PeGS: Perturbed Gibbs Samplers that Generate Privacy-Compliant Synthetic Data

Yubin Park and Joydeep Ghosh

Department of Electrical and Computer Engineering

The University of Texas at Austin, USA

Abstract. This paper proposes a categorical data synthesizer algorithm that guarantees a quantifiable disclosure risk. Our algorithm, named Perturbed Gibbs Sampler (PeGS), can handle high-dimensional categorical data that are intractable if represented as contingency tables. PeGS involves three intuitive steps: 1) disintegration, 2) noise injection, and 3) synthesis. We first disintegrate the original data into building blocks that (approximately) capture essential statistical characteristics of the original data. This process is efficiently implemented using feature hashing and non-parametric distribution approximation. In the next step, an optimal amount of noise is injected into the estimated statistical building blocks to guarantee differential privacy or l-diversity. Finally, synthetic samples are drawn using a Gibbs sampler approach. California Patient Discharge data are used to demonstrate statistical properties of the proposed synthetic methodology. Marginal and conditional distributions as well as regression coefficients obtained from the synthesized data are compared to those obtained from the original data. Intruder scenarios are simulated to evaluate disclosure risks of the synthesized data from multiple angles. Limitations and extensions of the proposed algorithm are also discussed.

1 Introduction

Public use data, which are often released by government and other data collecting agencies, typically need to satisfy two competing objectives: maintaining relevant statistical properties of the original data and protecting privacy of individuals. To address these two goals, various statistical disclosure limitation techniques have been developed (Willenborg and de Waal, 2001). Some popular disclosure techniques are data swapping (Dalenius and Reiss, 1978; Fienberg and McIntyre, 2005), top-coding, feature generalization such as k-anonymity (Sweeney, 2002) or l-diversity (Machanavajjhala et al., 2007), and additive random noise with measurement error models (Fuller, 1993). Each method has distinct utility and risk aspects. In practice, a disclosure limitation technique is carefully chosen by domain experts and statisticians. Sometimes, multiple techniques are mixed and applied to a single dataset to achieve better privacy protection before being released to the public (Centers for Medicare and Medicaid Services, 2013). Such public use datasets have served as valuable information sources for decision makings in economics, healthcare, and business analytics.

The generation of synthetic data (Rubin, 1993) is an alternative approach to data transforming disclosure techniques. For example, multiple imputation, which was originally developed to impute missing values in survey responses (Rubin, 1987), can also be used to generate either partially or fully synthetic data. As synthetic data preserves the structure and resolution of the original data, preprocessing steps and analytical procedures on synthetic data can be effortlessly transferred to the original data. This aspect has contributed to popular adoption of synthetic data in diverse research areas. Thus far, there has been notable progress on valid inferences using synthetic data and extensions to different applications: Abowd and Woodcock (2001) synthesized a French longitudinal linked database, and Raghunathan et al. (2003) provided general methods for obtaining valid inferences using multiply imputed data. Beyond typically used generalized linear models and Markov Chain Monte Carlo simulation methods, decision trees models, such as CART and Random forests, can also be used as imputation models in multiple imputation (Reiter, 2005b; Caiola and Reiter, 2010). Some illustrative empirical studies have used U.S. census data (Drechsler and Reiter, 2010), German business database (Reiter and Drechsler, 2010), and U.S. American Community Survey (Sakshaug and Raghunathan, 2011). A very different approach to imputing missing values in binary or binarized datasets can be taken using association rule mining. Vreenken and Siebes (2008) used the minimum description length principle to develop a set of heuristics that are used to (approximately) represent the dataset in terms of a concise set of frequent itemsets. These rules can be then used to impute missing values, and in principle could also be used to generate synthetic data. However, their current work does not quantify the privacy afforded by the compressed data or use a given privacy criterion to determine the derived itemsets.

The two competing requirements for public use data similarly apply to synthetic data disclosure. Synthetic data need to be accurate enough to answer relevant statistical queries without revealing private information to third parties. Statistical properties of synthetic data are primarily determined by imputation models (Reiter, 2005a), and models that are too accurate tend to leak private information (Abowd and Vilhuber, 2008).

The balance between accuracy and privacy can be addressed by using cryptographic privacy measures such as ϵ-differential privacy (Dwork, 2006). However, several attempts to achieve such strong privacy guarantees have shown to be impractical to implement. For example, Barak et al. (2007) showed that it is possible to release contingency tables under the differential privacy regime using Fourier transform and additive Laplace noise. However, this proposed release mechanism was later criticized for being too conservative and disrupting statistical properties of the original data (Yang et al., 2012; Charest, 2012). On the other hand, Soria-Cormas and Drechsler (2013) claimed that ϵ-differential privacy can be a useful privacy measure when disclosing a large size of data with a limited number of variables. For example, differentially private synthetic data have been demonstrated using the Census Bureau's OnTheMap data that consists of approximately one million records with two variables (Machanavajjhala et al., 2008).

In this paper, we propose a *practical* multi-dimensional categorical data synthesizer that satisfies ϵ-differential privacy. The proposed synthesizer can handle multi-dimensional data that are not practical to be represented as contingency tables. We demonstrate our algorithm using a subset of California Patient Discharge data, and generate multiple synthetic discharge datasets. Although ϵ-differential privacy is extensively used in our algorithm analyses, we note that ϵ-differential privacy is one of many descriptive measures for disclosure risks. Differential privacy is a measure for functions, not for data (Fienberg et al., 2010), and this measure can be overly pessimistic for data-specific applications. Thus, we also evaluate disclosure risks of the proposed algorithms using the population uniqueness of synthetic records (Dale and Elliot, 2001) and indirect-matching probabilistic disclosure risks (Duncan and Lambert, 1986). To measure the statistical similarities between synthetic and the original data, we compare 1) marginal and 2) conditional distributions, and 3) regression coefficients of predictive models that are estimated from the original and syn-

Table 1: Synthesizer algorithms discussed in this paper.

Name	Abbreviated Model Eq.	Parameters
Contingency table	$\Pr_{\mathcal{D}}(\mathbf{x})$	non-parametric
Marginal Bayesian Bootstrap	$\prod_i^M \Pr_{\mathcal{D}}(x_i)$	non-parametric
Multiple imputation	$\prod_i^M \Pr_{\hat{\mathbf{w}}}(x_i \mid \mathbf{x}_{-i})$	$\hat{\mathbf{w}}$: model parameter
Perturbed Gibbs Sampler	$\prod_i^M \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$	α: privacy parameter
Block PeGS with Reset	$\prod_b^B \prod_i^M \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$	B: sample block size

thesized data.

There are two brute-force approaches to generating synthetic categorical data. As statistical properties of categorical data are fully captured in contingency tables, in theory, a synthetic sample $\mathbf{x}$ can be drawn directly from an M-way full contingency table $\Pr_{\mathcal{D}}(\mathbf{x})$, where M is the total number of features. For data with a small number of features, this contingency table can be estimated by either direct counting or log-linear models (Winkler, 2003, 2010). However, this strategy does not scale for high-dimensional datasets. As we will see in Section 4, our experiment dataset has 13 features and their possible feature combinations are approximately 2.6 trillion. More importantly, sampling from an exact distribution may reveal too much detail about the original data, thus this is not a privacy-safe disclosure method. On the other extreme, one may model the joint distribution as a product of univariate marginal distributions. Although this approach can easily achieve differential privacy (McClure and Reiter, 2012), the synthetic data loses critical joint distributional information about the original data.

The proposed algorithm generates *realistic but not real* synthetic samples by calibrating a privacy parameter α. In addition, the exponentially number of cells in a contingency table is avoided by using chained equations and feature hashing (Weinberger et al., 2009) as follows:

$$\begin{aligned} &\text{for } i \text{ in } 1:M \\ &\quad x_i \sim \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i})) \end{aligned} \tag{1}$$

where $\mathbf{x}_{-i}$ is a feature vector except for the ith feature and $h(\mathbf{x}_{-i})$ represents a hashed feature vector. $\Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$ is the compressed and perturbed conditional distribution of the ith feature and M is the total number of features. The joint probability distribution is represented as M conditional distributions. Note that the conditional distribution in Equation (1) is not exact. The full condition $\mathbf{x}_{-i}$ is compressed using a hash function $h(\mathbf{x}_{-i})$ and perturbed by a privacy parameter α. Ignoring these two additional components i.e. $h(\mathbf{x}_{-i})$ and α, if the probability is modeled using generalized linear models, then the proposed algorithm is the same as a multiple imputation algorithm for fully synthetic data. The proposed synthesizer is named as Perturbed Gibbs Sampler (PeGS). This process is somewhat analogous to multivariate imputation by chained equations (also known as sequential regression multiple imputation) (Raghunathan et al., 2001; van Buuren and Groothuis-Oudshoorn, 2011). In Section 3.4, we will show that this synthesis cycle can also be recursively applied multiple times i.e. $\mathbf{s} = \text{PeGS}(\text{PeGS}(\ldots\text{PeGS}(\mathbf{x})))$. This recursive synthesis will be shown to be very effective in our block sampling algorithm.

Table 1 summarizes the synthesizer models that are described in this paper. More details on this list and privacy guarantees for both one iteration and multiple iterations are described in Section 3.

The objective of PeGS is to generate a single realistic synthetic dataset that adheres to rigorous privacy metrics, balancing the trade-off between utility and risk in a flexible and effective manner. This is substantially different from the goal of multiple imputation, which primarily focuses on improving the analytical validity of missing data imputation and not on privacy vs. utility trade-off. The name *multiple imputation* refers to the fact that multiple imputed datasets are released to alleviate imputation uncertainty. Disclosing multiple datasets can be helpful for numerous statistical analyses, but at the same time, the improved accuracy may lead to an unexpected privacy breach. Consider a statistical database that provides a synthetic data row per query. To obtain N rows, intuitively, we need N queries. According to the sequential composition rule of differential privacy (McSherry, 2009), the privacy risk for N queries is N times greater than the risk of one query. Similarly, releasing K imputed datasets can be K times more risky than releasing a single dataset.

The target use case of our synthetic data is also quite different from traditional uses of synthetic data. Our algorithms are primarily designed to protect the privacy of the original data, and then, within the privacy constraint, to maximize the statistical validity and utility of synthetic data. Such synthetic data can be useful when providing a single "realistic" dataset to third party data scientists so that they can explore and develop innovative data applications. As an illustrative example, Centers for Medicare & Medicaid Services recently released synthetic public use files[1], saying that:

> *... Although the DE-SynPUF has very limited inferential research value to draw conclusions about Medicare beneficiaries due to the synthetic processes used to create the file, the Medicare DE-SynPUF does increase access to a realistic Medicare claims data file in a timely and less expensive manner to spur the innovation necessary to achieve the goals of better care for beneficiaries and improve the health of the population. ...*

The users of our synthetic data can be from a wide range of disciplines such as statistics, computer science, and healthcare policy studies. Providing synthetic data is much more than just providing the data schema. Users can write scripts and codes for exploring and extracting information using the "realistic" synthetic data, and deliver their applications to the data owner to check the validity of their claims.

The rest of this paper is organized as follows: In Section 2, we cover the basics of privacy measures and synthetic data. In Section 3, the details of the PeGS algorithms are illustrated, and the privacy guarantees of the proposed algorithms are derived. We demonstrate our algorithms using California Patient Discharge dataset in Section 4. Finally, we discuss the limitation of the proposed methods and future extensions in Section 5.

2 Background

In this section, we overview two bodies of related work: privacy measures and approaches to synthetic data generation.

[1] http://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/SynPUFs/DE_Syn_PUF.html

2.1 Privacy Measures

Privacy is an abstract concept, and it can be defined and quantified in many different ways. We describe two privacy measures that are popular in computer science, ϵ-differential privacy and l-diversity. These two measures will be also used in our algorithm to quantify the privacy risks of synthetic data.

2.1.1 Differential Privacy

Differential privacy (Dwork, 2006) is a mathematical measure of privacy that quantifies disclosure risks of statistical functions. To satisfy ϵ-differential privacy, the inclusion or exclusion of any particular record in data cannot affect the outcome of functions by much. Specifically, a randomized function $f : \mathcal{D} \to f(\mathcal{D})$ provides ϵ-differential privacy, if it satisfies:

$$\frac{\Pr(f(\mathcal{D}_1) \in \mathcal{S})}{\Pr(f(\mathcal{D}_2) \in \mathcal{S})} \leq \exp(\epsilon)$$

for all possible $\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{D}$ where $\mathcal{D}_1$ and $\mathcal{D}_2$ differ by at most one element, and $\forall \mathcal{S} \in \text{Range}(f(\mathcal{D}))$. For a synthetic sample, this definition can be interpreted as follows (McClure and Reiter, 2012):

$$\frac{\Pr_{\mathcal{D}_1}(\mathbf{x})}{\Pr_{\mathcal{D}_2}(\mathbf{x})} \leq \exp(\epsilon) \tag{2}$$

where $\mathbf{x}$ represents a random sample from synthesizers. In other words, a data synthesizer $\Pr_{\mathcal{D}}(\mathbf{x})$ is ϵ-differentially private, if the probabilities of generating $\mathbf{x}$ from $\mathcal{D}_1$ and $\mathcal{D}_2$ are indistinguishable to the extent of $\exp(\epsilon)$.

Several mechanisms have been developed to achieve differential privacy. For numeric outputs, the most popular technique is to add Laplace noise with mean 0 and scale $\Delta f/\epsilon$ where Δf is the L_1 sensitivity of function f. Exponential mechanism (McSherry and Talwar, 2007) is a general differential privacy mechanism that can be applied to non-numeric outputs. For categorical data, Dirichlet prior can be used as a noise mechanism to achieve differential privacy (Machanavajjhala et al., 2008; McClure and Reiter, 2012).

2.1.2 l-diversity

A certain combination of features can identify an individual from an anonymized dataset, even if personal identifiers, such as driver license number and social security number, are removed from a dataset. Such threats are commonly prevented by generalizing or suppressing features; for example, ZIP codes with small population are replaced by corresponding county names (generalization), or can be replaced by * (suppression). Sweeney (2002) proposed a privacy definition for measuring the degree of such feature generalization and suppression, k-anonymity. To adhere the k-anonymity principle, each row in a dataset should be indistinguishable with at least $k-1$ other rows.

The definition of k-anonymity, however, does not include two important aspects of data privacy: feature diversity and attackers' background knowledge. Machanavajjhala (2007) illustrated two potential threats to a k-anonymized dataset, then proposed a new privacy criterion, l-diversity. The definition of l-diversity states that the diversity of sensitive features should be kept within a block of samples. There are several ways of achieving l-diversity; in this paper, we use Entropy l-diversity. A dataset is Entropy l-diverse if

$$-\sum_{x_i} \Pr(x_i \mid \mathbf{x}_{-i}) \log \Pr(x_i \mid \mathbf{x}_{-i}) \geq \log l \tag{3}$$

where $1 \leq l$. This definition originally applies to a dataset with feature generalization or suppression. For a synthetic sample, Park et al. (2013b) suggested an analogous definition of l-diversity: A synthetic dataset is synthetically l-diverse if a synthetic sample x_i is drawn from a distribution that satisfies l-diversity.

2.2 Synthetic Data Generation

Synthetic data can be used in a wide range of applications: system performance evaluation, statistical model validation, and privacy-preserving data publication. In this paper, we are primarily interested in producing privacy-preserving synthetic data. We start by describing general approaches to generate synthetic data. In the Bayesian statistics literature, multiple imputation is widely used to generate synthetic data. We illustrate the basic concepts of multiple imputation, and discuss some limitations of the multiple imputation approach in our setting.

2.2.1 General Approaches to Synthetic Data

Synthetic data generation typically involves two steps: 1) statistical modeling of the original data, and 2) sampling from the obtained model. In the modeling step, one can apply a wide range of statistical models, from a simple linear regression model to advanced Markov Chain Monte Carlo sampling methods. Disclosure risks of synthetic samples are traditionally analyzed after the sampling step, but recently several researchers have attempted to merge privacy metrics in the modeling and sampling steps. Depending on the application, synthetic data can replace either the entire original data (Rubin, 1993), or specific columns or values that bear high disclosure risks i.e. partially synthetic data (Little, 1993). The notion of fully synthetic data in the multiple imputation literature is slightly different from our notion: this aspect will be detailed in Section 2.2.2. Figure 1 shows various categories of synthetic data. Note that, in this paper, fully synthetic data refers to completely synthetic data with no original records.

The quality of a synthetic dataset is mainly determined by the quality of the statistical model used. Lombardo and Moniz (2008) proposed generating synthetic medical records for outbreak studies. They suggested using both domain knowledge and actual data. The underlying dynamics of the original data is modeled through a set of sub-models such as exposure model, infection model, disease model, and behavior model. Buczak et al. (2010) demonstrated a pilot study for generating synthetic medical records. In their pilot study, the synthetic data were generated through three steps: 1) patient information generation, 2) similar patients clustering, and 3) adapting care models to synthesized patients. As can be seen, these approaches require a considerable amount of domain knowledge and non-automated processes. Furthermore, no privacy measure was incorporated in the synthesizing processes.

Machanavajjhala et al. (2008) generated a differentially private synthetic dataset for commuting pattern studies. They derived an appropriate amount of Laplace smoothing parameters to guarantee ϵ-differential privacy. A subset from the Census Bureau's OnTheMap microdata was used in their study. However, the demonstrated dataset had only three columns (id, origin block, destination block), and the suggested algorithm was specific to the application. Barak et al. (2007) suggested a differentially private release mechanism for contingency tables. Note that releasing a contingency table is different from releasing a synthetic dataset, but one always can sample a synthetic sample from the released contingency table. Their approach used Fourier transform and Linear programming to guarantee differential privacy. Although the theoretical results are solid, experimental results using real datasets show that the suggested differential privacy mechanism is not practical for data mining purposes (Yang et al., 2012).

Synthetic data generation has also been used for privacy-aware distributed data mining scenarios. In the prototypical approach (Merugu and Ghosh, 2003, 2005), parametric models are separately learnt on individual (local) databases. Then, only the model parameters are transmitted to a trusted central site from each local database, instead of the raw data, to address privacy concerns. At the central site, the parameters received are used to generate synthetic data that (approximately) represents the union of the different databases. Finally, a global model is learnt using such data. However, privacy constraints such as l-diversity are not directly built into the data generation or modeling process, as it is in this paper. Also, the goal is to attain a global statistical model under sharing constraints, rather than create a synthetic dataset for public release.

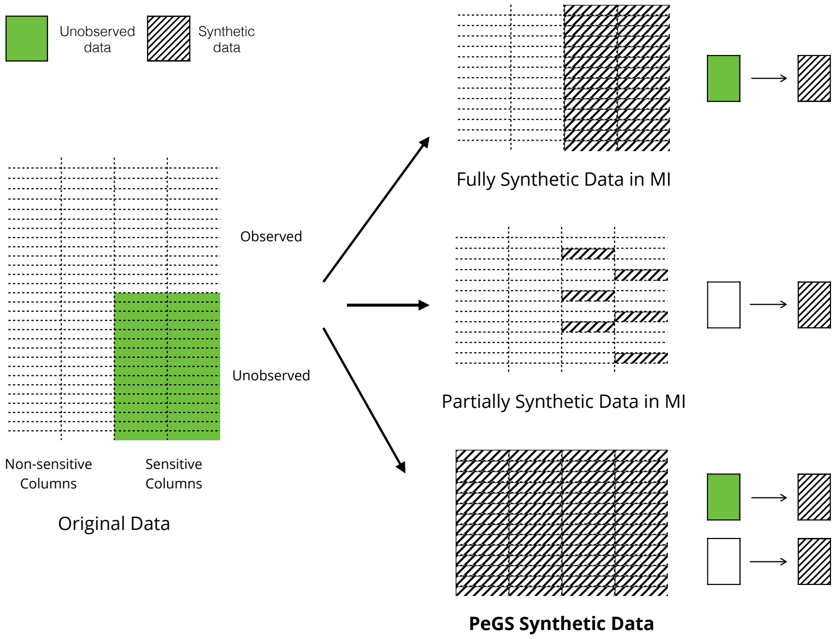


Figure 1: Three different notions of synthetic data. In multiple imputation, only sensitive columns are synthesized. To create a fully synthetic dataset, one replaces the unobserved sensitive values with synthetic values (Drechsler et al., 2008). A partially synthetic dataset only replaces the observed sensitive values that bear high disclosure risks. Our definition of fully synthetic data refers to "completely" synthetic data with no original records regardless of sensitive or non-sensitive columns.

2.2.2 Synthetic Data using Multiple Imputation

Multiple imputation was originally developed to impute missing values in survey responses (Rubin, 1987), and it was later applied to generate synthetic data. Let us start from the missing value imputation setting. Consider a survey with two variables x and z, $\mathcal{D} = \{(x, z)\}$, where some of the x responses are missing. Let x_{obs} be the observed subset of x. The unobserved responses are imputed using samples from a predictive posterior model as follows:

$$x \sim \Pr(x \mid x_{\text{obs}}, z)$$

Note that the posterior can be modeled using the observed subset, and often obtained using generalized linear models, Bayesian Bootstrapping methods, or Markov Chain Monte Carlo simulations (Schafer, 1997; Reiter, 2005a). For example, an R package for multivariate imputation for chained equation (van Buuren and Groothuis-Oudshoorn, 2011) provides nine different imputation models including predictive mean matching, Bayesian linear regression, Linear regression, Unconditional mean imputation, etc. Generating fully synthetic data is straightforward from this framework[2]. First, z is drawn from $\Pr(z)$, then x is drawn from the predictive posterior distribution. Typically, this entire process is repeated independently K times to obtain K different synthetic datasets.

Raghunathan et al. (2003) showed that valid inferences can be obtained from multiply imputed synthetic data. Let Q be a function of (x, z). For example, Q may represent the population mean of (x, z) or the population regression coefficients of x on z. Let q_i and v_i be the estimate of Q and its variance obtained from the ith synthetic dataset. Then, valid inferences on Q can be obtained as follows:

$$\bar{q}_K = \sum_{i=1}^{K} q_i / K$$

$$T_s = (1 + \frac{1}{K}) b_K - \bar{v}_K$$

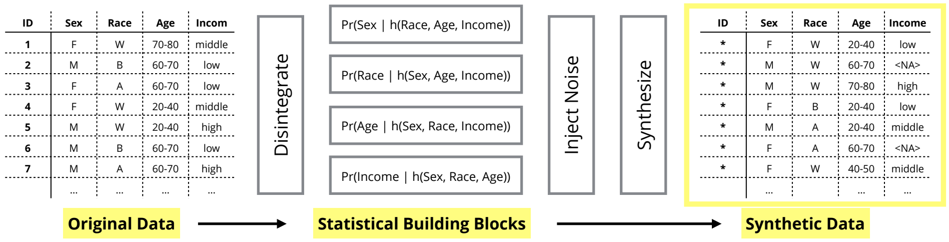
where $b_K = \sum_{i=1}^{K} (q_i - \bar{q}_K)^2 / (K - 1)$ and $\bar{v}_K = \sum_{i=1}^{K} v_i / K$. These two quantities $\bar{q}_K$ and T_s estimate the original Q and the variance from sampling.

The disclosure risks of multiply imputed synthetic datasets are typically measured after synthetic datasets are generated i.e. using *post hoc* risk analysis. Multiple imputation is a general imputation methodology, and the choice of posterior model is usually up to statisticians. This flexibility makes it difficult to apply and analyze rigorous privacy measures, such as differential privacy and l-diversity, in a unified framework. In contrast, we derive the relationship between the amount of Laplace smoothing and privacy measures (ϵ in differential privacy and l in l-diversity) by using a simple non-parametric model. Our algorithm directly incorporates these privacy measures in the synthesizing process, guaranteeing the desired level of privacy for synthetic data.

3 Perturbed Gibbs Sampler

In this section, we propose the Perturbed Gibbs Sampler (PeGS) for categorical synthetic data. We first overview the algorithm, then describe its three main components: feature hashing, statistical building blocks, and noise mechanism. Next, we illustrate how the

[2] To see the difference between partially and fully synthetic datasets, see (Drechsler et al., 2008)



ID	Sex	Race	Age	Incom
1	F	W	70-80	middle
2	M	B	60-70	low
3	F	A	60-70	low
4	F	W	20-40	middle
5	M	W	20-40	high
6	M	B	60-70	low
7	M	A	60-70	high
	...	...	...	...

ID	Sex	Race	Age	Income
*	F	W	20-40	low
*	M	W	60-70	<NA>
*	M	W	70-80	high
*	F	B	20-40	low
*	M	A	20-40	middle
*	F	A	60-70	<NA>
*	F	W	40-50	middle
	...		...	...

Disintegrate
Pr(Sex | h(Race, Age, Income))
Pr(Race | h(Sex, Age, Income))
Pr(Age | h(Sex, Race, Income))
Pr(Income | h(Sex, Race, Age))
Inject Noise
Synthesize
Original Data
Statistical Building Blocks
Synthetic Data

Figure 2: PeGS Process Diagram for a four feature dataset: {(Sex, Age, Race, Income)}. Four types of conditional distributions are estimated from the original data, then uniform Dirichlet priors are used to perturb the conditional distributions. Synthetic samples are drawn by iterating over the statistical building blocks.

PeGS algorithm can be efficiently extended to draw a block of random samples. Finally, we show that multiple imputation can be similarly extended to satisfy differential privacy, which will be used as our baseline model in Section 4.

3.1 Algorithm Overview

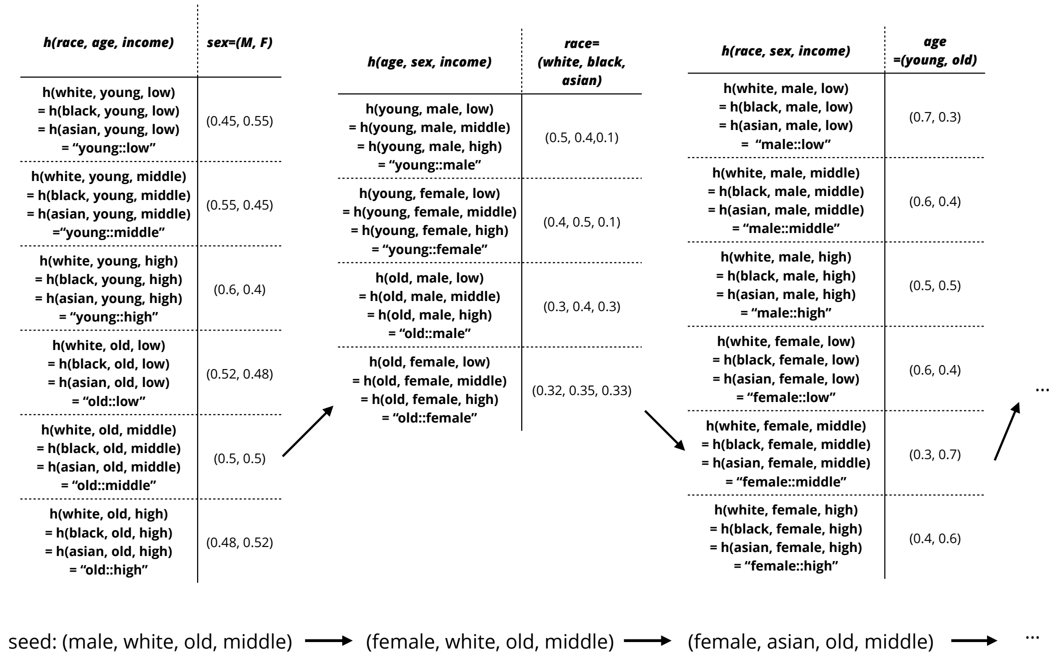
Perturbed Gibbs Sampler (PeGS) is a categorical data synthesizer that consists of three main steps:

1. *Disintegrate*: In this step, the original data $\mathcal{D}$ is disintegrated into statistical building blocks i.e. $\Pr_{\mathcal{D}}(x_i \mid h(\mathbf{x}_{-i}))$ where h is a suitable hash function. These compressed conditional distributions are estimated by counting the corresponding occurrences in the original data.

2. *Inject Noise*: For a specified privacy parameter α, the statistical building blocks are modified to satisfy differential privacy or l-diversity, $\Pr_{\mathcal{D}}(x_i \mid h(\mathbf{x}_{-i})) \rightarrow \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$.

3. *Synthesize*: We first pick a random seed from a predefined pool; this can be regarded as a query to our model. The seed sample is transformed to a synthetic sample by iteratively sampling each feature from the statistical building blocks, $x_i \sim \Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$.

Figure 2 visualizes the overall sequential steps of the PeGS algorithm. Figure 3 illustrates the synthesis step. Three components are essential in the PeGS algorithm: feature hashing, statistical building blocks, and perturbation. The number of possible conditions is exponential with respect to the number of features, Therefore, feature hashing is used to compress the number of the possible conditions $\mathbf{x}_{-i}$. Statistical building blocks are built based on this feature hashing, which are essentially multiple hash-tables describing compressed conditional distributions. They serve a key role when we try to sample a block of synthetic examples. Perturbation is required to guarantee the differential privacy. Without perturbation, synthetic samples may reveal too much about the original data.



h(race, age, income)	*sex=(M, F)*
h(white, young, low) = h(black, young, low) = h(asian, young, low) = "young::low"	(0.45, 0.55)
h(white, young, middle) = h(black, young, middle) = h(asian, young, middle) ="young::middle"	(0.55, 0.45)
h(white, young, high) = h(black, young, high) = h(asian, young, high) = "young::high"	(0.6, 0.4)
h(white, old, low) = h(black, old, low) = h(asian, old, low) = "old::low"	(0.52, 0.48)
h(white, old, middle) = h(black, old, middle) = h(asian, old, middle) = "old::middle"	(0.5, 0.5)
h(white, old, high) = h(black, old, high) = h(asian, old, high) = "old::high"	(0.48, 0.52)

h(age, sex, income)	*race= (white, black, asian)*
h(young, male, low) = h(young, male, middle) = h(young, male, high) = "young::male"	(0.5, 0.4,0.1)
h(young, female, low) = h(young, female, middle) = h(young, female, high) = "young::female"	(0.4, 0.5, 0.1)
h(old, male, low) = h(old, male, middle) = h(old, male, high) = "old::male"	(0.3, 0.4, 0.3)
h(old, female, low) = h(old, female, middle) = h(old, female, high) = "old::female"	(0.32, 0.35, 0.33)

h(race, sex, income)	*age =(young, old)*
h(white, male, low) = h(black, male, low) = h(asian, male, low) = "male::low"	(0.7, 0.3)
h(white, male, middle) = h(black, male, middle) = h(asian, male, middle) = "male::middle"	(0.6, 0.4)
h(white, male, high) = h(black, male, high) = h(asian, male, high) = "male::high"	(0.5, 0.5)
h(white, female, low) = h(black, female, low) = h(asian, female, low) = "female::low"	(0.6, 0.4)
h(white, female, middle) = h(black, female, middle) = h(asian, female, middle) = "female::middle"	(0.3, 0.7)
h(white, female, high) = h(black, female, high) = h(asian, female, high) = "female::high"	(0.4, 0.6)

...
seed: (male, white, old, middle) ⟶ (female, white, old, middle) ⟶ (female, asian, old, middle) ⟶ ...

Figure 3: Synthesis Steps in PeGS. Three tables represent the statistical building blocks of the example in Figure 2. In the disintegration step, these three statistical building blocks are stored. In the noise injection step, the probability vectors of the tables are perturbed. In the synthesis step, a new sample is generated by iteratively sampling over the tables.

3.2 Feature Hashing

The hash function $h(\mathbf{x}_{-i})$ in PeGS maps a feature vector to an integer key, where the range of the hash key is much smaller than 2^M (exponential in the number of features). In essence, our purpose is to design a hash function that exhibits good compression while maintaining the statistical properties of data. Such a hash function has been deeply investigated in the machine learning literature for compressing high-dimensional feature spaces. This technique is sometimes known as the hashing trick (Weinberger et al., 2009). For extremely high-dimensional, sparse, and unstructured data such as natural language texts, Locality Sensitive Hashing (Indyk and Motwani, 1998) and min-hashing (Gionis et al., 1999) can be good candidates for the PeGS hash function.

In this paper, we use a simpler approach to compress the feature space, as we are using lower dimensional data. We select the m variables that have the most mutual information with x_i to form the hash key, and ignore the other variables. Thus:

$$\underbrace{x_{o(1)} :: \ldots :: x_{o(m)}}_{\text{trivial hash}} \longrightarrow \underbrace{1...H}_{\text{Hash key}}$$

where $H \ll 2^M$ and $x_{o(j)}$ represents the feature with the jth highest mutual information with x_i. Let C_i be the number of categories for x_i, and $C_{\max} = \max_i C_i$. The key space of this simple hash function is upper bounded by $(C_{\max})^m \ll \prod_i C_i$.

Figure 3 illustrates the basic idea of feature hashing. The left and right columns of the conditional tables represent hashed features $h(\mathbf{x}_{-i})$ and smoothed probability estimates $\Pr_{\mathcal{D},\alpha}(x_i \mid h(\mathbf{x}_{-i}))$, respectively. The first table uses `race`, `age`, `income` as conditional variables, and `sex` as a target variable. For illustrative purposes, we use $m = 2$, and assume that the target variable is closely related to the `age` and `income` variables. In other words, the `race` variable is ignored while constructing the conditional table i.e. $h(\text{white}, \text{young}, \text{low})$, $h(\text{black}, \text{young}, \text{low})$, and $h(\text{asian}, \text{young}, \text{low})$ belong to the same bin. The smoothed probability distributions are estimated based on a subset of samples that have the same hash key. Synthetic samples are sequentially drawn from these based and smoothed estimates, as Figure 3 shows. The details about smoothing and sampling processes will be discussed in Section 3.3.

The compressed conditional distribution $\Pr(x_i \mid h(\mathbf{x}_{-i}))$, which is basically a occurrence count hash-table for a given hash key, can now be stored in either memory or disk. There are several advantages of using this compressed conditional distribution over parametric modeling. First, the process of building statistical building blocks does not involve complicated statistical procedures such as parameter estimation and model selection. Second, the resulting statistical building blocks are robust to overfitting. Overfitting may occur when there are not enough samples in a table entry. Hashing reduces the number of table cells and smoothes out the estimated probability vector. Finally, this simple table representation is intuitive, and the process is easily extensible. This aspect is critical in our efficient block sampling scheme, which will be illustrated in Section 3.4.

Note that our feature hashing is different from multinomial models in which certain main effects and interactions are set to zero. The key difference is that our process is iterative. As an illustrative example, suppose that we have three features x_1, x_2, and x_3, and we use $m = 1$. Let us assume that x_1 depends on x_2, x_2 depends on x_3, and x_3 depends on x_1. Then x_1 and x_3 get coupled and are not independent. Thus the synthetic process does not translate to simple multinomial models.

We now provide a brief guideline for determining the value of m. As a rule of thumb, we suggest that each cell approximately contains at least 30 data points to estimate probabilities. There are other physical constraints on the value of m such as the size of memory and hard disk. For example, if we want to minimize the access to hard disk, m should also satisfy $2^m <$ Memory Size. But, if m is too small, then this hash function effectively imposes an unrealistic conditional independence assumption. Therefore, the value of m should be carefully determined considering these listed aspects.

3.3 Perturbed Conditional Distribution

To satisfy the differential privacy, a certain amount of noise should be injected to the compressed conditional distributions. The form of noise may depend on applications and privacy measures. For example, noise can be added to maximize entropy (Polettini, 2003) or to satisfy l-diversity (Park et al., 2013a,b). In this paper, we use the Dirichlet prior perturbation to smooth out raw count based estimators to satisfy differential privacy and l-diversity. Specifically, α virtual samples are added to each category of the variable x_i, when the conditional distribution $\Pr_\alpha(x_i \mid h(\mathbf{x}_{-i}))$ is estimated. The amount α is a privacy parameter that controls the degrees of differential privacy and l-diversity. To be more precise, our differentially private perturbation requires a single value of α, while our l-diverse perturbation needs different α values for each hashed condition $h(\mathbf{x}_{-i})$ i.e. $\alpha_{h(\mathbf{x}_{-i})}$. For analytical simplicity, we assume α virtual samples, $\alpha_{h(\mathbf{x}_{-i})}$ virtual samples for l-diversity, are uniformly added to all the categories of the variable x_i (see Equation 6). In practice, different amounts of virtual samples can be added to different categories of the variable x_i; for example, α can be proportional to the corresponding marginal distribution i.e. $\alpha_j \propto \Pr(x_i = j)$.

We first derive the probability of sampling $\mathbf{x}$ from the PeGS algorithm. From a random seed sample $\mathbf{s}$ (or a query), the probability of synthesizing $\mathbf{x}$ is factorized as follows:

$$\Pr_{\mathcal{D}_1,\alpha}(\mathbf{x} \mid \mathbf{s}) = \prod_{i=1}^{M} \Pr_{\mathcal{D}_1,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M})) \tag{4}$$

where $x_{1:0}$ and $s_{(M+1):M}$ are just null values. For another dataset $\mathcal{D}_2$ that differs by at most one element, the probability of sampling $\mathbf{x}$ can be similarly derived.

For differential privacy (see Equation 2), the ratio between two quantities should satisfy the following relation:

$$\frac{\Pr_{\mathcal{D}_1,\alpha}(\mathbf{x} \mid \mathbf{s})}{\Pr_{\mathcal{D}_2,\alpha}(\mathbf{x} \mid \mathbf{s})} = \frac{\prod_{i=1}^{M} \Pr_{\mathcal{D}_1,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))}{\prod_{i=1}^{M} \Pr_{\mathcal{D}_2,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))} \leq \exp(\epsilon) \tag{5}$$

Let us focus on the ith component as follows:

$$\Pr_{\mathcal{D}_1,\alpha}(x_i = j \mid h(\mathbf{x}_{-i})) = \frac{n_{ij} + \alpha}{N_{h(\mathbf{x}_{-i})} + C_i\alpha} \tag{6}$$

$$N_{h(\mathbf{x}_{-i})} = \sum_{\mathbf{x}'_{-i}} \mathbb{1}(h(\mathbf{x}'_{-i}) = h(\mathbf{x}_{-i})) \tag{7}$$

where $N_{h(\mathbf{x}_{-i})}$ is the total number of rows that have the same hash key as $h(\mathbf{x}_{-i})$ and n_{ij} is the count of the jth category i.e. $x_i = j$ within the $N_{h(\mathbf{x}_{-i})}$ samples. In other words, the probability of sampling the jth category is proportional to the number of the original samples that have the jth category. The privacy parameter α acts as a uniform Dirichlet prior on this raw multinomial count estimate.

The value of α depends on the privacy criterion. We study two cases: differential privacy and l-diversity.

A. Differential Privacy. The two datasets for defining differential privacy $\mathcal{D}_1$ and $\mathcal{D}_2$ have at most one different row. Without loss of generality, let us assume that $\mathcal{D}_1$ has one more row than $\mathcal{D}_2$ i.e. $\mathcal{D}_1 = \mathcal{D}_2 \cup \mathbf{x}^d$. Except for the entry with hash keys $\{h(\mathbf{x}^d_{-i})\}_{i=1}^{M}$, the other entries of the two hash tables from $\mathcal{D}_1$ and $\mathcal{D}_2$ are identical; only one entry of the hash table is different. For the different entries of the hash tables, there are two possibilities:

$$\begin{aligned} \text{if} \quad x_i^d \neq j, \qquad & \Pr_{\mathcal{D}_1,\alpha}(x_i^d = j \mid h(\mathbf{x}^d_{-i})) = \frac{n_{ij} + \alpha}{N_{h(\mathbf{x}_{-i})} + 1 + C_i\alpha} \\ \text{else if} \quad x_i^d = j, \qquad & \Pr_{\mathcal{D}_1,\alpha}(x_i^d = j \mid h(\mathbf{x}^d_{-i})) = \frac{n_{ij} + 1 + \alpha}{N_{h(\mathbf{x}_{-i})} + 1 + C_i\alpha} \end{aligned}$$

Given $\alpha > 0$, we obtain the upper-bound for the ith component as follows:

$$\max_{\mathcal{D}_1,\mathcal{D}_2} \frac{\Pr_{\mathcal{D}_1,\alpha}(x_i = j \mid h(\mathbf{x}_{-i}))}{\Pr_{\mathcal{D}_2,\alpha}(x_i = j \mid h(\mathbf{x}_{-i}))} \leq \max_{\mathcal{D}_1,\mathcal{D}_2} \frac{\frac{n_{ij}+1+\alpha}{N_{h(\mathbf{x}_{-i})}+1+C_i\alpha}}{\frac{n_{ij}+\alpha}{N_{h(\mathbf{x}_{-i})}+C_i\alpha}} \leq 1 + \frac{1}{\alpha}$$

where the first inequality is because the two datasets only differ by at most one element. The second inequality comes from the fact that $\frac{N_{h(\mathbf{x}_{-i})}+C_i\alpha}{N_{h(\mathbf{x}_{-i})}+1+C_i\alpha} < 1$ and that the equation is maximized when $n_{ij} = 0$. As we iterate this process for the M variables, the differential probability is upper-bounded by:

$$\frac{\prod_{i=1}^{M} \mathrm{Pr}_{\mathcal{D}_1,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))}{\prod_{i=1}^{M} \mathrm{Pr}_{\mathcal{D}_2,\alpha}(x_i \mid h(x_{1:(i-1)}, s_{(i+1):M}))} \leq \prod_{i}^{M}(1 + \frac{1}{\alpha})$$

Therefore, we obtain the relation between α and ϵ as follows:

$$M \log(1 + \frac{1}{\alpha}) \leq \epsilon$$

Rearranging the terms, we have:

$$\alpha \geq \frac{1}{\exp(\epsilon/M) - 1} \tag{8}$$

Note that for univariate binary synthetic data, McClure and Reiter (2012) showed the relationship between α and ϵ as $\alpha = \frac{1}{\exp(\epsilon)-1}$. Equation (8) says that a higher level of privacy (low ϵ) needs a high value of α. Intuitively, high values of α mean stronger priors, thus the synthetic data are more strongly masked by the priors (or virtual samples).

B. l-Diversity. For l-diversity (See Equation 3), perturbed conditional distributions need to satisfy the synthetic l-diversity criterion:

$$H_\alpha(x_i \mid \mathbf{x}_{-i}) = -\sum_j \mathrm{Pr}_{\mathcal{D},\alpha} \log \mathrm{Pr}_{\mathcal{D},\alpha} \geq \log l$$

where $H_\alpha(x_i \mid \mathbf{x}_{-i})$ is the Shannon entropy of the perturbed distribution, $\mathrm{Pr}_{\mathcal{D},\alpha}$. The entropy H_α is a monotonically increasing function with respect to α. To satisfy the synthetic l-diversity criterion with minimal perturbation, we set α as follows:

$$\alpha = \begin{cases} \alpha^* \quad \text{s.t.} \quad -\sum_j \mathrm{Pr}_{\mathcal{D},\alpha} \log \mathrm{Pr}_{\mathcal{D},\alpha} = \log l, & \text{if} \quad H_\alpha < \log l \\ 0, & \text{otherwise} \end{cases}$$

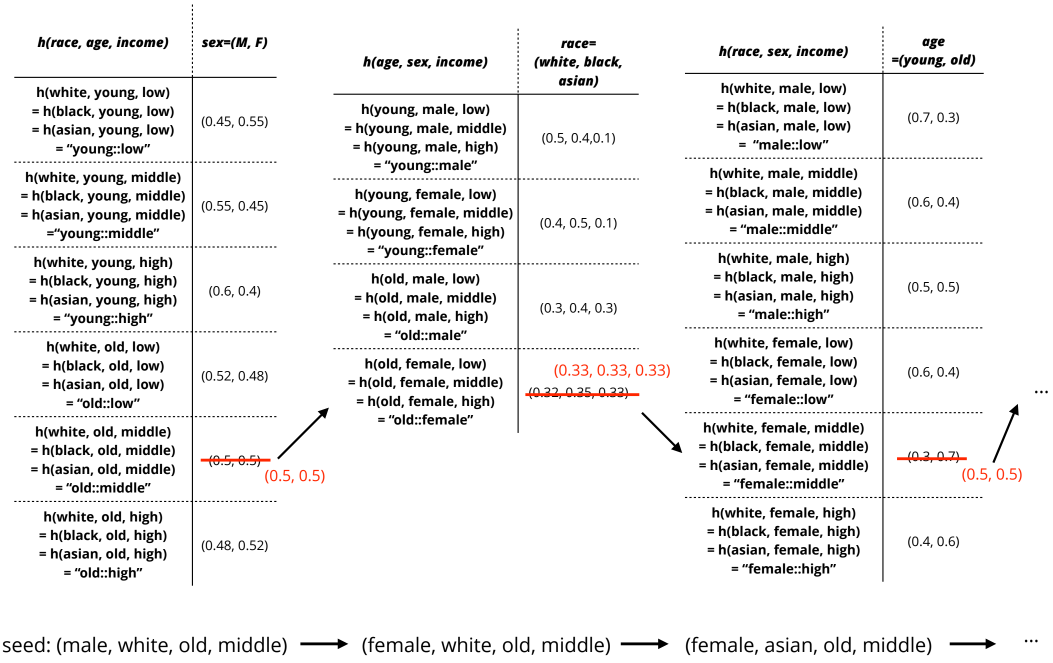
where α is set to zero when H_α already satisfies the l-diversity criterion. Unlike the single α for differential privacy, the α values for l-diversity vary depending on conditional distributions. This is because l-diversity applies to a dataset, whereas differential privacy applies to a function. l-diversity is data-aware, but may not provide rigorous guarantees for privacy. This is also noted in (Clifton and Tassa, 2013) who observed that syntactic methods such as k-anonymity and l-diversity are designed for privacy-preserving data publishing, while differential privacy is typically applicable for privacy-preserving data mining. Thus these two approaches are not directly competing, and indeed can be used side-by-side. This paper also provides a detailed assessment of both the limitations and promise of both types of approaches.

3.4 Removing Sampling Footprints

This section illustrates an effective block sampling extension of PeGS, and is specific to differential privacy. PeGS generates one synthetic sample for one seed sample. In other words, one synthetic sample costs ϵ in the differential privacy regime. We modify the PeGS algorithm to sample a block of samples from one seed sample, while achieving the same ϵ-differential privacy. One sampling iteration of PeGS is now repeated many times, but each time, the visited conditional distributions are reset. The procedure of Block PeGS with Reset (PeGS.rs) is as follows:

1. Pick a random seed s from a predefined pool.
2. For b in $1 : B$,
   - (a) Sample $\mathbf{x}^{(b)}$ using PeGS seeded by the previous sample $\mathbf{x}^{(b-1)}$, where $\mathbf{x}^{(0)} = \mathbf{s}$
   - (b) Reset all visited conditional distributions $\Pr(x_i \mid h(\mathbf{x}_{-i}))$ to uniform distributions

This algorithm produces a block of synthetic samples $(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(B)})$ with the same privacy cost ϵ. Figure 4 illustrates the process of PeGS with Reset. The synthesizing process of PeGS.rs is exactly the same as the process of PeGS Figure 3 except for the resetting step. After sampling from conditional tables, the probability distribution of the visited bin is set to a uniform distribution. The red lines in Figure 4 illustrate this resetting step. In our block sampling scheme, there is a chance of re-visiting the bins that are already visited in the previous sampling steps. For such cases, new samples are drawn from uniform distributions, since probability estimates are reset to uniform.



h(race, age, income)	*sex=(M, F)*
h(white, young, low) = h(black, young, low) = h(asian, young, low) = "young::low"	(0.45, 0.55)
h(white, young, middle) = h(black, young, middle) = h(asian, young, middle) ="young::middle"	(0.55, 0.45)
h(white, young, high) = h(black, young, high) = h(asian, young, high) = "young::high"	(0.6, 0.4)
h(white, old, low) = h(black, old, low) = h(asian, old, low) = "old::low"	(0.52, 0.48)
h(white, old, middle) = h(black, old, middle) = h(asian, old, middle) = "old::middle"	~~(0.5, 0.5)~~ (0.5, 0.5)
h(white, old, high) = h(black, old, high) = h(asian, old, high) = "old::high"	(0.48, 0.52)

h(age, sex, income)	*race= (white, black, asian)*
h(young, male, low) = h(young, male, middle) = h(young, male, high) = "young::male"	(0.5, 0.4,0.1)
h(young, female, low) = h(young, female, middle) = h(young, female, high) = "young::female"	(0.4, 0.5, 0.1)
h(old, male, low) = h(old, male, middle) = h(old, male, high) = "old::male"	(0.3, 0.4, 0.3)
h(old, female, low) = h(old, female, middle) = h(old, female, high) = "old::female"	(0.33, 0.33, 0.33) ~~(0.32, 0.35, 0.33)~~

h(race, sex, income)	*age =(young, old)*
h(white, male, low) = h(black, male, low) = h(asian, male, low) = "male::low"	(0.7, 0.3)
h(white, male, middle) = h(black, male, middle) = h(asian, male, middle) = "male::middle"	(0.6, 0.4)
h(white, male, high) = h(black, male, high) = h(asian, male, high) = "male::high"	(0.5, 0.5)
h(white, female, low) = h(black, female, low) = h(asian, female, low) = "female::low"	(0.6, 0.4)
h(white, female, middle) = h(black, female, middle) = h(asian, female, middle) = "female::middle"	~~(0.3, 0.7)~~ (0.5, 0.5)
h(white, female, high) = h(black, female, high) = h(asian, female, high) = "female::high"	(0.4, 0.6)

seed: (male, white, old, middle) → (female, white, old, middle) → (female, asian, old, middle) → ...

Figure 4: Synthesis Steps in PeGS with Reset. Visited rows in statistical building blocks are reset to the initial state. In this example, the initial states are uniform distributions over categories.

To analyze the privacy aspect of this modified PeGS algorithm, we first need to calculate the probability of synthesizing a block of samples:

$$\Pr^{B}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(B)} \mid \mathbf{s}) = \Pr^{(1)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(1)} \mid \mathbf{s}) \prod_{b=2}^{B} \Pr^{(b)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})$$

where $\Pr^{(b)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})$ is the transition probability from $\mathbf{x}^{(b-1)}$ to $\mathbf{x}^{(b)}$. Note that $\Pr^{(b)}$ and $\Pr^{(b+1)}$ are different conditional distributions, as M components of $\Pr^{(b)}$ are reset to the initial states. The ratio between two probabilities is written as follows:

$$\frac{\Pr^{(1)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(1)} \mid \mathbf{s}) \prod_{b=2}^{B} \Pr^{(b)}_{\mathcal{D}_1,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})}{\Pr^{(1)}_{\mathcal{D}_2,\alpha}(\mathbf{x}^{(1)} \mid \mathbf{s}) \prod_{b=2}^{B} \Pr^{(b)}_{\mathcal{D}_2,\alpha}(\mathbf{x}^{(b)} \mid \mathbf{x}^{(b-1)})} \le \exp(\epsilon)$$

Recall that the statistical building blocks from both datasets differ at most M components, as the two datasets differ at most one element. We provide a sketch of the proof that this algorithm satisfies ϵ-differential privacy as follows:

1. To generate the same block of samples, the sequences of statistical building blocks need to be the same as well. In other words, as the two samples, $\mathbf{x}^{(b)} \mid \mathcal{D}_1$ and $\mathbf{x}^{(b)} \mid \mathcal{D}_2$, are the same, $\mathbf{x}^{(b)}_{-i} \mid \mathcal{D}_1$ and $\mathbf{x}^{(b)}_{-i} \mid \mathcal{D}_2$ will also be the same. Thus, they use the building blocks from the same location for sampling x_i at the bth iteration, $\Pr^{(b)}_{\mathcal{D}_1,\alpha}(x_i \mid \mathbf{x}_{-i})$ and $\Pr^{(b)}_{\mathcal{D}_2,\alpha}(x_i \mid \mathbf{x}_{-i})$.

2. There are at most M different components between $\Pr^{(1)}_{\mathcal{D}_1,\alpha}$ and $\Pr^{(1)}_{\mathcal{D}_2,\alpha}$, and let $\mathcal{M}$ be the set of different components. This is because $\mathcal{D}_1$ and $\mathcal{D}_2$ differ by at most one row.

3. If $\Pr^{(1)}_{\mathcal{D}_1,\alpha}$ touched $(M-d)$ components in $\mathcal{M}$, then $\frac{(M-d)}{M}\epsilon$ privacy cost is spent in the process (see Section 3.3).

4. If $\Pr^{(1)}_{\mathcal{D}_1,\alpha}$ touched $(M-d)$ components in $\mathcal{M}$, then the rest of the sequences can differ at most d components. This is because those $(M-d)$ components are reset to uniform distributions, and they became indistinguishable i.e. the visited components from $\mathcal{D}_1$ and $\mathcal{D}_2$ became the same uniform distribution. Every visit of an element in $\mathcal{M}$ decreases the number of different elements.

5. Therefore, the whole sequence can differ at most M components (upper-bound), thus the proposed block sampling algorithm satisfies the same ϵ-differential privacy for generating a block of B samples.

As we have more samples for the same cost, the privacy cost per sample can be written as:

$$\alpha \ge \frac{1}{\exp(\epsilon' B/M) - 1} \tag{9}$$

where $\epsilon/B = \epsilon'$. The privacy cost is smaller by a factor of B. As an illustrative example, suppose that we need 10 synthetic samples that satisfy ϵ-differential privacy. To obtain 10 samples from PeGS, we perturb the statistical building block by $\alpha_{\text{PeGS}} = \frac{1}{\exp(\epsilon/M)-1}$. On the other hand, if we use $B = 10$, the amount of perturbation for PeGS.rs is given as $\alpha_{\text{PeGS.rs}} = \frac{1}{\exp(10\epsilon/M)-1}$, which can be much smaller than α_{PeGS}. However, the block size B cannot be arbitrarily large. As every visited statistical building block is reset, the synthetic samples tend to be more noisy as we increase the size of the block.

The relationship between α and ϵ represents the trade-off between utility and risk. Low α values can generate more realistic synthetic data, and low ϵ values can provide higher levels of privacy protection. Note the difference between Equation 8 and Equation 9. PeGS.rs has the additional parameter B that can fine-tune the relationship between α and ϵ. A smart choice of B can improve the trade-off curve depending on the characteristics of a dataset. This property will be illustrated using a real dataset in Section 4.

3.5 Perturbed Multiple Imputation

The Dirichlet perturbation can similarly be applied to multiple imputation, specifically the multiple imputation using sequential regressions. Perturbed Multiple Imputation is a naive extension of multiple imputation that satisfies ϵ-differential privacy. A multiple imputation with generalized linear models can be written as follows:

$$\Pr_{\hat{\mathbf{w}}(\mathbf{x})}(\mathbf{x}) = \prod_{i=1}^{M} g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}))$$

where $g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}))$ is the estimated response probability of x_i using a generalized linear model. We assume that the response is a *normalized* probability measure, thus $g_{x_i} \in [0, 1]$. We propose perturbed multiple imputation as follows:

$$\Pr_{\hat{\mathbf{w}}(\mathbf{x}),\alpha}(\mathbf{x}) = \prod_{i=1}^{M} g^{\alpha}_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}))$$

Perturbed multiple imputation satisfies ϵ-differential privacy, if the output is perturbed as

$$g^{\alpha}_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}) = \frac{g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}) + \alpha}{\sum_{x_i} g_{x_i \in X_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}) + C_i\alpha} = \frac{g_{x_i}(\hat{\mathbf{w}}_i(\mathcal{D}_1)^\top \mathbf{x}_{-i}) + \alpha}{1 + C_i\alpha}$$

where $\alpha = 1/(\exp(\epsilon/M) - 1)$. The proof is analogous to the proof for the PeGS algorithm. With $\alpha = 0$, this algorithm is the same as a multiple imputation with generalized linear model.

4 Empirical Study

In this section, we evaluate the PeGS algorithm using a real dataset from two perspectives: utility and risk of the PeGS-synthesized data. The utility is measured by comparing marginal, conditional distributions and regression coefficients with those from the original data. The risk is first measured by the differential privacy parameter ϵ. As the differential privacy parameter can be too conservative for a real dataset, we also measure population uniqueness and indirect probabilistic disclosure risks. The presented experiments are mainly for the differentially private perturbation, and the experiment with the l-diversity perturbation can be found in (Park et al., 2013a,b).

4.1 Dataset Overview

We use public Patient Discharge Data from California Office of Statewide Health Planning and Development[3]. This dataset contains inpatient, emergency care, and ambulatory surgery data collected from licensed California hospitals. Each row of the data represents either one discharge event of a patient or one outpatient encounter. The data are already processed with several disclosure limitation techniques. Feature generalization and masking rules are applied to the data based on population uniqueness.

For our experiment, we use 2011 Los Angeles data. Although there are almost 40 variables in the provided data, we use 13 important variables. The selected variables are listed in

[3] http://www.oshpd.ca.gov/HID/Data_Request_Center/Manuals_Guides.html

Table 2: California discharge data. Los Angeles.

Variable	Description	Category Values
typ	Type of care	Acute Care, Skilled Nursing, etc. (6 levels)
age.yrs	Age of the patient (5 years bin)	0, 5, 10, 15, ..., 80, NA (18 levels)
sex	Gender of the patient	Male, Female, NA (3 levels)
ethncty	Ethnicity of the patient	Hispanic, Non-Hispanic, etc. (4 levels)
race	Race of the patient	White, Black, Asian, etc. (7 levels)
patzip	Patient ZIP code (in LA)	900xx, 902xx, ... , 935xx (16 levels)
los	Length of stay (in days)	0, 1, 2, ... , 9, 50-70, 90+, NA (16 levels)
disp	The consequent arrangement	Routine, Acute Care, etc. (13 levels)
pay	Payment category	Meicare, Medi-Cal, Private, etc. (9 levels)
charge	Total hospital charges	0, 2K, 6K, 8K, 10K, ..., 100K+ (25 levels)
MDC	Major diagnostic category	Nervous sys., Eye, ENMT, etc. (25 levels)
sev	Severity code	0, 1, 2 (3 levels)
cat	Category code	Medical, Surgical (2 levels)

Table 2. For the numeric variables such as age and charge, we transformed the variables into categorical variables by grouping. We subset the data to focus on populous zip code areas, and use this preprocessed dataset to be our ground-truth original data. As can be seen, the possible combinations of the categories are approximately 2 trillion: $2 \times 10^{12} \approx 6 \times 18 \times 3 \times 4 \times 7 \times 16 \times 16 \times 13 \times 9 \times 25 \times 25 \times 3 \times 2$. A table of this size cannot be stored in a personal computer.

Diagnostic and procedural codes are not included in this experiment. In the original data, diagnoses and procedures are coded following the rules of International Classification of Diseases (ICD-9). Both codes can specify very fine levels of diagnoses and procedures; for example, the ICD-9 codes include information about a underlying disease and a manifestation in a particular organ. These diagnostic and procedural codes can be grouped into a smaller number of categories. Major Diagnostic Categories (MDC) and Medicare Severity Diagnosis-Related Group (MSDRG) are two examples of coarser diagnostic codes. In this example, we only include higher level abstractions of the detailed features. To keep the semantics of the data, we recommend a two step procedure: first generating a higher level feature, then synthesizing detailed features based on the higher level feature.

Three numeric variables, `age`, length-of-stay (`los`), and `charge`, are grouped and transformed into categorical features. The age variable is equipartitioned to have 5 years gap between consecutive categories. The `los` and `charge` variables are grouped based on their marginal distributions. For example, almost half of the population stayed less than 10 days in a hospital. Thus, the `los` variable is grouped to have 1 day gap before 10 days threshold, and 20 days gap after 10 days. The `charge` variable exhibited a similar marginal distribution; almost a half of the population pay less than 20K dollars, and we binned this variable to have almost equal sizes of population. The grouping rules are illustrated in Table 2. In Section 5, we will discuss the limitations and extension of treating numeric variables in the PeGS framework.

Table 3: Detailed Sampling Steps in the PeGS synthesis step. Four variables, sex, race, payment category (pay), category code (cat), are not changed in the final transformed example.

sequence	typ	age	sex	eth	race	zip	los	disp	pay	chg	MDC	sev	cat
seed	4	55	2	1	1	917	8	1	3	40K	25	1	M
$X_1 \mid X_{-1}$	*5*	55	2	1	1	917	8	1	3	40K	25	1	M
$X_2 \mid X_{-2}$	5	*75*	2	1	1	917	8	1	3	40K	25	1	M
$X_3 \mid X_{-3}$	5	75	*2*	1	1	917	8	1	3	40K	25	1	M
$X_4 \mid X_{-4}$	5	75	2	*2*	1	917	8	1	3	40K	25	1	M
$X_5 \mid X_{-5}$	5	75	2	2	*1*	917	8	1	3	40K	25	1	M
$X_6 \mid X_{-6}$	5	75	2	2	1	*913*	8	1	3	40K	25	1	M
$X_7 \mid X_{-7}$	5	75	2	2	1	913	*9*	1	3	40K	25	1	M
$X_8 \mid X_{-8}$	5	75	2	2	1	913	9	*5*	3	40K	25	1	M
$X_9 \mid X_{-9}$	5	75	2	2	1	913	9	5	*3*	40K	25	1	M
$X_{10} \mid X_{-10}$	5	75	2	2	1	913	9	5	3	*65K*	25	1	M
$X_{11} \mid X_{-11}$	5	75	2	2	1	913	9	5	3	65K	*7*	1	M
$X_{12} \mid X_{-12}$	5	75	2	2	1	913	9	5	3	65K	7	*0*	M
$X_{13} \mid X_{-13}$	5	75	2	2	1	913	9	5	3	65K	7	0	*M*

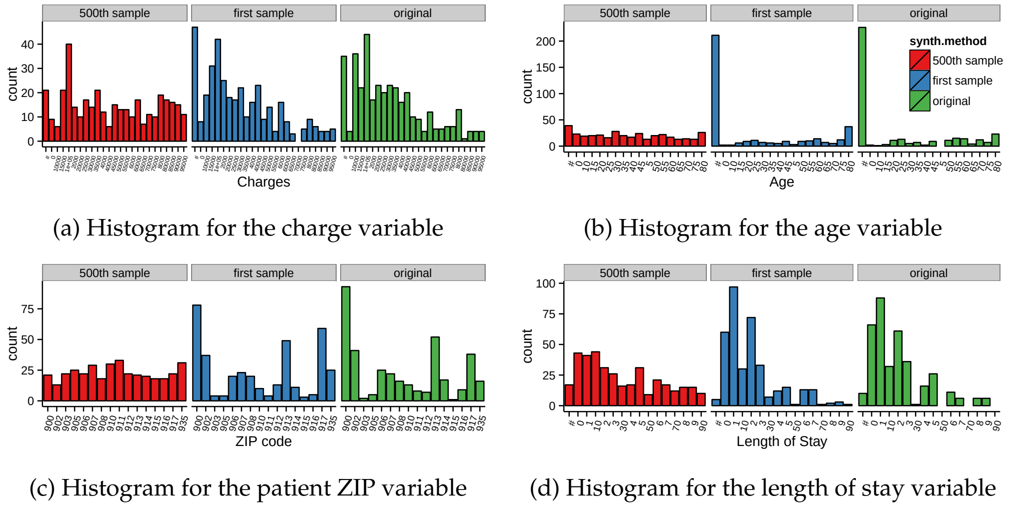
4.2 Sampling Demonstration

PeGS transforms each feature one by one conditioned on the rest of the features. This approach differs from a multiple imputation strategy in two aspects. First, PeGS estimates compressed conditional distributions rather than parameterized approximations e.g., generalized linear models. Second, the compressed conditional distributions can be further perturbed by calibrating the privacy parameter, which makes synthetic data ϵ-differentially private. Table 3 shows how PeGS transforms a random seed into a private synthetic sample. The first row of the table is a random seed, and each consecutive row shows the corresponding sampling step. Note that some features change their values, whereas other features maintain the original values. The final sample is shown in the last row. As can be seen, the final transformed sample is different from the seed; for example, it has a different age, zip code, and disposition code.

PeGS can be iterated many times, however, without the reset option, there is no gain on the privacy cost. The reset option in PeGS.rs removes sampling footsteps, and can generate multiple (or a block of) samples for the same privacy cost. However, the synthetic samples after many iterations may not be useful for representing the original data. We demonstrate this characteristics by comparing the first and 500th samples from PeGS.rs. We set $\epsilon = 1$ for each sample block, and used 350 random seeds to initialize PeGS.rs i.e. 350 sample blocks are generated. The sampling process is as follows:

```
For $i$ in $1:350$
    $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(500)}\} \sim \text{PeGS.rs}(\mathbf{s}^{(i)})$
    Put $\mathbf{x}^{(1)}$ in $\mathcal{D}_{\text{first samples}}$
    Put $\mathbf{x}^{(500)}$ in $\mathcal{D}_{\text{500th samples}}$
```

Figure 5 shows histograms from the generated samples. As expected, the samples from the later iterations are more uniformly distributed than those from PeGS. Moreover, as can be seen in the age and ZIP code variables, this behavior is more noticeable when the original distribution is skewed.



(a) Histogram for the charge variable

(b) Histogram for the age variable

(c) Histogram for the patient ZIP variable

(d) Histogram for the length of stay variable

Figure 5: Effect of block size on univariate distributions in the synthetic data. Distributions obtained based on the first sample are compared with those returned by the 500th sample.

4.3 Risk (ϵ) vs. Utility

Reducing disclosure risk and improving data utility are two competing objectives when publishing privacy-safe synthetic data. As these two goals cannot be satisfied at the same time, a certain trade-off is necessary for preparing public use data. This trade-off has been traditionally represented using a graphical measure, called R-U confidentiality map (Duncan et al., 2001). The R-U confidentiality map consists of two axis: typically a risk measure on the x-axis and a utility measure on the y-axis. Note that risk and utility measures can be domain and application specific. In this paper, we first show R-U maps where the risk is measured using differential privacy. The utility is primarily measured by comparing statistics from the original data and synthetic data.

We use three different algorithms and seven different privacy parameters for each algorithm as follows:

- **PeGS**: Perturbed Gibbs Sampler
- **PeGS.rs**: Perturbed Gibbs Block Sampler with Reset. Block size = 10.
- **PMI**: Perturbed Multiple Imputation (baseline algorithm). With higher values of ϵ, this is the same as a multiple imputation strategy for fully synthetic data. In PMI, the conditional distributions are modeled using the elastic-net regularized multinomial logistic regression, specifically `glmnet` package in `R` 2.15.3 (Friedman et al., 2010). The variable x_i is regressed on the rest of the variables $\mathbf{x}_{-i}$, and the regularization parameter λ was tuned based-on cross-validation:

$$\Pr(x_i = j \mid \mathbf{x}_{-i}) \propto \exp(c_{ij} + \boldsymbol{\beta}_{ij}^{\top}\mathbf{x}_{-i})$$

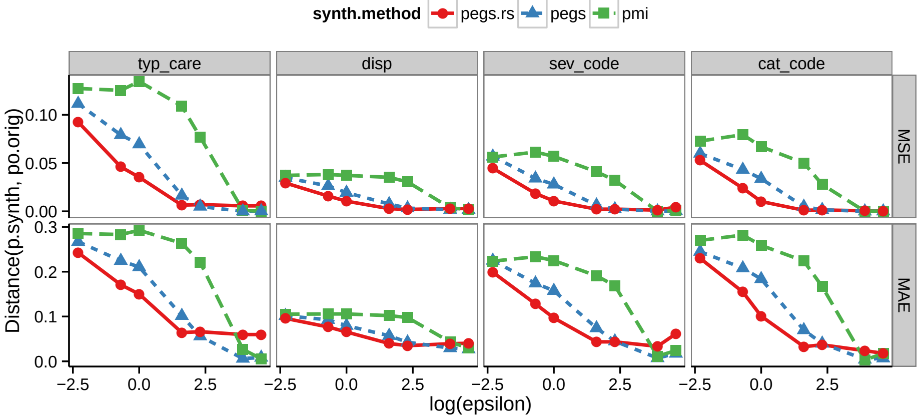


Figure 6: R-U maps where the negative utility is measured as the difference in marginal distributions; smaller distances imply greater utilities.

where c_{ij} and $\boldsymbol{\beta}_{ij}$ are estimated from the data.

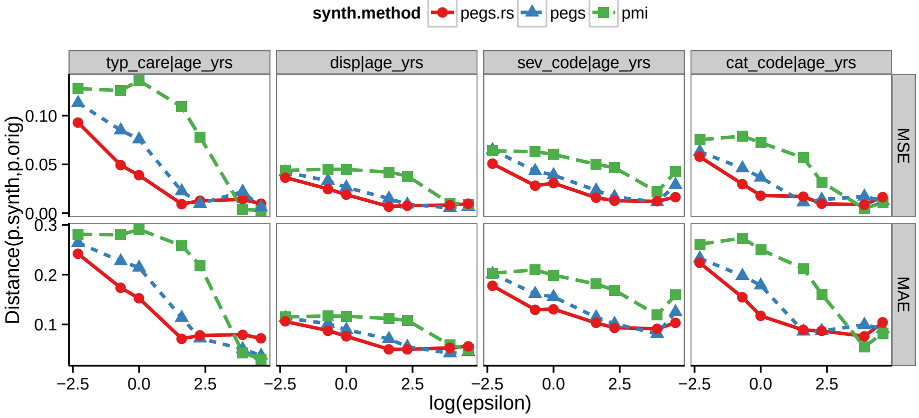
where the privacy parameters are given as $\epsilon \in \{0.1, 0.5, 1, 5, 10, 50, 100\}$ per synthetic sample. We generated 1000 samples for each case. As a result, we have $21 = 7 \times 3$ synthetic datasets and one original dataset.

The negative utility is first measured using the distance between marginal and conditional distributions. The term "negative" implies that smaller the distances, greater the utilities. Marginal and conditional distributions are measured from the original and synthetic datasets, then distances are calculated as follows:

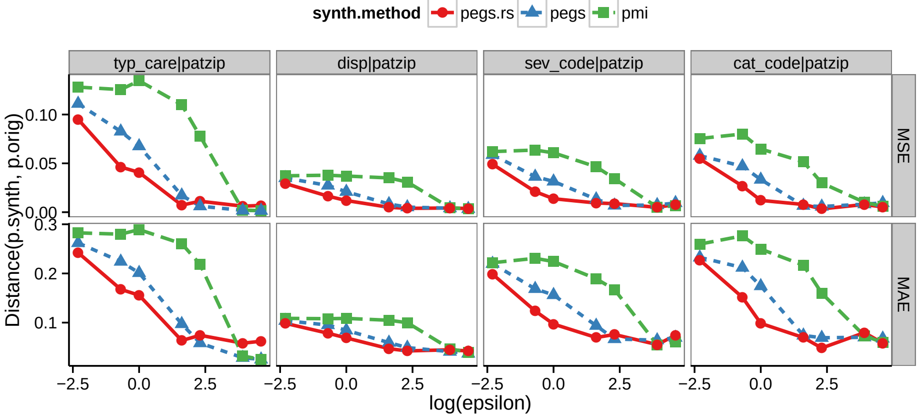
$$\begin{aligned}
\text{Marginal MSE} &= \text{Avg}_{x_i \in X_i} (\hat{\Pr}_{\text{synth},\epsilon}(x_i) - \hat{\Pr}_{\text{orig}}(x_i))^2 \\
\text{Marginal MAE} &= \text{Avg}_{x_i \in X_i} |\hat{\Pr}_{\text{synth},\epsilon}(x_i) - \hat{\Pr}_{\text{orig}}(x_i)| \\
\text{Conditional MSE} &= \text{Avg}_{x_j \in X_j} \text{Avg}_{x_i \in X_i} (\hat{\Pr}_{\text{synth},\epsilon}(x_i \mid x_j) - \hat{\Pr}_{\text{orig}}(x_i \mid x_j))^2 \\
\text{Conditional MAE} &= \text{Avg}_{x_j \in X_j} \text{Avg}_{x_i \in X_i} |\hat{\Pr}_{\text{synth},\epsilon}(x_i \mid x_j) - \hat{\Pr}_{\text{orig}}(x_i \mid x_j)|
\end{aligned}$$

where these distances are inverse surrogates for the utility. Figure 6 and Figure 7 show the R-U maps where the utility is measured as the difference in marginal and conditional distributions, respectively. As can be seen, all synthetic datasets become similar to the original data with higher values of ϵ. However, for smaller values of ϵ, the synthetic data from PeGS.rs are much more similar to the original than the others. The distributional distances of PeGS are slightly smaller than those of PeGS.rs for higher values of ϵ. Since α values are very small for these privacy parameters, the reset operation of PeGS.rs becomes more noticeable, and it pushes synthetic samples away from the original distributions.

The utility can be measured in many different ways. In this example, we examine whether synthetic samples preserve the ordering of marginal and conditional distributions of the original data. Marginal and conditional distributions are first ranked based on frequencies. We then compare the ranks from the original and synthetic distributions using Kendall's τ



(a) Distributional distances conditioned on the age variable, $X_i \mid$ age.yrs



(b) Distributional distances conditioned on the ZIP code variable, $X_i \mid$ patzip

Figure 7: R-U maps where the negative utility is measured as the difference in conditional distributions; smaller distances imply greater utilities.

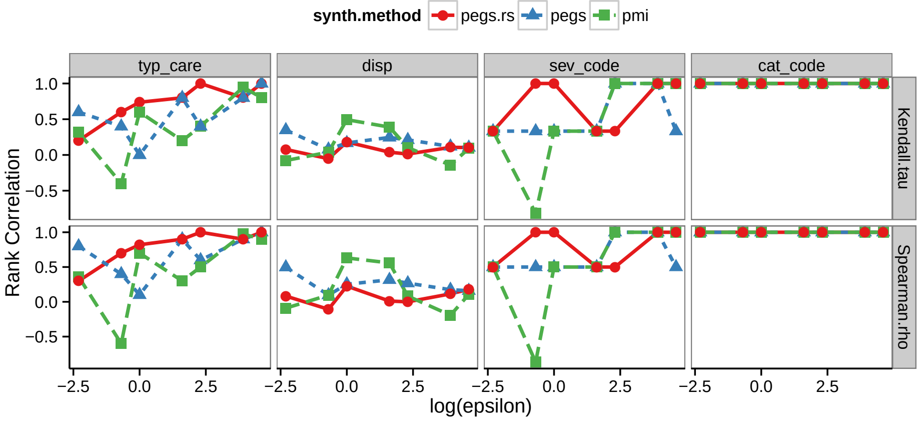


Figure 8: Rank correlation between marginal distributions vs. privacy parameter ϵ. As ϵ increases, the ordering of a marginal distribution remains the same as the original ordering.

and Spearman's ρ:

$$\text{Kendall's } \tau = \frac{\text{(number of concordant pairs)} - \text{(number of discordant pairs)}}{\frac{1}{2}C_i(1 - C_i)}$$

$$\text{Spearman's } \rho = \frac{\sum_j (r_j^o - C_i/2)(r_j^s - C_i/2)}{\sqrt{\sum_j (r_j^o - C_i/2)^2 \sum_j (r_j^s - C_i/2)^2}}$$

where C_i represents the number of categories for x_i, and r_j^o and r_j^s are ranks of the j category from the original and synthetic data, respectively. Both τ and ρ lie between -1 (strong negative correlation) and 1 (strong positive correlation). These rank correlation statistics are visualized in Figure 8 and 9 with respect to the privacy parameter ϵ. As can be seen, less perturbed synthetic datasets better preserve the ordering of distributions. Overall, PeGS.rs best preserves the frequency order of the original distributions.

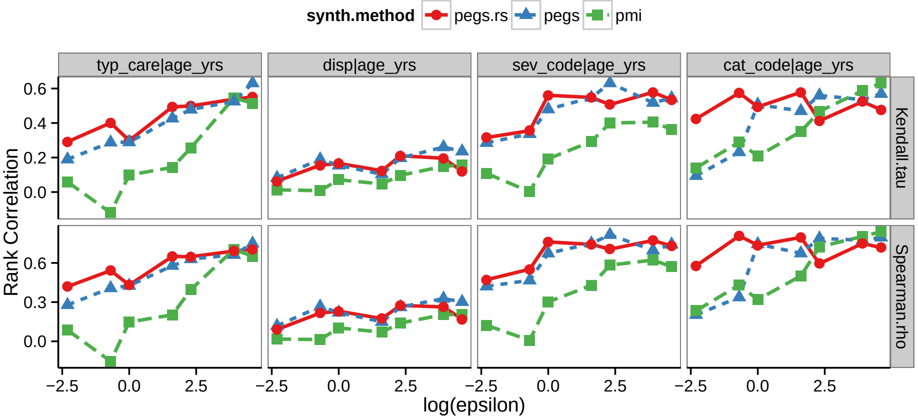
Next, we compare the coefficients from regression models learned on the datasets. We learned logistic and linear models as follows:

$$I(\text{charge} > 25K) \sim \text{as.numeric(age.yrs)} + \text{sev} + \text{cat} + \text{as.numeric(los)}$$
$$\text{as.numeric(charge)} \sim \text{as.numeric(age.yrs)} + \text{sev} + \text{cat} + \text{as.numeric(los)}$$

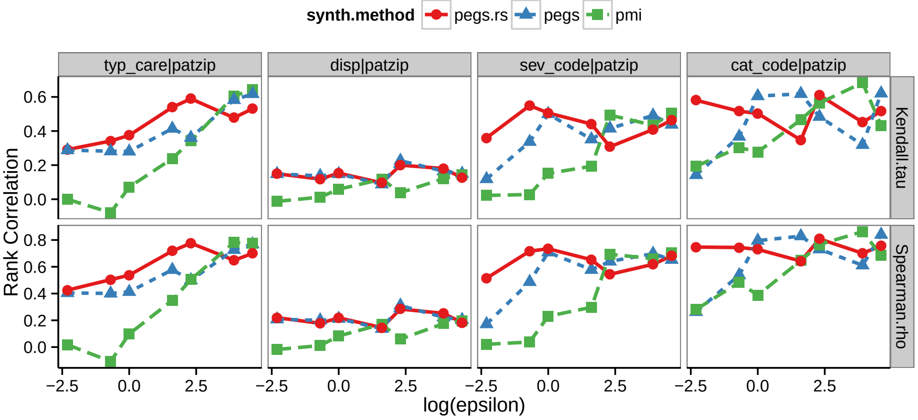
where some of the features are changed to numeric features based on their actual meaning. The choice of the target variable was arbitrary, as the goal of this illustrative experiment is to show the applicability of synthetic data in predictive modeling tasks. After learning the coefficients of each model, the distance between the coefficients is measured as follows:

$$\text{Regression Distance} = \sum_i \left|\frac{\beta_{i,\text{synth}} - \beta_{i,\text{orig}}}{\beta_{i,\text{orig}}}\right|$$

Figure 10 shows the R-U map from the regression experiment. As can be seen, the synthetic samples from PeGS.rs provide the most similar coefficients to those from the original



(a) Rank correlations conditioned on the age variable, $X_i \mid$ age.yrs



(b) Rank correlations conditioned on the ZIP code variable, $X_i \mid$ patzip

Figure 9: Rank correlation between conditional distributions vs. privacy parameter ϵ. As ϵ increases, the ordering of a marginal distribution remains the same as the original ordering.

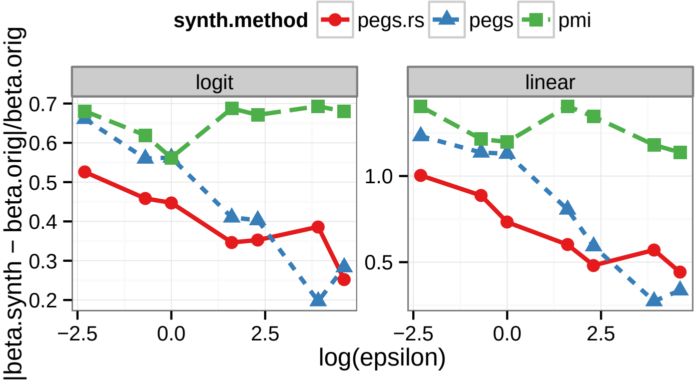
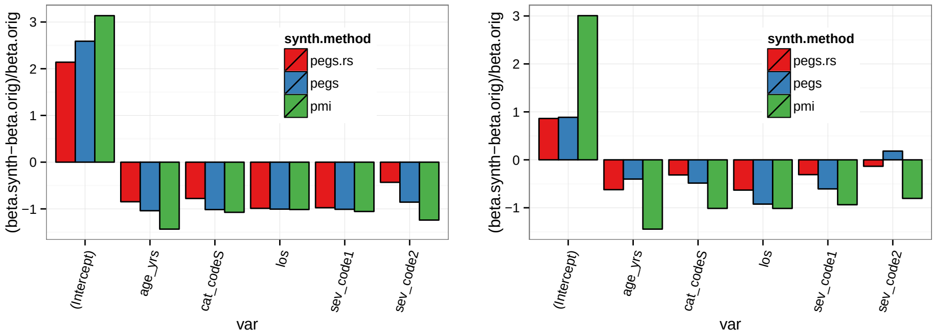


Figure 10: R-U maps where the utility is measured as the difference in regression coefficients.



(a) $\epsilon = 0.1$

(b) $\epsilon = 10$

Figure 11: Estimated logistic regression coefficients for (a) $\epsilon = 0.1$ and (b) $\epsilon = 10$. The coefficients from $\epsilon = 10$ (lower level of privacy) are closer to the original coefficients than those from $\epsilon = 0.1$ (higher level of privacy).

data. Figure 11 shows each coefficient deviation from the linear regression example. Notice that the intercept coefficients from the synthetic datasets tend to overshoot the actual value, while the other feature coefficients tend to undershoot. This is because the perturbation decreases all feature correlations including the correlation between the target and independent variables.

Another interesting observation is the poor performance of PMI in Figure 10 and 11. This behavior is mainly because of the model specification of PMI. In PMI, condition variables are treated as independent variables, thus the interaction effects of the conditional variables are not captured in its imputation model. Hospital charges are interactively affected by multiple factors such as severity and length of stay, and the hashing mechanism of PeGS is better suited for capturing such interactions.

4.4 Estimating Re-identification Risk

Although differential privacy provides a theoretically sound framework for measuring disclosure risks, the measure is originally designed for functions, not data (Dankar and Emam, 2012). For many cases, the measures can be overly conservative or strict for a real dataset. In the statistical disclosure limitation literature, there have been many attempts to measure disclosure risks for synthetic data. Franconi and Stander (2002) proposed a method to quantify disclosure risks for model-based synthetic data. Their proposed approach checks whether it is possible to recognize a unit in the released data assuming the original data are given to an intruder. This provides a somewhat conservative measure, but is still useful to compare the risks from different release mechanisms. Reiter (2005c) later formalized measuring probabilistic disclosure risk scores for partially or fully synthetic data. Probabilistic disclosure risks are used to asses the risks of the fully synthetic data using Random Forests in Caiola and Reiter (2010).

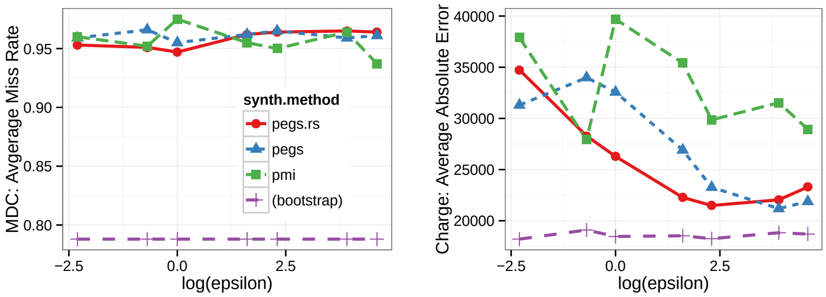
In this paper, we measure the disclosure risks by measuring the recoverability of feature values. Assuming that the intruder knows someone's `age`, `sex`, `los`, and `zip`, we measure the likelihood of getting the correct values as follows:

$$\mathrm{E}[\mathbb{1}(\text{inferred MDC} \neq \text{correct MDC}) \mid \text{age}, \text{sex}, \text{zip}]$$
$$\mathrm{E}[|\text{inferred charge} - \text{correct charge}| \mid \text{age}, \text{los}, \text{zip}]$$

where the inferred values are (1) the most frequent MDC categories and (2) sample means from conditioned synthetic samples. Figure 12 shows the results from this simulated intruder experiment. Private records are more difficult to reconstruct if misclassification rates and absolute errors are high. The probability of recovering MDC is significantly lower than using a simple bootstrap method, but no one method is distinctly better than the other. The absolute distance of hospital charges shows that synthetic data has comparable predictive power with the bootstrap method. Noticeably, the absolute errors are higher when the differential privacy parameters are low, and this finding partially supports our use of differential privacy as a disclosure risk measure.

5 Concluding Remarks

In this paper, we proposed a categorical data synthesizer that guarantees prescribed differential privacy or l-diversity levels. The use of a hash function allows the Perturbed Gibbs Sampler to handle high-dimensional categorical data. The non-parametric modeling of categorial data provides a flexible alternative to traditional (GLM-based) Multiple Imputation



(a) Attack on the MDC variable

(b) Attack on the charge variable

Figure 12: Simulated attack scenarios on the MDC and charge variables (left and right, respectively).

techniques. Additionally, this simple representation of conditional distributions is a crucial component of our block sampling algorithm that enhances the utility of synthetic data given a fixed privacy budget.

The California Patient Discharge dataset was used to demonstrate the analytical validity and utility of the proposed synthetic methodologies. Marginal and conditional distributions, as well as regression coefficients of predictive models learned from the synthesized data were compared to those from the original data to quantify the amount of distortion introduced by the synthesization process. Simulated intruder scenarios were studied to show the confidentiality of the synthesized data. The empirical studies showed that the proposed mechanisms can provide useful risk-calibrated synthetic data.

Currently, PeGS only deals with categorical variables. Numeric variables need to be binned to form categorical variables. Although this approach may be adequate enough for some applications, brute-force binning ignores numeric similarity or ordering information. For example, two consecutive values from an ordinal variable are more similar than separated values. Consider a size variable with three values: small, medium, and large. The ordering information states that similarity(small, medium) > similarity(small, big), but this information is lost if we bin the size variable into three (non-ordered) categories. Such semantic correlation cannot be captured in the current synthetic and perturbation model.

Although it was originally designed for computational efficiency, the hashing step of PeGS also provides an added degree of privacy protection. When building the PeGS statistical building blocks, each row $\mathbf{x}$ of the original data is hashed based on $h(\mathbf{x}_{-i})$, and aggregated with other rows with the same hash key, $\{\mathbf{z} \mid h(\mathbf{z}_{-i}) = h(\mathbf{x}_{-i})\}$. This aggregation (or hashing) step should be also incorporated for a tighter guarantee of privacy. The privacy guarantee of PeGS will be affected by different hash resolutions and mechanisms, and this topic needs to be covered in future work.

Although the proposed algorithms show substantially better performance on ϵ-differential privacy and l-diversity[4] measures, they were only marginally better than PMI in other probabilistic disclosure risk measures. The differential privacy measure may be too conservative for real data, and the probabilistic measure may not exhaustively capture all the attack scenarios. This is why we provided multiple risk measures. The connection between

[4] Experimental results on l-diversified synthetic data are presented in (Park et al., 2013a).

the differential privacy and disclosure risks should be further addressed to better evaluate the validity and utility of the synthetic data.

In practice, multiple disclosure techniques are sequentially mixed to achieve better protection of the records. For example, PeGS can be applied on top of feature generalization or masking techniques. Furthermore, some features can be modeled using generalized linear models; for example, numeric features. It would be worthwhile to investigate novel cocktails of different statistical disclosure limitation techniques.

Acknowledgment

We thank Mallikarjun Shankar and anonymous reviewers for helpful suggestions. This research is supported by Oak Ridge National Laboratory and NSF grant No. 1418511.

References

J. M. Abowd and L. Vilhuber. How protective are synthetic data? *Privacy in Statistical Databases*, 5262:239–246, 2008.

J. M. Abowd and S. D. Woodcock. Disclosure limitation in longitudinal linked data. *Confidentiality Diclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 215–277, 2001.

B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: A holistic solution to contingency table release. In *The 26th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2007.

A. L. Buczak, S. Babin, and L. J. Moniz. Data-driven approach for creating synthetic electronic medical records. *BMC Medical Informatics and Decision Making*, 10(59), 2010.

G. Caiola and J. P. Reiter. Random Forests for Generating Partially Synthetic, Categorical Data. *Transactions on Data Privacy*, 3:27–42, 2010.

Centers for Medicare and Medicaid Services. Medicare Claims Synthetic Public Use Files (SynPUFs). `http://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/SynPUFs/`, 2013.

A.-S. Charest. Empirical evaluation of statistical inference from differentially-private contingency tables. *Privacy in Statistical Databases*, 7556:257–272, 2012.

C. Clifton and T. Tassa. On syntactic anonymity and differential privacy. *Transactions on Data Privacy*, 6(2):161–183, 2013.

A. Dale and M. Elliot. Proposals for 2001 samples of anonymized records: an assessment of disclosure risk. *Journal of Royal Statistical Society: Series A*, 2001.

T. Dalenius and S. P. Reiss. Data-swapping: A technique for disclosure control (exteded abstract). In *Proceedings of the Section on Survey Research Methods*, 1978.

F. K. Dankar and K. E. Emam. The application of differential privacy to health data. In *Proceedings of Privacy and Anonymity in the Information Society (PAIS)*, 2012.

J. Drechsler and J. P. Reiter. Sampling with synthesis: A new approach for releasing public use census microdata. *Journal of the American Statistical Association*, 105(492):1347–1357, 2010.

J. Drechsler, S. Bender, and S. Rassler. Comparing fully and partially synthetic datasets for statistical disclosure control in the german IAB establishment panel. *Transactions on Data Privacy*, 1:105–130, 2008.

G. T. Duncan and D. Lambert. Disclosure-limited data dissemination. *Journal of the American Statistical Association*, 1986.

G. T. Duncan, S. A. Keller-McNulty, and S. L. Stokes. Disclosure risk vs. data utility: The R-U confidentiality map. Technical report, National Institute of Statistical Sciences, 2001.

C. Dwork. Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, volume 4052, pages 1–12, 2006.

S. E. Fienberg and J. McIntyre. Data swapping: Variations on a theme by Dalenius and Reiss. *Journal of Official Statistics*, 21:309–323, 2005.

S. E. Fienberg, A. Rinaldo, and X. Yang. Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables. In *The 2010 International Conference on Privacy in Statistical Databases*, pages 187–199, 2010.

L. Franconi and J. Stander. A model-based method for disclosure limitation of business microdata. *Journal of the Royal Statistical Society, Series D*, 51(1):51–61, 2002.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 2010.

W. A. Fuller. Masking procedures for microdata disclosure limitation. *Journal of Official Statistics*, 9(2):383–406, 1993.

A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th Very Large Database*, 1999.

P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of 30th Symposium on Theory of Computing*, 1998.

R. J. A. Little. Statistical analysis of masked data. *Journal of Official statistics*, 9(2):407–426, 1993.

J. S. Lombardo and L. J. Moniz. A method for generation and distribution of synthetic medical record data for evaluation of disease-monitoring systems. *Johns Hopkins APL Technical Digest*, 27(4):356–365, 2008.

A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramanian. l-diversity: Privacy beyond k-anonymity. *Transactions on Knowledge Discovery from Data*, 1, 2007.

A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets Practice on the Map. In *Proceedings of the 24th International Conference on Data Engineering*, 2008.

D. McClure and J. P. Reiter. Differential privacy and statistical disclosure risk measures: An investigation with binary synthetic data. *Transactions on Data Privacy*, 5:535–552, 2012.

F. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2009.

F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual Symposium of Foundations of Computer Science*, 2007.

S. Merugu and J. Ghosh. Privacy preserving distributed clustering using generative models. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 211–218, 2003.

S. Merugu and J. Ghosh. A distributed learning framework for heterogeneous data sources. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 208–217, 2005.

Y. Park, J. Ghosh, and M. Shankar. Lugs: A scalable non-parametric data synthesizer for privacy preserving big health data publication. In *International Conference on Machine Learning WHEALTH 2013*, 2013a.

Y. Park, J. Ghosh, and M. Shankar. Perturbed gibbs samplers for generating large-scale privacy-safe synthetic health data. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pages 493–498, 2013b.

S. Polettini. Maximum entropy simulation for microdata protection. *Statistics and Computing*, 13:307–320, 2003.

T. E. Raghunathan, J. M. Lepkowski, J. V. Hoewyk, and P. Solenberger. A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, 27(1):85–95, 2001.

T. E. Raghunathan, J. P. Reiter, and D. B. Rubin. Multiple imputation for statistical disclosure limitation. *Journal of Official Statistics*, 19(1):1–16, 2003.

J. P. Reiter. Releasing multiply imputed, synthetic public use microdata: an illustration and empirical study. *Journal of the Royal Statistical Society, Series A*, 168:185–205, 2005a.

J. P. Reiter. Using CART to generate partially synthetic, public use microdata. *Journal of Official Statistics*, (441-462), 2005b.

J. P. Reiter. Estimating risks of identification disclosure in microdata. *Journal of American Statistical Association*, 100(472):1103–1112, 2005c.

J. P. Reiter and J. Drechsler. Releasing multiply imputed synthetic data generated in two stages to protect confidentiality. *Statistica Sinica*, 20:405–421, 2010.

D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley, 1987.

D. B. Rubin. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9: 462–468, 1993.

J. W. Sakshaug and T. E. Raghunathan. Synthetic data for small area estimation. *Privacy in Statistical Databases*, 6344:162–173, 2011.

J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman and Hall, 1997.

J. Soria-Cormas and J. Drechsler. Evaluating the potential of differential privacy mechanisms for census data. In *UNECE Conference of European Statisticans*, 2013.

L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, October 2002. ISSN 0218-4885.

S. van Buuren and K. Groothuis-Oudshoorn. MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3), 2011.

J. Vreeken and A. Siebes. Filling in the blanks - KRIMP minimisation for missing data. In *the 8th IEEE International Conference on Data Mining*, pages 1067–1072, 2008.

K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

L. Willenborg and T. de Waal. *Elements of Statistical Disclosure Control*, volume 155. Springer, 2001.

W. E. Winkler. A contingency-table model for imputing data satisfying analytic constraints. *U.S. Census Bureau Statistical Research Division Research Report Series*, 2003.

W. E. Winkler. General discrete-data modeling methods for producing synthetic data with reduced re-identification risk that preserve analytic properties. *U.S. Census Bureau Statistical Research Division Research Report Series*, 2010.

X. Yang, S. E. Fienberg, and A. Rinaldo. Differential privacy for protecting multi-dimensional contingency table data: Extensions and applications. *Journal of Privacy and Confidentiality*, 4(1):101–125, 2012.