Optimizing Data Masking for Predictive Utility: A Correlation-Based Approach

Omar Islam Laskar Fatemeh Ramezani Khozestani Ishika Nankani Sohrab Namazi Nia
Senjuti Basu Roy Kaustubh Beedkar
IIT Delhi NJIT USA

Abstract

Data regulations such as GDPR or CCPA mandate that data publishers anonymize datasets before publication. A critical challenge thus arises in ensuring that sensitive information remains protected while maintaining the dataset's utility for downstream applications. This work addresses the critical problem of selecting optimal masking configurations that preserve correlations/associations between feature attributes and the predictor variable, thereby safeguarding predictive utility in the downstream task. A major challenge arises from the fact that, due to privacy restrictions, the original raw dataset is typically inaccessible, and only limited statistical summaries (such as 1D histograms) are available. To overcome this, we propose a novel utility optimizer that generalizes to arbitrary correlation and association measure as long as the measure involves estimating joint distributions. Furthermore, we introduce an efficient framework that employs iterative proportional fitting (IPF) to approximate the joint distribution of features and labels from limited available statistics. We provide scalable solution: when only 1D histograms of the predictors are available, and when that too is not accessible. Experimental results demonstrate that our method effectively selects masking functions, optimizing data masking to preserve predictive performance in anonymized datasets. [KB#0: make more concrete once the evaluation is complete.]

ACM Reference Format:
Omar Islam Laskar Fatemeh Ramezani Khozestani Ishika Nankani Sohrab Namazi Nia Senjuti Basu Roy Kaustubh Beedkar. 2025. Optimizing Data Masking for Predictive Utility: A Correlation-Based Approach. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

The rise of data publishing platforms, such as AWS Data Exchange, Snowflake Data Marketplace, and various open data portals, has been fueled by the growing use of data-driven decision-making and machine learning across industries like finance, healthcare, and public policy. These platforms support research, innovation, and economic growth by providing curated datasets. However, the increased use of personal and sensitive data has led to the need for strict compliance with privacy regulations like GDPR, California Consumer Privacy Act (CCPA), which require anonymization of datasets before publication. While anonymization protects individual identities, it often transforms datasets in ways that can affect their usability, particularly for tasks that depend on their statistical or predictive utility.

The predictive utility of a dataset is often measured by the correlation between the predictors and the class label. There are various methods available for quantifying correlation. Mutual Information (MI), a model-agnostic metric that measures the dependency between variables, has been widely adopted for quantifying correlation between attributes [1, 2, 7, 9, 13, 20, 26]. The Chi-Square Test of Independence [3] evaluates whether two categorical variables are associated by comparing observed and expected frequencies. A recent study [14] links approximate functional dependency measures [11], particularly the g_3 error, to an upper bound on classification accuracy.

Data publishing challenge centers on balancing privacy and predictive utility. Techniques like generalization, suppression, or perturbation are used to mask data and limit disclosure risks, but they often reduce the dataset's predictive utility, especially for tasks like machine learning that rely on correlation between the predictors and the class label. For example, imagine a healthcare data set that is published to predict *health status* (class label) using predictors such as age and zip code. To comply with privacy regulations, quasi-identifiers like age and zip code must be anonymized, but different masking strategies can distort feature-class label correlations. For example, generalizing age (wide intervals of age, like, 0-30, 31-60, 61+) may lose important distinctions between younger and older individuals, or truncating zip codes to the first two digits may weaken its predictive power, while adding noise to weight can disrupt subtle correlations between weight and health status. If the masking strategy is poorly chosen, it could break the dataset's ability to produce meaningful insights.

[KB#1: todo: add an experiment showing how different masking configurations impact the correlation.]

Example 1. *Consider a dataset on a data publishing platform, consisting of m distinct attributes and a class label. Due to stringent privacy regulations and data security policies, the original raw dataset is not accessible to the data publisher. In many practical scenarios, the platform only sees the masked attribute values upon selecting a masking function from the available ones, and at best, limited statistical summaries (such as 1D histograms of individual attributes) can be obtained. However, joint distributions that capture dependencies between multiple attributes are typically out of reach.*

The platform has access to multiple masking configurations, with each configuration contains a set of pairs. Each pair is a masking function f to be applied on a predictor A. The platform's goal is to select the optimal masking configuration for the predictors, so that

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

when the predictors are masked using that configuration, the change in its predictive utility is minimized (i.e., the change in correlation between each masked predictor and the class label, relative to the correlation between the unmasked predictor and the class label).

To that end, this work proposes a framework that selects the optimal masking configuration from a set of available ones, aiming to preserve the correlation strength between the attributes (specifically between the predictors and the class label) as much as possible. The framework is generic and can accommodate multiple correlation measures, provided the correlation is quantified based on the joint distribution of the involved variables. The framework is orthogonal to k-anonymity [24] and other privacy-preserving techniques [15, 17], meaning it can be integrated into existing privacy-preserving methods to maintain both privacy and predictive utility.

Contributions. We propose the notion of *predictive utility deviation* under different correlation measures. Given a set of predictors, a class label, a set of masking functions to be applied to the predictors, and a correlation measure, predictive utility deviation quantifies *the total change in correlation between the unmasked predictors and the class label compared to the masked predictors and the class label.* We then define the **Optimal Predictive Utility Aware Masking Configuration Selection Problem**, as follows: Given a set of possible masking configurations to be applied over a set of predictors (attributes) and a class label, where the joint frequency distributions between the unmasked predictors and the class label are unknown (due to the sensitive nature of the predictors), select the masking configuration to be applied on the predictors, such that, the *predictive utility deviation is minimized.*

To minimize predictive utility deviation, the grand challenge is to be able to estimate the unmasked predictor from the known information and reconstruct the joint distribution of the predictor and the class label from there. Examples of known information could be 1D histograms of the predictor, e.g., for Age, there exists 5 individuals within age 10-16, 45 individuals between 17-80, masked values of the predictor (i.e., masked Age), and the masking function. For the latter, it may be known that there exists 10 young and 40 old individuals in the masked Age, and the masking function dictates that *Young* $\rightarrow$ $10-45$, *Old* $\rightarrow>$ 45. From this, the reconstructed Age has to satisfy the following: there are 50 individuals of Age between $10-80$, among them 5 individuals are between $10-16$, 10 individuals are between $10-45$, 45 individuals between $17-80$, and 40 individuals are above 45. Clearly, there exists a prohibitively large number of possible distributions of *Age* that satisfies all these constraints and any of these could be a likely solution. In the absence of any additional information, we assume that any value of *Age* between $10-80$ is equally likely. We formalize this as an optimization problem, i.e., estimate joint distribution of each predictor and the class label (i.e., *Age* and *Health Status*) such that the distribution of the predictor (i.e, *Age*) is as uniform as possible within the given range (i.e, between $10-80$,) but the reconstructed joint distribution satisfies all constraints imposed on the marginals (e.g., available known information on *Age*). Once the joint distribution is estimated for each predictor-masking-function pair per configuration, the problem selects the one that minimizes the predictive utility deviation. We solve this problem by adapting Iterative Proportional Fitting (or IPF) [10, 12], an iterative algorithm that is designed to adjust distribution in statistics.

Age	Weight	Zipcode	Health
21	55	21162	Good
25	58	21168	Good
30	63	22170	Moderate
42	71	23175	Poor
48	80	23173	Poor
55	78	25165	Good

Figure 1: Original Dataset D.

We study two variants of the *Optimal Predictive Utility Aware Masking Configuration Selection Problem* : (a) When 1D histograms of unmasked predictors are available. Additionally, masking functions to be applied on the predictors, as well as the masked predictors are also known; (b) When no distributional statistics of the unmasked predictors are available, but the masking functions to be applied on the predictors are known along with their masked values. We formalize both of these variants as optimization problems and demonstrate how IPF could be adapted to solve both. Clearly, when more information is known about the predictors (case a), the joint estimation turns out to be more accurate.

We demonstrate the efficacy of the proposed framework considering three different correlation measures, namely, *Mutual Information, Chi-Squared Test, and g_3 eror.*[SBR#0: write more on what we have observed. - what are SOTA. what are other baselines, datasets, and key observations.]

[KB#2: write a para on why related work don't sufficiently address these challenges.]

2 Preliminaries & Data Model

We start by introducing basic definitions about data masking and presenting the notation we use throughout the paper.

Dataset. We consider a dataset as a collection of related observations comprising features/ attributes/predictors and a label organized in tabular format. More formally, denote by $D = \{ (x_i, y_i) \}_{i=1}^{N}$ a dataset with N rows (data points). The i-th data point $x_i = (x_{i1}, x_{i2}, \ldots, x_{im}) \in \mathbb{X}^m$) is a feature vector over m attributes/predictors ($\mathcal{A} = \{A_1, A_2 \ldots A_m\}$). For the i-th data point, $y_i \in \mathbb{Y}$ is the label associated. Figure 1 shows an example dataset with three attributes (`Age`, `Weight`, and `Zipcode`) and a label (`Health`).

Domain of an attribute. For the j-th attribute A_j, $Dom(A_j)$ denotes its domain, referring to the set of all possible values that A_j can take. The i-th domain value for A_j is referred to as $Dom(A_j^i)$. Similarly, domain of the label $\mathbb{Y}$ is $Dom(\mathbb{Y})$.

[SBR#1: $\mathbb{X}_j \rightarrow \mathbb{X}'_j$ - may not be needed]

Marginal Distribution of an Attribute. The marginal distribution of an attribute is the frequency distribution of the different values of that attribute when considered independently of other attributes. For attribute A_j with domain $Dom(A_j)$, let $f(A_j^i)$ denote the frequency of the domain value A_j^i. Marginal distribution of A_j is denoted as $marginal A_j$, and defined as follows: $i \in Dom(A_j) f(A_j^i)$, $\Sigma_{i \in Dom(A_j)} f(A_j^i) = N$.

Masking Function. A *masking function* M_{jk} takes as input the i-th domain value $Dom(A_j^i)$ of A_j and outputs its masked value $Dom(A_j^{i'})$. In this work, we only focus on scalar and value-based functions. For example, a masking function $\texttt{blur}(v, c)$ to blue zip codes takes as input a zip code value v, and a number characters c and outputs a masked zip code, e.g., $\texttt{blur}(21162,2) = 211xx$.

For each attribute $A_j, j \in \{1, 2, \ldots, m\}$, there exists a set of candidate masking functions

$$\mathcal{M}_j = \left\{ M_{j1}, M_{j2}, \ldots, M_{j|\mathcal{M}_j|} \right\}$$

[SBR#2: there are too many M's in different form of notations - could be confusing to the readers]

Masking Configuration & Masked Dataset. A *masking configuration* is a tuple $\boldsymbol{M} = (M_1, M_2, ..., M_m)$, which applies the masking function M_j on attribute A_j, where $M_j \in \mathcal{M}_j$. Applying $\boldsymbol{M}$ to D produces a *masked dataset* $D_{\boldsymbol{M}} = \left\{ (x_i', y_i) \right\}_{i=1}^{N}$, where

$$x_i' = (M_1(x_{i1}), M_2(x_{i2}), ..., M_m(x_{im}))$$

. For example, consider three masking configurations (for brevity we omit other function parameters)

$\boldsymbol{M}_1$ = (`bucketize(Age), blur(Weight), Suppress(Zipcode)`)
$\boldsymbol{M}_2$ = (`bucketize(Age), bucketize(Weight), Zipcode`)
$\boldsymbol{M}_3$ = (`blur(Age), blur(Weight), Zipcode`)

Figure 3 shows masked datasets $D_{\boldsymbol{M}_1}$, $D_{\boldsymbol{M}_2}$, and $D_{\boldsymbol{M}_3}$ as a result of applying $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, and $\boldsymbol{M}_3$ to D.

Predictive Utility Measures. The predictive utility of an attribute refers to the amount of information it contributes to the model's ability to predict the class label $\mathbb{Y}$. Correlation, association between the predictors and the class label are some natural ways to quantify predictive utility. Our framework generalizes to all such measures designed for discrete/categorical data as long as the predictive quality is quantified by measuring the joint distribution of the predictor and the class label. For continuous attributes, we assume that the data is effectively discretized. Some of the correlation and association measures that we closely study are defined below.

Chi-Square Test. The Chi-Square test is used for categorical attributes to assess the association between each feature and the class label. If a feature is highly associated with the class labe;, it is more likely to provide useful information to the model, and thus, it is selected for inclusion in the model.

Mutual Information. The mutual information between predictor A_j and class label Y is defined as:

$$I(A_j; \mathbb{Y}) = \sum_{i \in Dom(A_j)} \sum_{y \in Dom(\mathbb{Y})} P(A_j^i, y) \log \left(\frac{P(A_j^i, y)}{P(A_j^i, y) P(y)} \right)$$

Where:

- $P(A_j^i, y)$ is the joint distribution of A_j and $\mathbb{Y}$.
- $P(A_j)$ and $P(y)$ are the marginal probability distributions of A_j and $\mathbb{Y}$, respectively.

g_3 **Error.** Given the attribute A_j and class label $\mathbb{Y}$, g_3 the error counts the number of records that must be deleted to satisfy the full functional dependency between $A_j \rightarrow \mathbb{Y}$.

$$g_3(A_j, \mathbb{Y}) = N - max\{|s| \mid s \in \mathcal{D},\ s \models X \rightarrow Z\}$$
$$g_3(X \rightarrow Z, \mathcal{D}) = G_3(X \rightarrow Z, \mathcal{D})/N$$

[SBR#3: I need to edit g_3 further] **Chi-Square Statistic.** The Chi-Square statistic between attribute A_j and class label $\mathbb{Y}$ is given by:

$$\chi^2 = \sum_{i=1}^{Dom(A_j) \times Dom(\mathbb{Y})} \frac{(O_i - E_i)^2}{E_i}$$

Where:

- χ^2 = Chi-Square statistic
- O_i = Observed frequency for the i-th category
- E_i = Expected frequency for the i-th category

[SBR#4: the notations were inconsistent earlier]

Predictive Utility Deviation. For a given dataset D, a feature/predictor A_j, and a class label $\mathbb{Y}$, let ρ be the predictive utility measure, where $\rho(D, A_j, \mathbb{Y})$ returns a scalar value that reflects predictive utility between A_j and $\mathbb{Y}$. If the m predictors in D are masked using a masking configuration $\boldsymbol{M} = (M_1, M_2, ..., M_m)$, we define the *predictive utility deviation* (PUD) as the sum of the absolute differences between the predictive utility computed on the original dataset D and that of the masked dataset $D_{\boldsymbol{M}}$:

$$\Delta_{PU}(D, D_{\boldsymbol{M}}, \rho) = \sum_{i=1}^{m} |\rho(D, A_j, \mathbb{Y}) - \rho(D_{\boldsymbol{M}}, M(A_j), \mathbb{Y})| \quad (1)$$

2.1 Problem Definition

Problem Statement. **(Identify Optimal Masking Configuration to Minimize Predictive Utility Deviation.)** *Given a dataset D, a predictive utlity measure ρ, and a set $\mathcal{M}_{set} = \{ \boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_K \}$ of K predefined masking configurations, find the masking configuration $\boldsymbol{M}^* \in \mathcal{M}_{set}$ that minimizes predictive utility deviation, i.e.,*

$$\boldsymbol{M}^* = \underset{\boldsymbol{M} \in \mathcal{M}_{set}}{\text{argmin}}\ \Delta_{PU}(D, D_{\boldsymbol{M}}, \rho)$$

The primary challenge comes in the form that to be able to compute any predictive utility measure ρ, the joint frequency distribution between each A_j and $\mathbb{Y}$, as well as the marginal distributions of each A_j and $\mathbb{Y}$ must be known. When the raw data D itself is not accessible, the joint distributions between the predictors and the class label are unknown, so Δ_{PU} (Equation 1) cannot be directly calculated from the data. We study two variants of this problem to that end.

Problem Statement. **(Variant I: Identify optimal masking configuration in the presence of marginal distributions.)** *Given a dataset D where the joint distributions of the attributes and the class label are not accessible, but the marginal distributions $marginalA_j$ of each attribute A_j is available, and a set $\mathcal{M}_{set} = \{ \boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_K \}$ of K predefined masking configurations, for a given predictive utility measure ρ, find the masking configuration $\boldsymbol{M}^* \in \mathcal{M}_{set}$ that minimizes the predictive utility deviation.*

Problem Statement. **(Variant II: Identify optimal masking configuration in the absence of marginal distributions.)** *Given a dataset D where neither marginals nor joint distributions are accessible, for a given predictive utlity measure ρ, and a given*

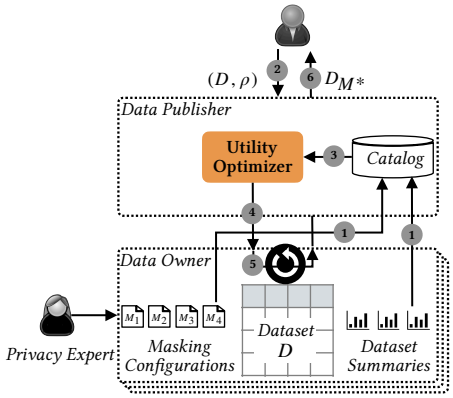

Figure 2: Framework Overview

set $\mathcal{M}_{set} = \{ \boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_K \}$ of K predefined masking configurations, find the masking configuration $\boldsymbol{M}^* \in \mathcal{M}_{set}$ that minimizes the predictive utility deviation.

3 Framework Overview

Figure 2 illustrates the schematics of our framework. [KB#3: elaborate along the lines]

- Allows users to request for datasets via a data publisher (e.g., exchange, data market) specifying a certain predictive utility metric ρ.
- Data owner collaborates with a privacy expert to define candidate masking configurations.
- The data owner shares masking configurations and data summaries with the publisher that stored in the catalog.
- The catalog also stores schema, dataset, etc.
- The utility optimizer selects with optimal masking configuration
- The data owner anonymizes the dataset using the optimal masking configration, which is shared with the user via the publisher.

4 Predictive Utility

Intuition. The key idea behind devising a utility metric is to quantify how well the "essential properties" of a dataset are preserved after anonymization, particularly in the context of its intended use. Specifically, for datasets utilized in machine learning, this typically involves preserving the predictive relationships between features and the target variable. A well-designed utility metric must strike a balance between capturing the dataset's fidelity (its closeness to the original) and providing a robust framework for evaluating different anonymization strategies. The metric should align with the specific objectives of the downstream task, whether classification, regression, or clustering, ensuring that the anonymized dataset remains as effective as the original in supporting accurate model development.

4.1 G_3 Error Measure-Based Utility

A utility metric also reflects the broader trade-off between data privacy and usability. While the application of a masking configuration $\boldsymbol{M}$ to a dataset D "distorts" data to allow meeting privacy constraints, the utility metric must allow identifying configurations that minimize these distortions. This involves taking into account dataset's structural dependencies and patterns, like functional dependencies $X \rightarrow Y$, that are relevant to the task. We propose to leverage the G_3 error measure for functional dependency errors, which quantifies how well a dependency $X \rightarrow Y$ is preserved. In predictive modeling, the relationship between input features (X) and the target variable (Y) is crucial. The G_3 measures captures thi relationship by calculating the smallest subset of data that needs to be removed to make the dependency hold exactly. A high G_3 value indicates that many tuples must be removed, suggesting the dependency is weak or violated in the dataset. Conversely, a low G_3 value suggests the dependency is well-preserved. Therefore, a masking configuration that minimally alter the G_3 measure is likely to maintain the predictive utility of the dataset.

G_3 Error Measure. Following [11], for a dependency $X \rightarrow Y$ in a dataset D, the G_3 error measure $g_3(X \rightarrow Y, D)$ is defined as:

$$g_3(X \rightarrow Y, D) = \frac{\mathrm{G}_3(X \rightarrow Y, D)}{N},$$

where:

$$\mathrm{G}_3(X \rightarrow Y, D) = N - \max \{ |d| \mid d \subseteq D, d \models X \rightarrow Y \}.$$

Here, $\mathrm{G}_3(X \rightarrow Y, D)$ denotes the minimum number of tuples that need to be removed from D to make the dependency $X \rightarrow Y$ hold exactly. $g_3(X \rightarrow Y, D)$ scales this value relative to the size of the dataset, providing a normalized measure of the dependency's error in D.

G_3 Measure for Predictive Utility. The utility of anonymized data depends on preserving its suitability for the downstream ML task. For the dependency $X \rightarrow Y$, where X is a feature and Y is the predictor (label):

> A high-utility anonymized dataset should exhibit minimal change in the G_3 measure for $X \rightarrow Y$ compared to the original dataset.

This ensures that the predictive relationship between X and Y is preserved as much as possible during data masking.

Thus, the G_3 Error Difference for $X \rightarrow Y$ is:

$$\Delta g_3(X \rightarrow Y) = |g_3(X \rightarrow Y, D) - g_3(X' \rightarrow Y, D')|,$$

where D is the original dataset, and D' is the masked dataset.

Optimization Problem. We model the problem as selecting the masking configuration that minimizes tha total change in the G_3 measure for dependencies $X \rightarrow Y$, where X represents each attribute and Y is the target variable.

$$\boldsymbol{M}^* = \arg \min_{\boldsymbol{M}_k \in \mathcal{M}_{\text{set}}} \sum_{j=1}^{m} \Delta g_3(X_j \rightarrow Y), \tag{2}$$

where: X_j represents the j-th attribute, Y is the predictor variable (label), and

$$\Delta g_3(X_j \rightarrow Y) = |g_3(X_j \rightarrow Y, D) - g_3(X'_j \rightarrow Y, D'_k)| \tag{3}$$

with D'_k being the dataset masked using configuration $\boldsymbol{M}_k$.

Age'	Weight'	Height'	Health
20-29	5x	*	Good
20-29	5x	*	Good
30-39	6x	*	Moderate
40-49	7x	*	Poor
40-49	8x	*	Poor
50-59	8x	*	Good

(a) Masked Dataset D'_1.

Age'	Weight'	Height	Health
20-40	50-60	162	Good
20-40	50-60	168	Good
20-40	60-69	170	Moderate
40-60	70-79	175	Poor
40-60	80-89	173	Poor
40-60	70-79	165	Good

(b) Masked Dataset D'_2.

Age'	Weight'	Height	Health
2x	5x	162	Good
2x	5x	168	Good
3x	6x	170	Moderate
4x	7x	175	Poor
4x	8x	173	Poor
5x	8x	165	Good

(c) Masked Dataset D'_3.

Figure 3: Masked datasets after applying masking configurations M_1, M_2, and M_3.

5 Utility Estimation

We now delve into the methodology for estimating the change in the g_3 measure (Δg_3) resulting from applying a masking configuration to a dataset. In this section, we discuss our approach based on reconstructing the joint distribution of attribute-label (X_i, Y) pairs using limited knowledge of D. In particular, we assume availability of 1D histograms, and leverage iterative proportional fitting (IPF) to approximate the original contingency matrix. In what follows, we describe how these approximations enable the computation of g_3 for both the original and masked datasets, identifying violations of functional dependencies.

5.1 Overall Idea

Recall that the g_3 measures assesses how well a functional dependency $X \rightarrow Y$ is preserved by quantifying the minimum fraction of tuples that need to be removed to make the dependency hold exactly.

Computation of g_3 Using Contingency Matrix. We can compute g_3 by the join distribution $P(X, Y)$, which specifies the co-occurence frequencies of the values of X and Y. This distribution is typically represented in a *contingency matrix*, where

- rows correspond to distinct values of X,
- columns correspond to distinct values of Y, and
- entries represent the frequency or probability of each (X, Y) pair.

More formally, let $C_{X,Y}$ denote the contingency matrix for an attribute X and label Y and denote by $C_{X,Y}(x, y)$ the count of tuples with $X = x$ and $Y = y$. A violation of the dependency $X \rightarrow Y$ occur when a single row (fixed $X = x$) has non-zero entries in more than one column, indicating that x maps to multiple y values. The g_3 measure relies on identifying the largest subset of rows that satisfy $X \rightarrow Y$, and scaling this subset by the total number of tuples, i.e.,

$$g_3(X \rightarrow Y) = \frac{|D| - \#\text{violations}}{|D|}$$

Matrix Reconstruction for Original Data. While a masked dataset can be accessed (by applying a masking configuration on D), we cannot access the original data. Also, note that when a masking configuration is applied, the join distribution $P(X, Y)$ changes. We can estimate the utility loss due to masking by comparing the g_3 measure fore the original data and the masked data. However, since the the original data is inaccessible, we reconstruct $P(X, Y)$ for the original data using limited knowledge of D and masked dataset D' by leveraging iterative proportional fitting (IPF).

5.2 Reconstructing Joint Distribution Using 1D Histograms and IPF

Here we assume knowledge of 1D histograms for X and Y. Given that the original dataset is unavailable, we utilize 1D histograms to represent the marginal frequency distribution of each attribute X_i and the target variable Y. Let $h_{X_i}(x)$ and $h_Y(y)$ denote the frequency of attribute value x for X_i and y for Y, respectively.

IPF-based Reconstruction. The overall approach is described as Algorithm 1. We start by initializing the contingency matrix $C'_{X',Y}$ obtained by applying masking configuration $\boldsymbol{M}_k$. We then leverage IPF to iteratively adjust $C'_{X',Y}$ to ensure that its row sums match h_{X_i}, and its column sums match h_Y, while preserving the proportionality of the original matrix. After convergence, the adjusted matrix $C^*_{X_i,Y}$ estimates the join distribution $P(X_i, Y)$.

Computing Change in g_3. After reconstructing the original contingency matrix $C^*_{X,Y}$ and obtaining the masked contingency matrix $C'_{X,Y}$, we compute the g_3 measure for both as described above and subsequently Δg_3. The total utility loss U_k is the sum of Δg_3 across all attributes. Finally, the configuration with smallest U_k is selected.

6 Predictive Utility

Depending on the available information, we handle three cases: (1) 1D histograms known, (2) 2D histograms known, and (3) no histograms known. In what follows, we describe how these approximations enable the computation of g_3 for both the original and masked datasets, identifying violations of functional dependencies.

6.1 Overall Idea

Recall that the g_3 measure assesses how well a functional dependency $X \rightarrow Y$ is preserved by quantifying the minimum fraction of tuples that need to be removed to make the dependency hold exactly.

Computation of g_3 Using Contingency Matrix. We can compute g_3 by the join distribution $P(X, Y)$, which specifies the co-occurence frequencies of the values of X and Y. This distribution is typically represented in a *contingency matrix*, where

- rows correspond to distinct values of X,
- columns correspond to distinct values of Y, and

- entries represent the frequency or probability of each (X, Y) pair.

More formally, let $C_{X,Y}$ denote the contingency matrix for an attribute X and label Y and denote by $C_{X,Y}(x, y)$ the count of tuples with $X = x$ and $Y = y$. A violation of the dependency $X \to Y$ occurs when a single row (fixed $X = x$) has non-zero entries in more than one column, indicating that x maps to multiple y values. The g_3 measure relies on identifying the largest subset of rows that satisfy $X \to Y$, and scaling this subset by the total number of tuples, i.e.,

$$g_3(X \to Y) = \frac{|D| - \#\text{violations}}{|D|}.$$

6.2 Case 1: 1D Histogram Known

Here we assume knowledge of 1D histograms for X and Y. Given that the original dataset is unavailable, we utilize 1D histograms to represent the marginal frequency distribution of each attribute X_i and the target variable Y. Let $h_{X_i}(x)$ and $h_Y(y)$ denote the frequency of attribute value x for X_i and y for Y, respectively.

IPF-based Reconstruction. The overall approach is described as Algorithm 1. We start by initializing the contingency matrix $C'_{X',Y}$ obtained by applying masking configuration $\boldsymbol{M}_k$. We then leverage IPF to iteratively adjust $C'_{X',Y}$ to ensure that its row sums match h_{X_i}, and its column sums match h_Y, while preserving the proportionality of the original matrix. After convergence, the adjusted matrix $C^*_{X_i,Y}$ estimates the join distribution $P(X_i, Y)$.

Computing Change in g_3. After reconstructing the original contingency matrix $C^*_{X,Y}$ and obtaining the masked contingency matrix $C'_{X,Y}$, we compute the g_3 measure for both as described above and subsequently Δg_3. The total utility loss U_k is the sum of Δg_3 across all attributes. Finally, the configuration with the smallest U_k is selected.

6.3 Case 2: 2D Histogram Known

In this case, we assume the full 2D histogram (contingency matrix) of the original data is known. This eliminates the need for reconstruction using IPF. Instead, the joint distribution $P(X, Y)$ is directly available from the original data.

Direct Computation of g_3. Since $C_{X,Y}$ is known for the original data, we directly compute $g_3(X \to Y)$ for both the original and masked datasets. The Δg_3 is then calculated as the difference in g_3 values between the original and masked datasets. This case provides the most accurate estimation of Δg_3 as no reconstruction is required.

6.4 Case 3: No Histogram Known

When no histogram data is available, we face the challenge of estimating the joint distribution $P(X, Y)$ with minimal prior knowledge. In this case, only the total row sums (overall counts for X) are known, and we aim to reconstruct the contingency matrix by minimizing the difference between all cell values while satisfying the row sum constraints.

Estimation by Minimizing Differences. We initialize the contingency matrix with equal probabilities across all cells and iteratively adjust the values to match the known row sums while minimizing the differences between cell values. This method provides an approximate reconstruction of $P(X, Y)$, enabling the computation of g_3 for the masked data. The Δg_3 is then calculated as the difference between this estimated g_3 and the g_3 for the masked dataset.

Challenges and Limitations. This approach is less accurate than the previous cases due to the lack of marginal or joint distributions, but it still provides a reasonable approximation for evaluating utility loss.

Algorithm 1 Finding the Best Masking Configuration

Require: Set of attributes $\{X_1, X_2, \ldots, X_m\}$, target Y, set of masking configurations $\mathcal{M}_{\text{set}}$, available histogram data (1D or 2D)
Ensure: Optimal masking configuration $\boldsymbol{M}^*$

```
Initialize U_best ← ∞, M* ← None
for each M_k ∈ M_set do
    Apply masking configuration M_k and compute contingency matrices {C'_{X_iY}}_{i=1}^m
    for each attribute X_i do
        if 2D histogram is known then
            Use original C_{X_iY} directly without reconstruction
        else if 1D histogram is known then
            Initialize C*_{X_iY} ← C'_{X_iY} {Start with masked contingency matrix}
            while Row/column sums of C*_{X_iY} do not match h_{X_i} and h_Y do
                Adjust rows of C*_{X_iY} to match h_{X_i}
                Adjust columns of C*_{X_iY} to match h_Y
        else if No histogram is known then
            Initialize C*_{X_iY} with equal values such that Σ_y C*_{X_iY}(x, y) = h_{X_i}(x) for all x
            while Σ_x C*_{X_iY}(x, y) ≠ h_Y(y) for any y do
                Update C*_{X_iY}(x, y) ← C*_{X_iY}(x, y) + Δ such that Δ minimizes the deviation across all cells
                Ensure column sums Σ_x C*_{X_iY}(x, y) = h_Y(y)
        Estimate g3(X_i → Y, D) using C*_{X_iY} or C_{X_iY}
        Estimate g3(X_i → Y, D') using C'_{X_iY}
        Compute Δg3(X_i → Y) ← |g3(X_i → Y, D) − g3(X_i → Y, D')|
    Compute total utility loss U_k ← Σ_{i=1}^m Δg3(X_i → Y)
    if U_k < U_best then
        Update U_best ← U_k, M* ← M_k
return M*
```

6.5 Discussions

Overall, our approach enables efficient computation of utility loss Δg_3 without direct access to the original dataset. By using histograms and IPF (when applicable), or minimal assumptions (when no histograms are available), we reconstruct a reasonable approximation of the original joint distribution. This ensures that our evaluations of masking configurations are both practical and accurate. Moreover, avoiding the computational and storage overhead of materializing full datasets for every configuration makes our methodology scalable to large datasets and numerous configurations.

Discussions. Overall, our approach enables efficient computation of utility loss Δg_3 without direct access to the original dataset. By using histograms and IPF, we reconstruct a reasonable approximation of the original joint distribution, ensuring that our evaluations of masking configurations are both practical and accurate. This method avoids the computational and storage overhead of materializing full datasets for every configuration, making it scalable to large datasets and numerous configurations.

6.6 Illustrative Example

6.6.1 1D Histogram Known.

1D Histograms of the D. Table 1 shows the simplified 1D histograms for each attribute from the original data, used as row sums during IPF:

Table 1: 1D Histograms of Original Data D.

Value	AgeCnt	WeightCnt	HeightCnt
21	1	–	–
25	1	–	–
30	1	–	–
42	1	–	–
48	1	–	–
55	1	–	–
55	–	1	–
58	–	1	–
63	–	1	–
71	–	1	–
80	–	1	–
78	–	1	–
162	–	–	1
168	–	–	1
170	–	–	1
175	–	–	1
173	–	–	1
165	–	–	1

IPF Reconstruction for Each Masked Dataset. We now reconstruct $(X, \texttt{Health})$ for each attribute $X \in \{\texttt{Age}, \texttt{Weight}, \texttt{Height}\}$ under $\texttt{D}_1\texttt{'}$, $\texttt{D}_2\texttt{'}$, and $\texttt{D}_3\texttt{'}$.

For brevity, we show detailed steps only for Age.

Masking 1 ($\texttt{D}_1\texttt{'}$). : Age vs. Health

Paired with Age row sums $\{21, 25, 30, 42, 48, 55\} \mapsto 1$ from Table 1, we set up IPF constraints. After iterative scaling, the final table is consistent with original Age row sums and masked Health column sums.

Masking 2 ($\texttt{D}_2\texttt{'}$). : Age vs. Health

Again, IPF uses row sums $\{21, 25, 30, 42, 48, 55\} \mapsto 1$. The final IPF output satisfies these row sums for actual Ages while matching the above column sums for Health.

Table 2: 2D Histogram from Masked $\texttt{D}_1\texttt{'}$ (Age' vs. Health).

	Health			ColSum
Age'	Good	Mod	Poor	
20-29	2	0	0	2
30-39	0	1	0	1
40-49	0	0	2	2
50-59	1	0	0	1
	3	1	2	6

Table 3: 2D Histogram from Masked $\texttt{D}_2\texttt{'}$ (Age' vs. Health).

	Health			ColSum
Age'	Good	Mod	Poor	
20-40	3	1	0	4
40-60	1	0	1	2
	4	1	1	6

Table 4: 2D Histogram from Masked $\texttt{D}_3\texttt{'}$ (Age' vs. Health).

	Health			ColSum
Age'	Good	Mod	Poor	
20:20	2	0	0	2
30:30	0	1	0	1
40:40	0	0	2	2
50:50	0	0	1	1
	2	1	3	6

Masking 3 ($\texttt{D}_3\texttt{'}$). : Age vs. Health

Similarly, we run IPF with the same $\{21, 25, 30, 42, 48, 55\} \mapsto 1$ row sums. A final table emerges consistent with those row sums and the masked column sums from $\texttt{D}_3\texttt{'}$.

6.6.2 2D Histogram Known. This subsection will include a detailed analysis for cases where the 2D histogram of the original table is known. Placeholders for tables and discussions:

Table 5: 2D Histograms of Original Data D.

X	Health: Good	Health: Mod	Health: Poor
X1	–	–	–
X2	–	–	–

6.6.3 No Histogram Known. This subsection will describe scenarios where neither the 1D nor the 2D histogram of the original data is available. Placeholders for related content and discussions will be provided.

Example Reconstructions. TBD: Steps to handle reconstruction without histogram knowledge.

Table 6: Overview of benchmark dataset collection.

Name	ID	#Rows	#Attributes	Label
AirQuality [?]	AQ			
Customer [?]	ICH			
Income [?]	IN			

7 Evaluation

We experimentally evaluate PUMA on real-world datasets with the goal of investigating the

(1) effectiveness of our utility optimizer
(2) accuracy of our IPF-based estimation of joint distributions
(3) impact of presence and absence of marginal distributions
(4) impact of different parameters such as dataset size, number of attributes, and number of masking configurations

Our evaluation shows that [KB#4: write key take away messages from the experiments.]

7.1 Experimental Setup

We implemented our prototype for PUMA in Python. [KB#5: Were any performance optimization packages/libraries were used?]. Our experiments used a TODO: XYZ server with TODO: processor and TODO: XYZ GB RAM.

Datasets. We used TODO: XXX real-world datasets that are summarized in Table 6. The Air Quality data [?] TODO: briefly describe this data. Customer and Hotel Interaction [?] is TODO: describe. Lastly, we used Income dataset [] that ...TODO: describe.. Additionally, we used a synthetic dataset to test the scalability of our our approach with respect to different parameters (see Section **??**).

Summaries. For each of the attributes in our dataset, we also created 1D histograms. TODO: Describe how you created these; are these equidepth/width ?.

Masking Configurations. We implemented a masking configuration generator that for a given dataset can generate different masking configurations. The generator, supports generalization-based masking functions as well as attribute suppression. Each masking function can be parameterized in terms of generalizing granularity. TODO: Describe precisely how this works.

Baselines. We compare our approach to the utility estimator of Mascara [21], a state-of-the-art approach for data disclosure, which also allows determining the best masking configuration for a given dataset. Additionally, we implement a sampling based approach TODO: Describe this.

7.2 Overall Effectiveness

We start our evaluation of PUMA by analyzing its overall effectiveness in determining an optimal masking configuration.

Methodology. For each of our benchmark dataset, we consider the following downstream ML task. A Random Forest, TODO: XX, and TODO: XX. We compare our approach to Mascara and the Sampling-based approach by analyzing the accuracy of these downstream tasks on masked datasets. We considered TODO: XX masking configurations for each dataset. TODO: describe the setup We assume availability of 1D histograms for these approach (same as Mascara)

As a frame of reference, we also compare to the "gold" accuracy. TODO: Describe gold here, i.e., we evaluate the down stream tasks on each of the masked dataset and then pick the one with best accuracy. Stress that is infeasible in practise, yet we compare for completeness

Results. TODO: We need accuracy numbers for

Dataset ML-Task Gold Mascara Sampling-based PUMA +g3 PUMA +MI PUMA +chi-sq

7.3 Impact of Different Data Summaries

In these set of experiments, we evaluate

Methodology. Default setup;

Results. TODO: We need accuracy numbers for

Dataset ML-Task PUMA +g3+1D PUMA +g3+2D PUMA +g3+NoHist

For the same dataset & ML-task combo

Dataset ML-Task PUMA +MI+1D PUMA +MI+2D PUMA +MI+NoHist and

Dataset ML-Task PUMA +Chi-Sq+1D PUMA +Chi-Sq+2D PUMA +Chi-Sq+NoHist

7.4 Impact of Parameters

We now evaluate how dataset size, number of attributes, and number of masking configurations impact the performance of PUMA. We use synthetic dataset as TODO: justify

Impact of #Rows.

Impact of #Attributes.

Impact of #Masking Configurations.

7.5 Quality of IPF-based Reconstruction

In these experiments, we evaluate ...

Setup. Compare joint distribution estimated by our approach and compare it to actual ground truth

Reconstruction From 1D.

Reconstruction in the absence of marginals.

7.6 Framework Efficiency

Impact of #Rows.

Impact of #Attributes.

Impact of #Masking Configurations.

8 Related Work

- [22]
- [8]
- [25]
- [4]
- [23]
- [18]
- [5]
- [24], also include work on l-diversity, t-closeness
- [6]
- [16]

Privacy-preserving data transformations such as masking and anonymization techniques aim to protect sensitive information. However, these transformations can significantly alter statistical relationships, particularly the correlation between features and labels, potentially impacting downstream analyses. To address this, we review relevant literature in two key areas: (1) Data Obfuscation,

which examines anonymization and masking techniques designed to protect privacy, and (2) Data Correlation, which explores methods for measuring statistical dependencies with a specific focus on correlation. Given our goal, quantifying how much different masking configurations alter correlation structures and identifying the configuration with the least correlation change is essential.

8.1 Data Obfuscation

Traditional Anonymization Techniques

Data anonymization is the process of modifying datasets to prevent the identification of individuals while retaining useful information for analysis. Various anonymization techniques have been proposed to balance privacy protection and data utility. One of the most widely used approaches is k-anonymity [24], which ensures that each record is indistinguishable from at least $k-1$ others. However, k-anonymity suffers from vulnerabilities, such as homogeneity attacks and background knowledge attacks, where an adversary can infer sensitive attributes if equivalence classes lack sufficient diversity [17]. To mitigate these risks, l-diversity was introduced, ensuring that each equivalence class contains at least l well-represented values for sensitive attributes. Despite this improvement, l-diversity fails to account for attribute distributions, leading to the development of t-closeness. This method ensures that the distribution of sensitive attributes within an equivalence class remains close to that of the overall dataset [15].

Differential Privacy

Another widely studied technique is differential privacy, which provides strong theoretical guarantees by adding noise into query responses, ensuring that individual records remain indistinguishable [6]. This method is particularly effective for protecting privacy in aggregate statistical analyses. However, its impact on predictive modeling can be substantial because the added noise can distort feature-label correlations, reducing the accuracy of machine learning models. As a result, researchers have explored utility-aware perturbation techniques that attempt to balance privacy and data utility [16, 18].

Structural Anonymization

In addition to attribute-based techniques, structural anonymization methods have been explored to protect relational and graph-based data. Some methods focus on anonymizing bipartite graphs using safe grouping strategies to maintain structural integrity while ensuring privacy [4]. Others attempt to balance efficiency and utility trade-offs to achieve anonymization with minimal information loss [8].

Membership Disclosure Attacks

Even with structural anonymization, datasets remain vulnerable to indirect attacks such as membership disclosure, where adversaries can infer whether an individual is present in an anonymized dataset. The δ-presence model was introduced to mitigate this risk by limiting the probability of an individual's presence being inferred [19]. Alternative techniques focus on disrupting the linkage between quasi-identifiers and sensitive attributes without modifying attribute values [25].

Data Masking

Data masking techniques protect sensitive information by altering original values into modified forms that restrict unauthorized access. One widely used approach, generalization, replaces specific values with broader categories to reduce identifiability while maintaining analytical usability. Another common technique, suppression, removes or hides certain data points entirely, making it particularly effective for highly sensitive attributes where any exposure poses a privacy risk. To introduce controlled randomness, perturbation adds noise or random modifications to the data, ensuring that individual values cannot be precisely reconstructed while preserving overall statistical properties. In addition, tokenization replaces sensitive data with unique identifiers that retain internal relationships within the dataset but prevent direct linkage to the original information. Finally, shuffling disrupts direct associations by randomly reordering values within an attribute while maintaining the overall distribution. These techniques represent some of the widely used approaches to data masking, each offering varying levels of privacy protection depending on the sensitivity of the data and the intended use case.

8.2 Data Correlation

Ensuring that the correlation structure of a dataset remains consistent after applying masking functions is crucial for maintaining data utility. Statistical correlation measures are commonly used to evaluate the impact of anonymization techniques on data dependencies. Categorical dependencies are typically assessed with the Chi-square test for independence. It is a widely used statistical method to determine whether two categorical variables are independent or associated based on their observed and expected frequencies in contingency tables.

Mutual Information is another widely used measure that captures both linear and nonlinear associations by quantifying the shared information between variables. It measures how much knowing one variable reduces uncertainty about another, making it suitable for both numerical and categorical data. More complex relationships can be analyzed using correlation matrices or partial correlation, which capture interdependencies among multiple attributes. A correlation matrix presents a structured representation of pairwise correlation coefficients, indicating the strength of relationships between variables. Additionally, partial correlation quantifies the direct relationship between two variables while controlling for the influence of other factors, effectively isolating true dependencies and reducing the impact of confounding variables.

In addition to these correlation-based methods, error measures such as G1, G2, and are used to assess the extent to which a functional dependency $X \rightarrow Y$ holds in a relational dataset. G1 counts the number of violating tuple pairs, where two tuples agree on X but differ on Y. G2 measures the number of individual violating tuples, where at least one tuple is involved in a violation. G3 represents the minimum number of tuples that must be removed to ensure the dependency holds in the remaining dataset. [11]

Furthermore, information-theoretic measures, such as Kullback-Leibler (KL) Divergence and Jensen-Shannon Divergence (JSD), provide a framework for assessing the differences between probability distributions, enabling the detection of asymmetric and nonlinear dependencies.

These methods are valuable in evaluating the impact of data transformations on statistical relationships, helping ensure that

key dependencies are preserved for analytical integrity. In our framework, we employ G1, G2, and G3 metrics, Mutual Information, and the Chi-Square test to quantify correlation changes before and after masking, enabling the selection of a masking configuration that minimizes correlation change.

9 Conclusion

References

[1] Roberto Battiti. 1994. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks* 5, 4 (1994), 537–550.

[2] Xingguang Chen and Sibo Wang. 2021. Efficient approximate algorithms for empirical entropy and mutual information. In *Proceedings of the 2021 ACM SIGMOD international conference on Management of data*. 274–286.

[3] William G Cochran. 1952. The $\chi 2$ test of goodness of fit. *The Annals of mathematical statistics* (1952), 315–345.

[4] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. 2008. Anonymizing bipartite graph data using safe groupings. *Proc. VLDB Endow.* 1, 1 (Aug. 2008), 833–844. https://doi.org/10.14778/1453856.1453947

[5] Chenyun Dai, Gabriel Ghinita, Elisa Bertino, Ji-Won Byun, and Ninghui Li. 2009. TIAMAT: a tool for interactive analysis of microdata anonymization techniques. *Proceedings of the VLDB Endowment* 2, 2 (2009), 1618–1621.

[6] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.

[7] François Fleuret. 2004. Fast binary feature selection with conditional mutual information. *Journal of Machine learning research* 5, 9 (2004).

[8] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. 2007. Fast data anonymization with low information loss. In *Proceedings of the 33rd international conference on Very large data bases*. 758–769.

[9] Jinjie Huang, Yunze Cai, and Xiaoming Xu. 2007. A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters* 28, 13 (2007), 1825–1844. https://doi.org/10.1016/j.patrec.2007.05.011

[10] Martin Idel. 2016. A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps. *arXiv preprint arXiv:1609.06349* (2016).

[11] Jyrki Kivinen and Heikki Mannila. 1995. Approximate inference of functional dependencies from relations. *Theoretical Computer Science* 149, 1 (1995), 129–149. https://doi.org/10.1016/0304-3975(95)00028-U Fourth International Conference on Database Theory (ICDT '92).

[12] J Kruithof. 1937. Calculation of telephone traffic. *De Ingenieur* 52, 8 (1937), E15–E25.

[13] Nojun Kwak and Chong-Ho Choi. 2002. Input feature selection by mutual information based on Parzen window. *IEEE transactions on pattern analysis and machine intelligence* 24, 12 (2002), 1667–1671.

[14] Marie Le Guilly, Jean-Marc Petit, and Vasile-Marian Scuturici. 2020. Evaluating classification feasibility using functional dependencies. *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLIV: Special Issue on Data Management–Principles, Technologies, and Applications* (2020), 132–159.

[15] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity. *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)* (2007).

[16] Tiancheng Li and Ninghui Li. 2009. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 517–526.

[17] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. 2006. L-diversity: Privacy Beyond k-Anonymity. *Proceedings of the 22nd International Conference on Data Engineering (ICDE)* (2006).

[18] Gerome Miklau. 2022. Negotiating Privacy/Utility Trade-Offs under differential privacy. *Santa Clara, CA* (2022).

[19] M. Ercan Nergiz, Maurizio Atzori, and Christopher W. Clifton. 2007. Hiding the Presence of Individuals from Shared Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 665–676. https://doi.org/10.1145/1247480.1247554

[20] Cláudia Pascoal, M Rosário Oliveira, António Pacheco, and Rui Valadas. 2017. Theoretical evaluation of feature selection methods based on mutual information. *Neurocomputing* 226 (2017), 168–181.

[21] Rudi Poepsel-Lemaitre, Kaustubh Beedkar, and Volker Markl. 2024. Disclosure-Compliant Query Answering. *Proc. ACM Manag. Data* 2, 6, Article 233 (2024), 28 pages.

[22] Vibhor Rastogi, Dan Suciu, and Sungho Hong. 2007. The boundary between privacy and utility in data publishing. In *Proceedings of the 33rd international conference on Very large data bases*. 531–542.

[23] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic data–anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*. 1451–1468.

[24] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems* 10, 05 (2002), 557–570.

[25] Manolis Terrovitis, John Liagouris, Nikos Mamoulis, and Spiros Skiadopoulos. 2012. Privacy preservation by disassociation. *VLDB* (2012).

[26] Jorge R Vergara and Pablo A Estévez. 2014. A review of feature selection methods based on mutual information. *Neural computing and applications* 24 (2014), 175–186.