sum_{j=1}^{k} \sum_{i=1}^{m} \frac{leaves_i(\tilde{r}_i)}{|\Omega_i|} \right) - \left(\sum_{j=1}^{k-1} \sum_{i=1}^{m} \underbrace{\frac{leaves_i(*)}{|\Omega_i|}}_{=1} \right) \right| \\
&= \left| \underbrace{\left(\sum_{j=1}^{k} \sum_{i=1}^{m} \frac{leaves_i(\tilde{r}_i)}{|\Omega_i|} \right)}_{=:\sigma \in [0, km]} - (k-1)m \right| \\
&\leq \begin{cases} (k-1)m, & if \sigma \in [0, (k-1)m) \\ m, & if \sigma \in [(k-1)m, km] \end{cases}.
\end{aligned}$$

– If $g(r) \neq *$ appears more than k times in $g(D_1)$, then $g(r)$ is not suppressed in both $S(D_1)$ and $S(D_2)$ with $S(D_1) = S(D_2) \cup \{g(r)\}$. We can conclude:

$$|gran_k(D_1, g) - gran_k(D_2, g)| = \sum_{i=1}^{m} \underbrace{\frac{leaves_i(\tilde{r}_i)}{|\Omega_i|}}_{\leq 1} \leq m\ .$$

In summary we have:

$$|gran_k(D_1, g) - gran_k(D_2, g)| \leq \begin{cases} (k-1)m, & if k > 1 \\ m, & if k = 1 \end{cases}$$

□

B.2 Discernibility (Theorem 7)

In the following we will frequently employ the triangle inequality and indicate its application with (T). In order to prove the sensitivity of the Discernibility score function we will first propose two lemmas:

Lemma 12. *For all $D_1, D_2 \subseteq (\Omega_1 \cup \Lambda_1) \times ... \times (\Omega_m \cup \Lambda_m)$ with $D_1 = D_2 \cup \{r'\}$ the following holds: $|\phi(D_1) - \phi(D_2)| \leq 5$.*

Proof. If $D_2 = \emptyset$ holds we have $D_1 = \{r'\}$ and can conclude:

$$|\phi(D_1) - \phi(D_2)| = |1 - 0| = 1.$$

In the following we will assume $D_2 \neq \emptyset$ and define $c \coloneqq |D_1|$, $n \coloneqq |\{r' \in D_1\}|$, $y \coloneqq |\{* \in D_1\}|$ and

$$x \coloneqq \sum_{E \in EQ(D_1): r' \notin E} |E|^2 = \sum_{E \in EQ(D_2): r' \notin E} |E|^2.$$

– If $r' \neq *$ holds we have $|\{r' \in D_2\}| = n - 1$ and $|\{* \in D_2\}| = y$. Moreover, $x + n^2 = \sum_{E \in EQ(D_1)} |E|^2 = \sum_{r \in D_1 : r \neq *} |\{r \in D_1\}| \leq \sum_{r \in D_1} |D_1| = c^2$ holds. We can conclude:

$$\begin{aligned}
&|\phi(D_1) - \phi(D_2)| \\
&= \left| \frac{x + n^2 + yc}{c} - \frac{x + (n-1)^2 + y(c-1)}{c-1} \right| \\
&= \left| \frac{-x - n^2 + 2nc - c}{c(c-1)} \right| \underset{(T)}{\leq} \frac{x + n^2}{c(c-1)} + \frac{2n-1}{c-1} \\
&\leq \frac{c^2}{c(c-1)} + \frac{2c-1}{c-1} = \underbrace{\frac{3c-1}{c-1}}_{\searrow, c \nearrow \wedge\, c \geq 2} \leq 5.
\end{aligned}$$

– If $r' = *$ holds we have $|\{* \in D_2\}| = y - 1$ and we can conclude using $x = \sum_{E \in EQ(D_1)} |E|^2 = \sum_{r \in D_1 : r \neq *} |\{r \in D_1\}| \leq \sum_{r \in D_1 \setminus \{*\}} |D_1| \leq c(c-1)$:

$$\begin{aligned}
|\phi(D_1) - \phi(D_2)| &= \left| \frac{x + yc}{c} - \frac{x + (y-1)(c-1)}{c-1} \right| \\
&= \left| \frac{-x + c^2 - c}{c(c-1)} \right| \underset{(T)}{\leq} \frac{x}{c(c-1)} + 1 \leq 2.
\end{aligned}$$

In summary we have $|\phi(D_1) - \phi(D_2)| \leq 5$. □

Lemma 13. *For every integer $k \geq 2$ and all $D_1, D_2 \subseteq (\Omega_1 \cup \Lambda_1) \times ... \times (\Omega_m \cup \Lambda_m)$ satisfying*

$$D_1 = (D_2 \setminus \underbrace{\{*, ..., *\}}_{k-1-times}) \cup \underbrace{\{r', ..., r'\}}_{k-times}$$

with $r' \neq *$ the following holds:

$$|\phi(D_1) - \phi(D_2)| \leq \frac{k^2}{k-1} + 1.$$

Proof. With the definitions

$$\begin{aligned}
&c \coloneqq |D_1|, \\
&n \coloneqq |\{r' \in D_1\}| = |\{r' \in D_2\}| + k, \\
&y \coloneqq |\{* \in D_1\}| = |\{* \in D_2\}| - k + 1, \\
&x \coloneqq \sum_{E \in EQ(D_1): r' \notin E} |E|^2 = \sum_{E \in EQ(D_2): r' \notin E} |E|^2
\end{aligned}$$

and using $x + n^2 \leq \sum_{r \in D_1} |D_1| = c^2$ we have:

$$\begin{aligned}
&|\phi(D_1) - \phi(D_2)| \\
&= \left| \frac{x + n^2 + yc}{c} - \frac{x + (n-k)^2 + (y+k-1)(c-1)}{c-1} \right| \\
&= \left| \frac{-x - n^2 + 2knc - k^2 c - kc^2 + kc + c^2 - c}{c(c-1)} \right| \\
&\underset{(T)}{\leq} \left| \frac{-x - n^2}{c(c-1)} \right| + k \left| \frac{2n - k - c + 1}{c-1} \right| + 1 \\
&\leq \frac{c}{c-1} + k \left| \frac{2n + 1 - (k+c)}{c-1} \right| + 1. \qquad (8)
\end{aligned}$$

– If $2n+1 \geq k+c$ holds we can conclude:

$$\left|\frac{2n+1-(k+c)}{c-1}\right| = \frac{2n+1-(k+c)}{c-1} \underset{n\leq c}{\leq} \frac{2c+1-(k+c)}{c-1} = \frac{c+1-k}{c-1} \underset{k\geq 2}{\leq} 1.$$

– Otherwise we have:

$$\left|\frac{2n+1-(k+c)}{c-1}\right| = \frac{k+c-(2n+1)}{c-1} \underset{n\geq k}{\leq} \frac{k+c-(2k+1)}{c-1} = \frac{c-1-k}{c-1} \leq 1.$$

We can conclude from Inequation (8):

$$|\phi(D_1)-\phi(D_2)| \leq \underbrace{\frac{c}{c-1}}_{\searrow, c\nearrow \wedge\, c\geq k} + k + 1 = \frac{k^2}{k-1} + 1. \square$$

We can now prove Theorem 7 as follows:

Proof. Let $k \in \mathbb{N}$ be an arbitrary integer, let $g \in \mathcal{G}_m$ be an arbitrary generalization scheme and let $D_1, D_2 \in \mathcal{D}_m$ be arbitrary datasets satisfying $|D_1 \oplus D_2| = 1$. Without loss of generality we assume $D_1 = D_2 \cup \{r\}$.

– If $S(D_1) = S(D_2) \cup \{*\}$ or $S(D_1) = S(D_2) \cup \{g(r)\}$ holds (which is always satisfied in the case of $g(r) = *$ or $k = 1$) we can conclude using Lemma 12:

$$|disc_k(D_1, g) - disc_k(D_2, g)| \leq 5.$$

– If $k \geq 2$ and $g(r) \neq *$ hold and $g(r)$ is suppressed in $S(D_2)$ but not in $S(D_1)$ we have

$$S(D_1) = (S(D_2) \setminus \underbrace{\{*, ..., *\}}_{k-1-times}) \cup \underbrace{\{g(r), ..., g(r)\}}_{k-times}$$

and can conclude using Lemma 13:

$$|disc_k(D_1, g) - disc_k(D_2, g)| \leq \frac{k^2}{k-1} + 1.$$

In summary we can conclude:

$$|disc_k(D_1, g) - disc_k(D_2, g)| \leq \begin{cases} 5, & if k = 1 \\ \frac{k^2}{k-1} + 1, & if k > 1 \end{cases} \square$$

B.3 Non-Uniform Entropy (Theorem 8)

We can prove Theorem 8 using the two lemmas proposed in Appendix B.2 as follows:

Proof. Let $k \in \mathbb{N}$ be an arbitrary integer, let $g \in \mathcal{G}_m$ be an arbitrary generalization scheme and let $D_1, D_2 \in \mathcal{D}_m$ be arbitrary datasets satisfying $|D_1 \oplus D_2| = 1$. Without loss of generality we assume $D_1 = D_2 \cup \{r\}$. Then we have:

$$\begin{aligned} &|ent_k(D_1, g) - ent_k(D_2, g)| \\ &= \left|\sum_{i=1}^{m} \phi(p_i(S(D_1))) - \phi(p_i(S(D_2)))\right| \\ &\underset{(T)}{\leq} \sum_{i=1}^{m} |\phi(p_i(S(D_1))) - \phi(p_i(S(D_2)))|. \end{aligned} \tag{9}$$

Let us fix an arbitrary $i = 1, ...m$, define $g(r) =: (r'_1, ..., r'_m)$ and regard $p_i(S(D_1))$ and $p_i(S(D_2))$ as datasets with one attribute.

– If $p_i(S(D_1)) = p_i(S(D_2)) \cup \{*\}$ or $p_i(S(D_1)) = p_i(S(D_2)) \cup \{r'_i\}$ holds (which is always satisfied in the case of $r'_i = *$ or $k = 1$) we can conclude using Lemma 12:

$$|\phi(p_i(S(D_1))) - \phi(p_i(S(D_2)))| \leq 5.$$

– If $k \geq 2$ and $r'_i \neq *$ hold and $g(r)$ is suppressed in $S(D_2)$ but not in $S(D_1)$ we have

$$p_i(S(D_1)) = (p_i(S(D_2)) \setminus \underbrace{\{*, ..., *\}}_{k-1-times}) \cup \underbrace{\{r'_i, ..., r'_i\}}_{k-times}$$

and can conclude using Lemma 13:

$$|\phi(p_i(S(D_1))) - \phi(p_i(S(D_2)))| \leq \frac{k^2}{k-1} + 1.$$

In summary we have

$$|\phi(p_i(S(D_1))) - \phi(p_i(S(D_2)))| \leq \begin{cases} 5, & if k = 1 \\ \frac{k^2}{k-1} + 1, & if k > 1 \end{cases}$$

and can conclude from Inequation (9):

$$|ent_k(D_1, g) - ent_k(D_2, g)| \leq \begin{cases} 5m, & if k = 1 \\ (\frac{k^2}{k-1} + 1)m, & if k > 1 \end{cases} \square$$

B.4 Statistical Classification (Theorem 9)

Proof. Let $k \in \mathbb{N}$ be an arbitrary integer, let $g \in \mathcal{G}_m$ be an arbitrary generalization scheme and let $D_1, D_2 \in \mathcal{D}_m$ be arbitrary datasets satisfying $|D_1 \oplus D_2| = 1$. Without loss of generality we assume $D_1 = D_2 \cup \{r\}$. For ease of notation we define $w_1(\cdot) := w(S(D_1), \cdot)$ and $w_2(\cdot) := w(S(D_2), \cdot)$. Moreover, we define FV to be the subset of all records in $S(D_2)$ which have the same combination of feature attribute values as $g(r)$, i.e. $FV := \{r' \in S(D_2) \mid fv(r') = fv(g(r))\}$.

If $fv(g(r))$ is suppressed as a consequence of generalization then $S(D_1)$ and $S(D_2)$ differ only in records

with a weight of zero in either set, i.e. we have $C \coloneqq \{r' \in S(D_1) : w_1(r') = 1\} = \{r' \in S(D_2) : w_2(r') = 1\}$ which implies:

$$class_k(D_1, g) = \sum_{r' \in S(D_1)} w_1(r') = \sum_{r' \in C} w_1(r')$$
$$= \sum_{r' \in C} w_2(r') = \sum_{r' \in S(D_2)} w_2(r') = class_k(D_2, g).$$

In the following we will regard the case that $fv(g(r))$ is not suppressed, which implies $g(r) \neq *$.

- If $g(r)$ appears less than k times in $g(D_1)$ then it is suppressed in $S(D_1)$ with $S(D_1) = S(D_2) \cup \{*\}$. We can argue as above: $class_k(D_1, g) = class_k(D_2, g)$.
- If $g(r)$ appears k times in $g(D_1)$ then it is not suppressed in $S(D_1)$ while we have $g(r) \notin S(D_2)$, in particular $g(r) \notin FV$, and

$$S(D_1) = (S(D_2) \setminus \underbrace{\{*, ..., *\}}_{k-1-times}) \dot\cup \underbrace{\{g(r), ..., g(r)\}}_{k-times}.$$

Moreover, all records in $S(D_2)$ which have the same feature values as $g(r)$ are also contained in $S(D_1)$, i.e. $FV \subseteq S(D_1) \cap S(D_2)$ holds, and these are the only records contained in both $S(D_1)$ and $S(D_2)$ which may have different weights in these sets, i.e. $\forall r' \in (S(D_1) \cap S(D_2)) \setminus FV : w_1(r') = w_2(r')$ holds. We can conclude:

$$|class_k(D_1, g) - class_k(D_2, g)| =$$
$$\left| \left(k \cdot w_1(g(r)) + \sum_{r' \in S(D_1) \cap S(D_2)} w_1(r') \right) - \left((k-1) \cdot \underbrace{w_2(*)}_{=0} + \sum_{r' \in S(D_1) \cap S(D_2)} w_2(r') \right) \right| =$$
$$\left| k \cdot w_1(g(r)) + \sum_{r' \in FV} \big(w_1(r') - w_2(r') \big) \right|. \quad (10)$$

Let r'_{maj} denote the record with the most frequent class attribute value among all records in $S(D_1)$ which have the same feature values as $g(r)$.
If $r'_{maj} \neq g(r)$ holds we have $w_1(g(r)) = 0$ and r'_{maj} is also the record with the most frequent class attribute value in FV with

$$\forall r' \in FV : w_1(r') = w_2(r') = \begin{cases} 1, & \text{if } r' = r'_{maj} \\ 0, & \text{otherwise} \end{cases}.$$

Using Equation (10) we can conclude:

$$|class_k(D_1, g) - class_k(D_2, g)| = 0.$$

If $r'_{maj} = g(r)$ holds we have $w_1(g(r)) = 1$ and $\forall r' \in FV : w_1(r') = 0$. Moreover, the record $\tilde{r}_{maj}$ with the most frequent class value in FV can appear at most k times in FV (because otherwise, $\tilde{r}_{maj} \in FV \subseteq S(D_1)$ would have a class value more frequent than the one of $g(r)$ in $S(D_1)$, which contradicts $r'_{maj} = g(r)$). Hence, we have:

$$0 \leq \sum_{r' \in FV} \underbrace{w_2(r')}_{=1 \text{ iff } r' = \tilde{r}_{maj}} \leq k.$$

We can conclude using Equation (10):

$$|class_k(D_1, g) - class_k(D_2, g)| = k - \sum_{r' \in FV} w_2(r') \leq k.$$

- If $g(r)$ appears $l > k$ times in $g(D_1)$, then it appears $l - 1 \geq k$ times in $g(D_2)$. It follows that $g(r)$ is not suppressed in both $S(D_1)$ and $S(D_2)$ with $S(D_1) = S(D_2) \cup \{g(r)\}$. Moreover, $g(r) \in FV \subseteq S(D_2) \subseteq S(D_1)$ holds, and the records in FV are the only ones which may have a different weight in $S(D_1)$ and $S(D_2)$, i.e. $\forall r' \in S(D_2) \setminus FV : w_1(r') = w_2(r')$ holds. We can conclude:

$$|class_k(D_1, g) - class_k(D_2, g)| =$$
$$\left| w_1(g(r)) + \sum_{r' \in S(D_2)} w_1(r') - \sum_{r' \in S(D_2)} w_2(r') \right| =$$
$$\left| w_1(g(r)) + \sum_{r' \in FV} \big(w_1(r') - w_2(r') \big) \right|. \quad (11)$$

If $r'_{maj} \neq g(r)$ holds we can argue similar as above:

$$|class_k(D_1, g) - class_k(D_2, g)| = 0.$$

If $r'_{maj} = g(r)$ holds we have

$$\forall r' \in FV : w_1(r') = \begin{cases} 1, & \text{if } r' = g(r) \\ 0, & \text{otherwise} \end{cases},$$

$|\{g(r) \in FV\}| = l - 1$ (so that the record with the most frequent class value appears at least $l - 1$ times in FV) and $\forall r' \in FV, r' \neq g(r) : |\{r' \in S(D_2)\}| \leq l$ (because otherwise, there would exist a record $\tilde{r}_{maj} \in FV \subseteq S(D_1), \tilde{r}_{maj} \neq g(r)$ with a class value which is more frequent than the one of $g(r)$ in $S(D_1)$, which contradicts $r'_{maj} = g(r)$). Hence we have:

$$l - 1 \leq \sum_{r' \in FV} w_2(r') \leq l.$$

We can conclude using Equation (11):

$$|class_k(D_1, g) - class_k(D_2, g)| = l - \sum_{r' \in FV} w_2(r') \leq 1.$$

In summary we can conclude:

$$|class_k(D_1, g) - class_k(D_2, g)| \leq k \qquad \square$$